



Exceptional service in the national interest

SPIN-ORBIT INTERACTIONS IN QMCPACK

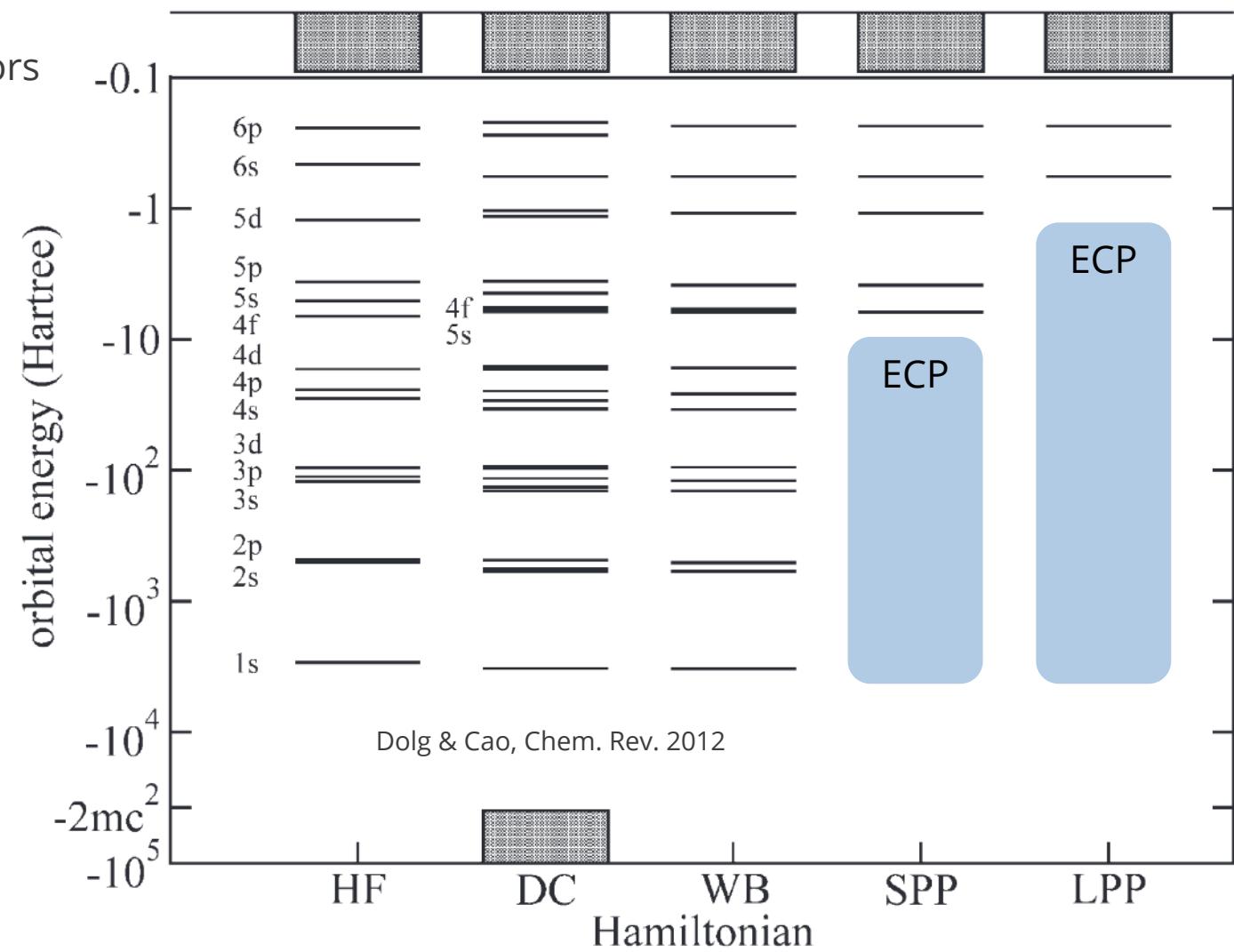
Cody A. Melton

December 13, 2023
2023 QMCPACK Users Workshop

RELATIVISTIC EFFECTIVE CORE POTENTIALS (RECP)

In normal calculations, the ECP that replaces the core electrons is written using semi-local projectors over spherical harmonics.

$$W = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$



RELATIVISTIC EFFECTIVE CORE POTENTIALS (RECP)

In normal calculations, the ECP that replaces the core electrons is written using semi-local projectors over spherical harmonics.

$$W = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

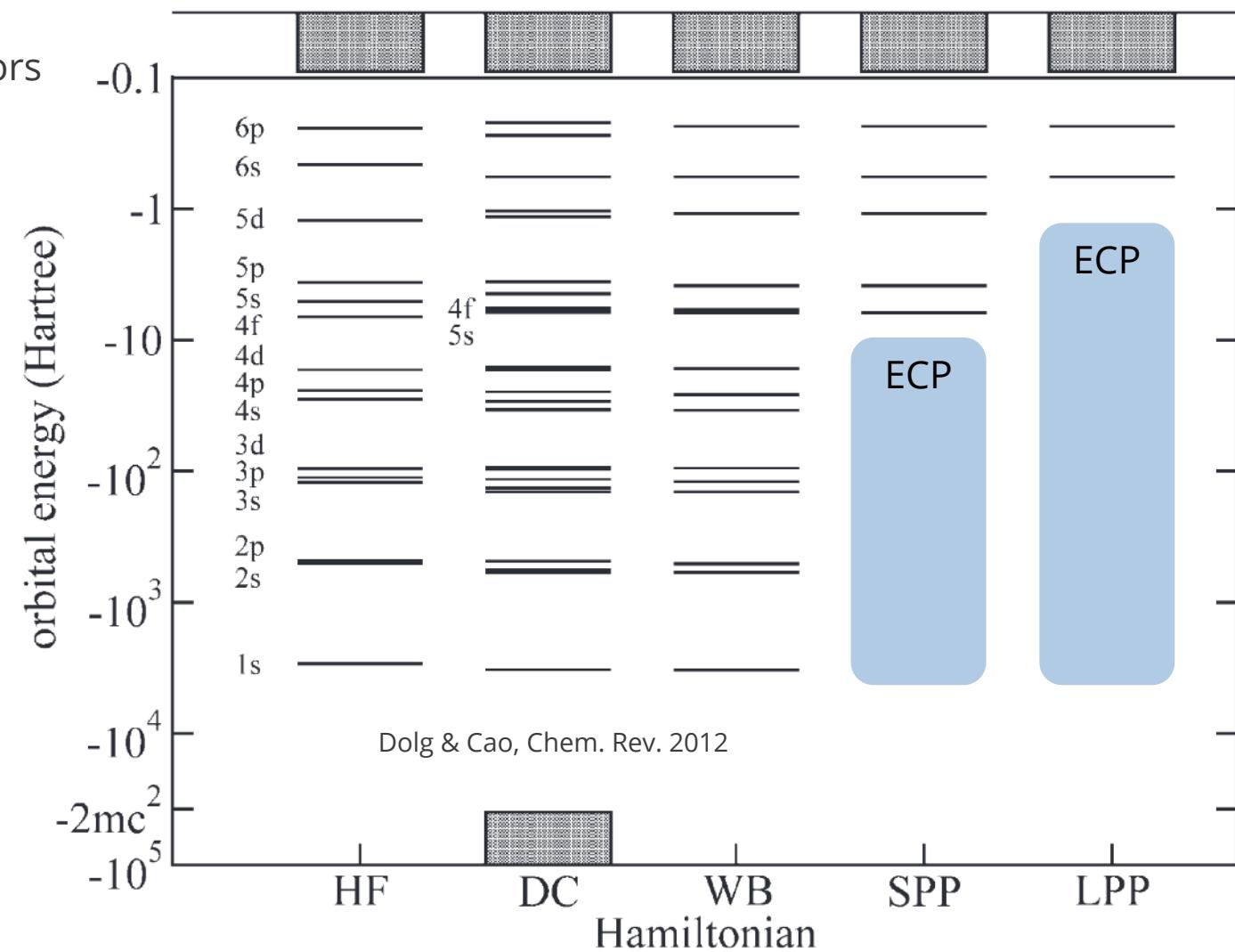
Recall that spin-orbit has an effective interaction

$$H_{SO} \propto \mathbf{L} \cdot \mathbf{S}$$

This implies that **L** and **S** aren't good quantum numbers. In this case, the total angular momentum **J** becomes a good quantum number.

$$W^{\text{avg. rel.}} = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

$$\Rightarrow W^{\text{rel.}} = \sum_{\ell, j, m_j} W_{\ell j}(r) |\ell j m_j\rangle \langle \ell j m_j|$$



RELATIVISTIC EFFECTIVE CORE POTENTIALS (RECP)

In normal calculations, the ECP that replaces the core electrons is written using semi-local projectors over spherical harmonics.

$$W = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

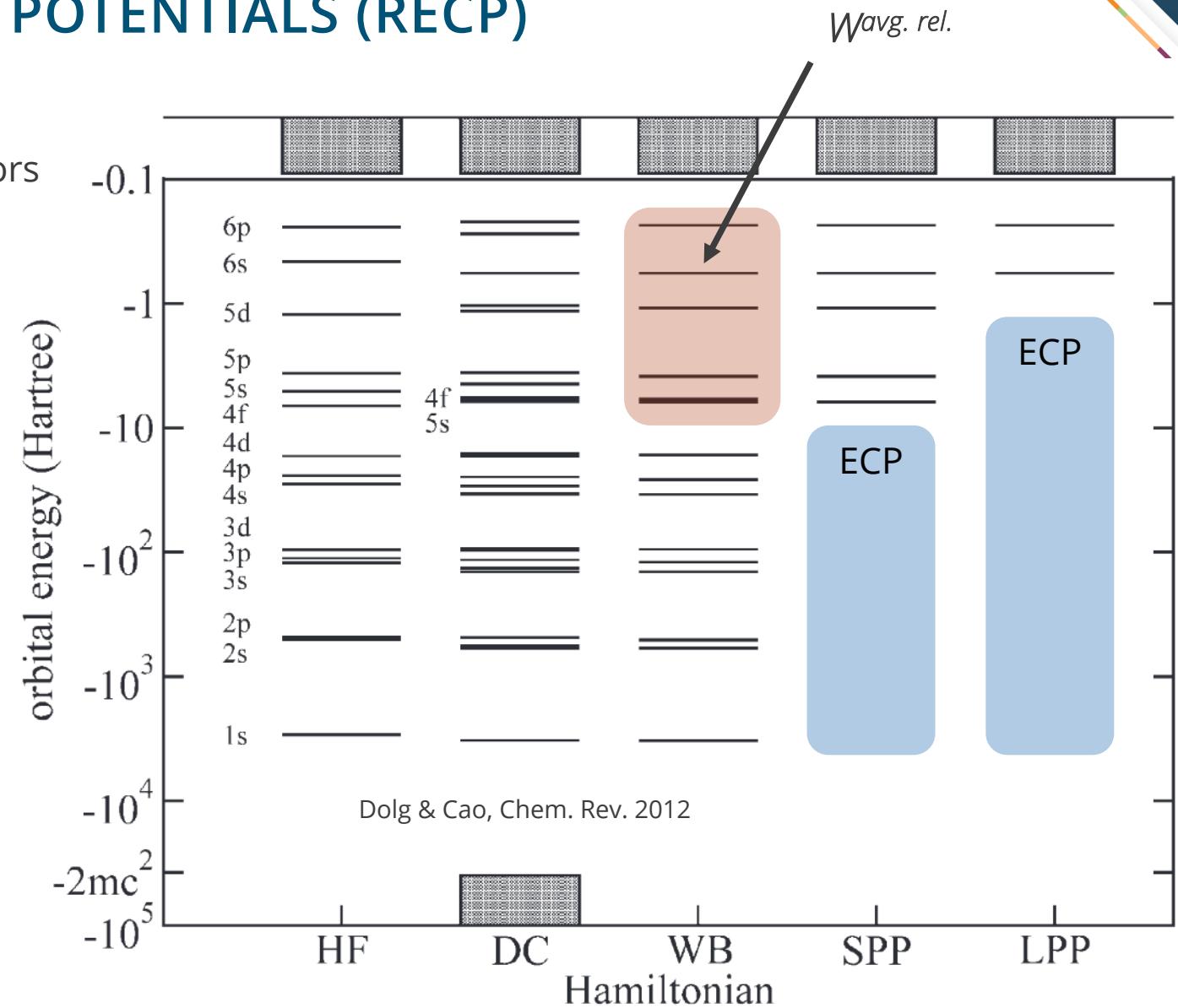
Recall that spin-orbit has an effective interaction

$$H_{SO} \propto \mathbf{L} \cdot \mathbf{S}$$

This implies that **L** and **S** aren't good quantum numbers. In this case, the total angular momentum **J** becomes a good quantum number.

$$W^{\text{avg. rel.}} = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

$$\Rightarrow W^{\text{rel.}} = \sum_{\ell, j, m_j} W_{\ell j}(r) |\ell j m_j\rangle \langle \ell j m_j|$$



RELATIVISTIC EFFECTIVE CORE POTENTIALS (RECP)

In normal calculations, the ECP that replaces the core electrons is written using semi-local projectors over spherical harmonics.

$$W = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

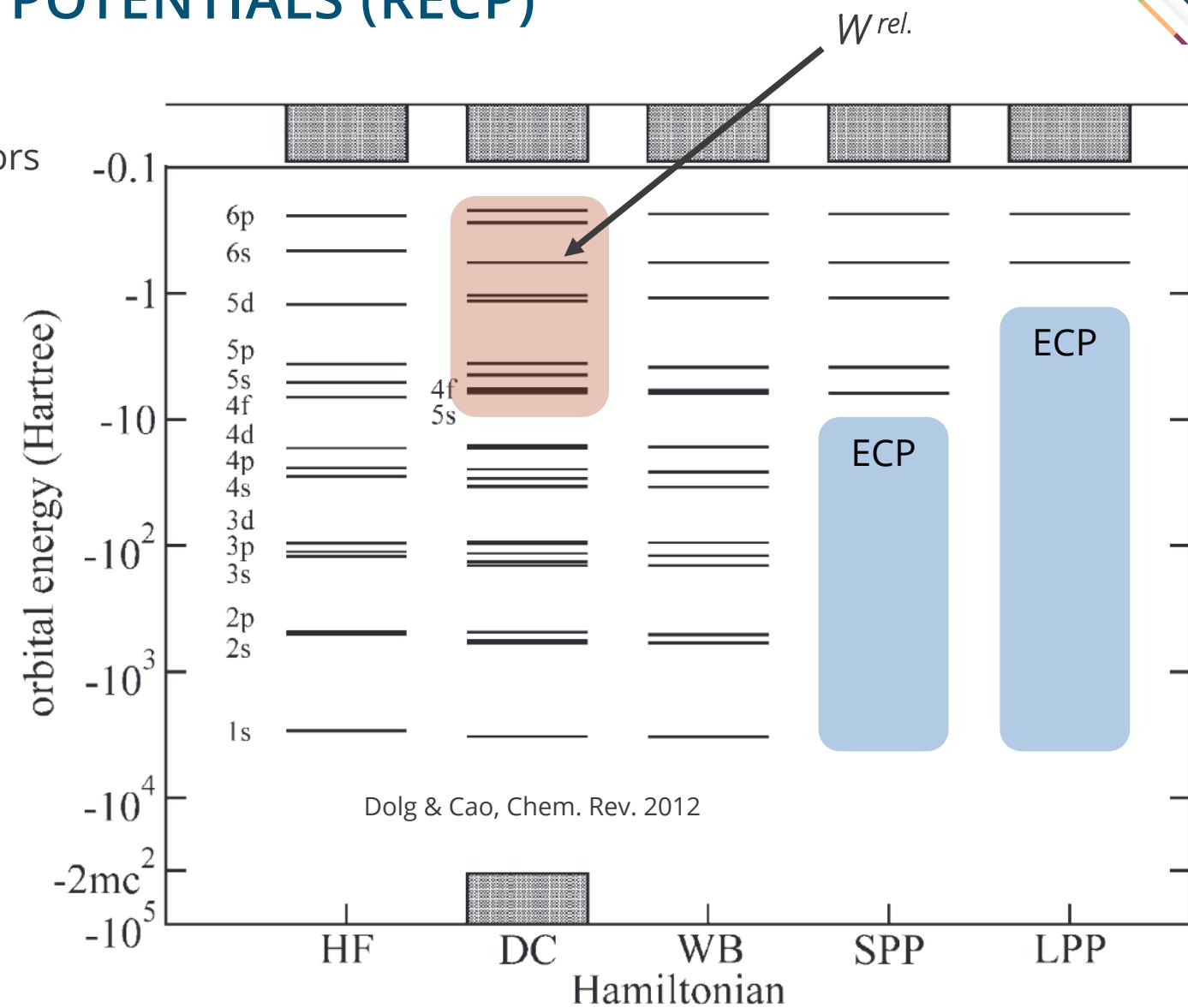
Recall that spin-orbit has an effective interaction

$$H_{SO} \propto \mathbf{L} \cdot \mathbf{S}$$

This implies that **L** and **S** aren't good quantum numbers. In this case, the total angular momentum **J** becomes a good quantum number.

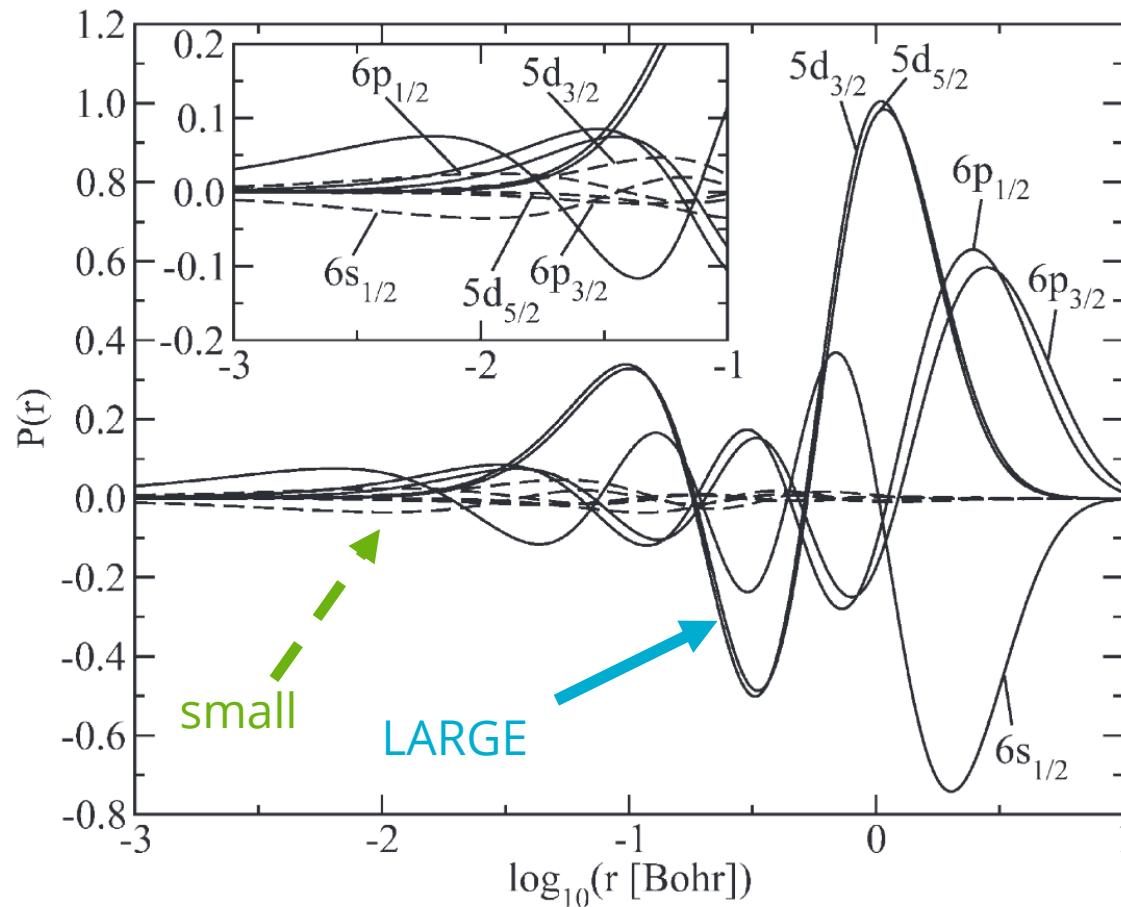
$$W^{\text{avg. rel.}} = \sum_{\ell, m_\ell} W_\ell(r) |\ell m_\ell\rangle \langle \ell m_\ell|$$

$$\Rightarrow W^{\text{rel.}} = \sum_{\ell, j, m_j} W_{\ell j}(r) |\ell j m_j\rangle \langle \ell j m_j|$$



HOW TO INCLUDE RELATIVITY IN QMC WITHOUT DIRAC EQUATION?

Core electrons provide an effective potential for valence electrons, implies we should use effective core potentials to remove inert core electrons. Do we still need to use 4-component wave functions?



Valence 4-component spinors for the Pb atom (left). The spinors can be written as “large” and “small” components, which are each 2-component spinors.

Note that the “small” components are completely negligible for the orbitals that participate in the chemical bonding of Pb.

This implies we should be able to incorporate relativistic effects for the chemically active electrons, using an effective core potential and 2-component spinors

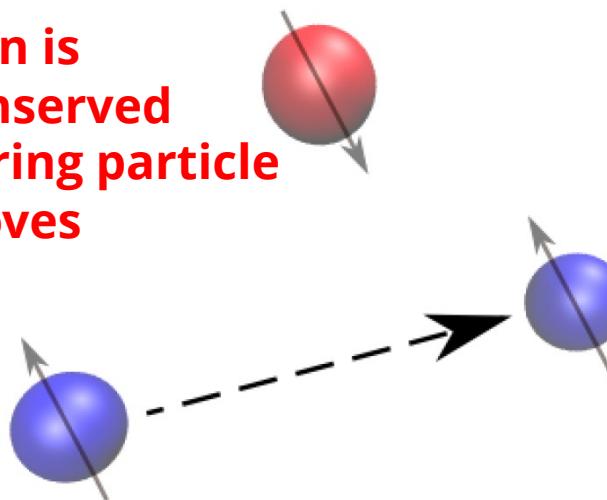
$$|\Psi\rangle = \begin{pmatrix} \Psi_L^{\uparrow} \\ \Psi_L^{\downarrow} \\ \Psi_S^{\uparrow} \\ \Psi_S^{\downarrow} \end{pmatrix} \xrightarrow{\text{valence}} \begin{pmatrix} \Psi_L^{\uparrow} \\ \Psi_L^{\downarrow} \end{pmatrix}$$

ELECTRONIC SPINS NEED TO BE SAMPLED

In standard QMC, since there are no spin interactions, spin commutes with the Hamiltonian. In this case, we can label each electron as "up" or "down" and that is preserved throughout the simulation.

$$[\mathcal{H}, S] = 0$$

**Spin is
conserved
during particle
moves**



ELECTRONIC SPINS NEED TO BE SAMPLED

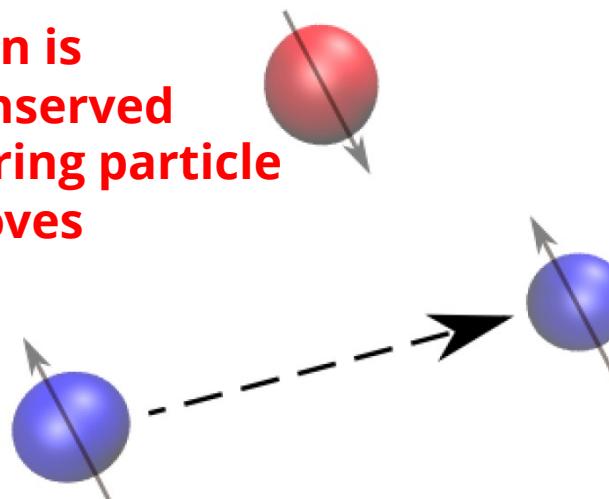
In standard QMC, since there are no spin interactions, spin commutes with the Hamiltonian. In this case, we can label each electron as “up” or “down” and that is preserved throughout the simulation.

With spin-orbit, the electron spin **isn't** conserved and will need to be sampled along with the spatial degrees of freedom.

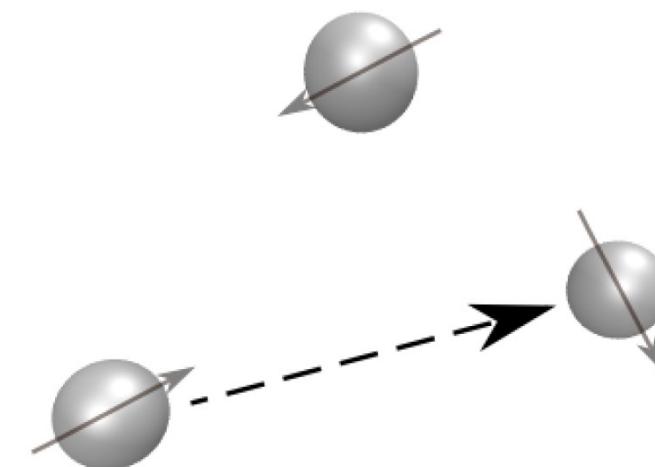
$$[\mathcal{H}, S] = 0$$

$$[\mathcal{H}, S] \neq 0$$

**Spin is
conserved
during particle
moves**



**The spin becomes
dynamic, and can
change during the
particle moves**



WHAT DO WE NEED TO IMPLEMENT IN QMCPACK?

- ❖ Single particle spinors in QMCPACK, and converters from DFT codes
- ❖ Many-body wave function, constructed from these single particle spinors
- ❖ VMC/DMC algorithm that incorporates sampling of the spin degree of freedom
- ❖ Spin-orbit part of the Hamiltonian using relativistic ECPs

Implementation following Melton, Bennett, & Mitas, JCP, 114 (2016)

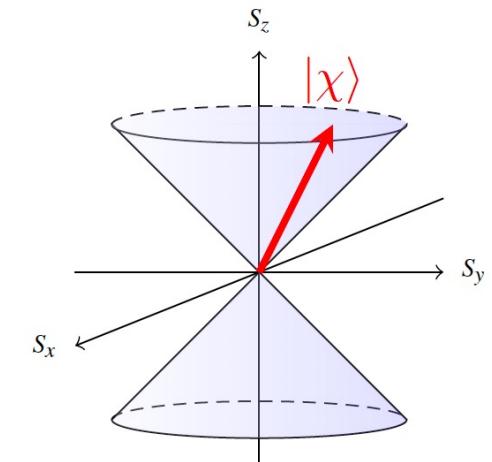
HOW TO REPRESENT SINGLE PARTICLE SPINORS

$$\langle \mathbf{x} | \psi \rangle = \phi^\uparrow(\mathbf{r}) \chi^\uparrow(s) + \phi^\downarrow(\mathbf{r}) \chi^\downarrow(s)$$

The general form of single particle spinors, with $|\mathbf{x}\rangle = |\mathbf{r}\rangle |s\rangle$

Standard (discrete) representation

$$\begin{aligned} \langle s | \chi^\uparrow \rangle &= \chi^\uparrow(s) = \delta_{s,1} \\ \langle s | \chi^\downarrow \rangle &= \chi^\downarrow(s) = \delta_{s,-1} \quad s = \pm 1 \end{aligned}$$



Once we move to the full many-body wave function, we need to sample the spin degree of freedom, i.e. sample the 2^N spin space. Within Monte Carlo sampling, a discrete change in an spin value could lead to a large energy change → increases the E_L fluctuations

The other option would be to just directly enumerate all spin configurations...however this has exponential scaling and becomes intractable for large systems.

This can become inefficient with large N. Can we find an alternative representation?

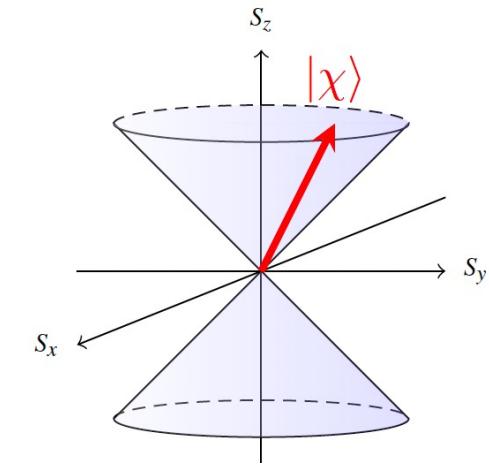
HOW TO REPRESENT SINGLE PARTICLE SPINORS

$$\langle \mathbf{x} | \psi \rangle = \phi^\uparrow(\mathbf{r}) \chi^\uparrow(s) + \phi^\downarrow(\mathbf{r}) \chi^\downarrow(s)$$

The general form of single particle spinors, with $|\mathbf{x}\rangle = |\mathbf{r}\rangle |s\rangle$

New continuous representation

$$\begin{aligned} \langle s | \chi^\uparrow \rangle &= \chi^\uparrow(s) = e^{is} \\ \langle s | \chi^\downarrow \rangle &= \chi^\downarrow(s) = e^{-is} \quad s \in [0, 2\pi) \end{aligned}$$



$$\langle \chi^\beta | \chi^\alpha \rangle = \int_0^{2\pi} \frac{ds}{2\pi} \langle \chi^\beta | s \rangle \langle s | \chi^\alpha \rangle = \delta_{\beta,\alpha}$$

Continuous representation preserves the properties of the spin states

With a continuous representation, the spin variable can be sampled in the same way the spatial variables are. Monte Carlo sampling is efficient, simply changes $\mathbf{R} \in \mathbb{R}^{3N} \Rightarrow \mathbf{X} \in \mathbb{R}^{4N}$

WHERE TO GET SINGLE-PARTICLE SPINORS?

Atoms and Molecules

P program
 • for
 A atomic
 • and
 M molecular

D direct
 I iterative
 R relativistic
 A all-electron
 C calculations



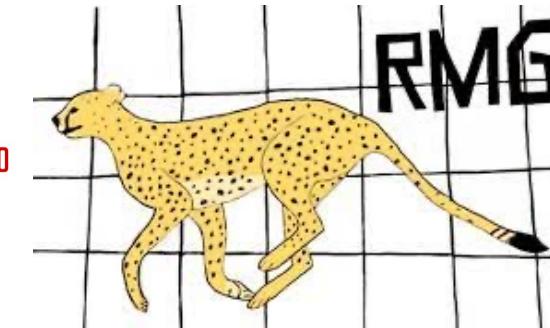
DIRAC is used to study relativistic quantum chemistry.

Can do non-relativistic, 2-component ECPs, 2-component all-electron, full 4-component treatments.

HF, COSCI, CISD, FCI, MP2, CCSD, and DFT

$$\psi_i(\mathbf{r}_j, s_j) = \phi_i^{\uparrow}(\mathbf{r}_j) e^{is_j} + \phi_i^{\downarrow}(\mathbf{r}_j) e^{-is_j}$$

Solid State Materials



Quantum Espresso and RMGDFT can provide DFT spinors for 2-component calculations with ECPs

Need to use a norm-conserving pseudopotential that includes spin-orbit terms.

Standard quantum chemistry pseudopotentials (with SOC) can be converted to Kleinmann-Bylander form for use in these codes. Currently, need to either use pseudopotentiallibrary.org to get relativistic potentials, or reach out for help converting potentials.

MANY-BODY WAVE FUNCTION

$$\Psi_T(\mathbf{R}) = e^{J(\mathbf{R})} \sum_{\alpha} c_{\alpha} \det_{\alpha}^{\uparrow} \left[\dots, \phi_i^{\uparrow}(\mathbf{r}_j), \dots \right] \det_{\alpha}^{\downarrow} \left[\dots, \phi_k^{\downarrow}(\mathbf{r}_l), \dots \right]$$



Change from orbitals $\{\phi_i^{\uparrow/\downarrow}(\mathbf{r}_j)\}$ to spinors $\{\psi_i(\mathbf{r}_j, s_j)\}$

$$\Psi_T(\mathbf{X}) = e^{J(\mathbf{R})} \sum_{\alpha} c_{\alpha} \det \left[\dots, \psi_i(\mathbf{r}_j, s_j), \dots \right]$$

Using spinors to construct a many-body wave function, there is no longer a separation of up and down determinants. Only one large determinant built from all the single particular spinors.

Only need to add appropriate APIs to accumulate the spin derivatives alongside the standard gradients (used for VMC with drift and DMC algorithm).

Notice that the single particle spinors $\psi_i(\mathbf{r}_j, s_j) = \phi_i^{\uparrow}(\mathbf{r}_j)e^{is_j} + \phi_i^{\downarrow}(\mathbf{r}_j)e^{-is_j}$ are inherently complex, and therefore the full wave function is complex. For complex wave functions, the normal fixed-node approximation doesn't apply

MANY-BODY WAVE FUNCTION

$$\Psi_T(\mathbf{R}) = e^{J(\mathbf{R})} \sum_{\alpha} c_{\alpha} \det_{\alpha}^{\uparrow} \left[\dots, \phi_i^{\uparrow}(\mathbf{r}_j), \dots \right] \det_{\alpha}^{\downarrow} \left[\dots, \phi_k^{\downarrow}(\mathbf{r}_l), \dots \right]$$

$N_{\uparrow} \times N_{\uparrow}$ $N_{\downarrow} \times N_{\downarrow}$ $N_{\uparrow} + N_{\downarrow} = N_{\text{el}}$



Change from orbitals $\{\phi_i^{\uparrow/\downarrow}(\mathbf{r}_j)\}$ to spinors $\{\psi_i(\mathbf{r}_j, s_j)\}$

$$\Psi_T(\mathbf{X}) = e^{J(\mathbf{R})} \sum_{\alpha} c_{\alpha} \det \left[\dots, \underbrace{\psi_i(\mathbf{r}_j, s_j)}_{N_{\text{el}} \times N_{\text{el}}}, \dots \right]$$

Using spinors to construct a many-body wave function, there is no longer a separation of up and down determinants. Only one large determinant built from all the single particular spinors.

Only need to add appropriate APIs to accumulate the spin derivatives alongside the standard gradients (used for VMC with drift and DMC algorithm).

Notice that the single particle spinors $\psi_i(\mathbf{r}_j, s_j) = \phi_i^{\uparrow}(\mathbf{r}_j)e^{is_j} + \phi_i^{\downarrow}(\mathbf{r}_j)e^{-is_j}$ are inherently complex, and therefore the full wave function is complex. For complex wave functions, the normal fixed-node approximation doesn't apply

FIXED-PHASE APPROXIMATION

For $\Psi(\mathbf{X}) \in \mathbb{C}$, the Fixed-Node approximation no longer applies

Let $\Psi(\mathbf{X}) = \rho(\mathbf{X}) \exp[i\Phi(\mathbf{X})]$, with *amplitude* ρ and *phase* Φ

Imaginary time Schrödinger equation \Rightarrow

$$\begin{aligned} -\frac{\partial \rho}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + V + \frac{1}{2} |\nabla \Phi|^2 \right] \rho && \text{Note that the amplitude is always positive, i.e. no sign structure} \\ -\frac{\partial \Phi}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + \frac{\nabla \rho \cdot \nabla}{\rho} \right] \Phi \end{aligned}$$

FIXED-PHASE APPROXIMATION

For $\Psi(\mathbf{X}) \in \mathbb{C}$, the Fixed-Node approximation no longer applies

Let $\Psi(\mathbf{X}) = \rho(\mathbf{X}) \exp [i\Phi(\mathbf{X})]$, with *amplitude* ρ and *phase* Φ

Imaginary time Schrödinger equation \Rightarrow

$$\begin{aligned} -\frac{\partial \rho}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + V + \frac{1}{2} |\nabla \Phi|^2 \right] \rho \\ -\frac{\partial \Phi}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + \frac{\nabla \rho \cdot \nabla}{\rho} \right] \Phi \end{aligned}$$

Note that the amplitude is always positive, i.e. no sign structure

$\rightarrow \Phi(\mathbf{X}, \tau) = \Phi_T(\mathbf{X})$

FIXED-PHASE APPROXIMATION

For $\Psi(\mathbf{X}) \in \mathbb{C}$, the Fixed-Node approximation no longer applies

Let $\Psi(\mathbf{X}) = \rho(\mathbf{X}) \exp[i\Phi(\mathbf{X})]$, with *amplitude* ρ and *phase* Φ

Imaginary time Schrödinger equation \Rightarrow

$$\begin{aligned} -\frac{\partial \rho}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + V + \frac{1}{2} |\nabla \Phi|^2 \right] \rho \\ -\frac{\partial \Phi}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + \frac{\nabla \rho \cdot \nabla}{\rho} \right] \Phi \end{aligned}$$

Note that the amplitude is always positive, i.e. no sign structure

$\rightarrow \Phi(\mathbf{X}, \tau) = \Phi_T(\mathbf{X})$

Fixed-Node

$$\Psi_T(\mathbf{X}) \in \mathbb{R}$$

$$\Rightarrow \Phi_T(\mathbf{X}) = \begin{cases} 0, & \Psi_T(\mathbf{X}) > 0 \\ \pi, & \Psi_T(\mathbf{X}) < 0 \end{cases}$$

Fixed-Phase

$$\Psi_T(\mathbf{X}) \in \mathbb{C}$$

$$\Rightarrow \Phi_T(\mathbf{X}) = \tan^{-1} \left[\frac{\Im(\Psi(\mathbf{X}))}{\Re(\Psi(\mathbf{X}))} \right]$$

FIXED-PHASE APPROXIMATION

For $\Psi(\mathbf{X}) \in \mathbb{C}$, the Fixed-Node approximation no longer applies

Let $\Psi(\mathbf{X}) = \rho(\mathbf{X}) \exp [i\Phi(\mathbf{X})]$, with *amplitude* ρ and *phase* Φ

Imaginary time
Schrödinger
equation \Rightarrow

$$\begin{aligned} -\frac{\partial \rho}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + V + \frac{1}{2} |\nabla \Phi|^2 \right] \rho \\ -\frac{\partial \Phi}{\partial \tau} &= \left[-\frac{1}{2} \nabla^2 + \frac{\nabla \rho \cdot \nabla}{\rho} \right] \Phi \end{aligned}$$

Note that the amplitude is always positive, i.e. no sign structure

$$\rightarrow \Phi(\mathbf{X}, \tau) = \Phi_T(\mathbf{X})$$

SOC calculations in QMCPACK will always be complex. Need to build with -D QMC_COMPLEX=1 and run qmcpack_complex

Fixed-Node

$$\Psi_T(\mathbf{X}) \in \mathbb{R}$$

$$\Rightarrow \Phi_T(\mathbf{X}) = \begin{cases} 0, & \Psi_T(\mathbf{X}) > 0 \\ \pi, & \Psi_T(\mathbf{X}) < 0 \end{cases}$$

Fixed-Phase

$$\Psi_T(\mathbf{X}) \in \mathbb{C}$$

$$\Rightarrow \Phi_T(\mathbf{X}) = \tan^{-1} \left[\frac{\Im(\Psi(\mathbf{X}))}{\Re(\Psi(\mathbf{X}))} \right]$$

DIFFUSION MONTE CARLO WITH DYNAMIC SPINS

$$|\Psi_0\rangle \propto \lim_{\tau \rightarrow \infty} \exp[-\tau \mathcal{H}] |\Psi_T\rangle \quad E_0 = \lim_{\tau \rightarrow \infty} \frac{\langle \Psi(\tau) | \mathcal{H} | \Psi_T \rangle}{\langle \Psi(\tau) | \Psi_T \rangle}$$

$$\langle \mathbf{X}' = (\mathbf{R}', \mathbf{S}') | \Psi^{(n+1)} \rangle = \int d\mathbf{X} G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \langle \mathbf{X} | \Psi^{(n)} \rangle$$

Imaginary time
evolution in integral
form

Modification to the Greens function

$$G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \propto G(\mathbf{R} \rightarrow \mathbf{R}'; \tau) \exp(-|\mathbf{S}' - \mathbf{S} - \tau_s \mathbf{v}_s(\mathbf{S})|^2 / 2\tau_s)$$



Standard Drift/Diffusion
and Branching Greens
Functions. Only acts on
particle positions



Drift/Diffusion Greens
function for the collective
spin variables

DIFFUSION MONTE CARLO WITH DYNAMIC SPINS

$$|\Psi_0\rangle \propto \lim_{\tau \rightarrow \infty} \exp[-\tau \mathcal{H}] |\Psi_T\rangle \quad E_0 = \lim_{\tau \rightarrow \infty} \frac{\langle \Psi(\tau) | \mathcal{H} | \Psi_T \rangle}{\langle \Psi(\tau) | \Psi_T \rangle}$$

$$\langle \mathbf{X}' = (\mathbf{R}', \mathbf{S}') | \Psi^{(n+1)} \rangle = \int d\mathbf{X} G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \langle \mathbf{X} | \Psi^{(n)} \rangle$$

Imaginary time
evolution in integral
form

Modification to the Greens function

$$G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \propto G(\mathbf{R} \rightarrow \mathbf{R}'; \tau) \exp(-|\mathbf{S}' - \mathbf{S} - \tau_s \mathbf{v}_s(\mathbf{S})|^2 / 2\tau_s)$$



Standard Drift/Diffusion
and Branching Greens
Functions. Only acts on
particle positions



Drift/Diffusion Greens
function for the collective
spin variables

To generate the spin dynamics, we introduce a “spin kinetic energy” to the Hamiltonian. This generates the same Green’s function as standard position drift/diffusion

$$H_{\text{DMC}} \rightarrow H_{\text{DMC}} + \sum_i T_i^s \quad T_i^s = -\frac{1}{2\mu_s} \left[\frac{\partial^2}{\partial s_i^2} + 1 \right]$$

By construction, this does not contribute to
the local energy since the spin kinetic energy
isn’t physical

$$\sum_i T_i^s \prod_j \psi_j(\mathbf{r}_i, s_i) = 0$$

DIFFUSION MONTE CARLO WITH DYNAMIC SPINS

$$|\Psi_0\rangle \propto \lim_{\tau \rightarrow \infty} \exp[-\tau \mathcal{H}] |\Psi_T\rangle \quad E_0 = \lim_{\tau \rightarrow \infty} \frac{\langle \Psi_0 | \mathcal{H} | \Psi_0 \rangle}{\langle \Psi_0 | \Psi_0 \rangle}$$

$$\langle \mathbf{X}' = (\mathbf{R}', \mathbf{S}') | \Psi^{(n+1)} \rangle = \int d\mathbf{X} G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \langle \mathbf{X} | \Psi^{(n)} \rangle$$

Imaginary time evolution in integral form

The spin mass becomes an convergence parameter similar to the timestep. Will discuss more later

Modification to the Greens function

$$G(\mathbf{X} \rightarrow \mathbf{X}'; \tau) \propto G(\mathbf{R} \rightarrow \mathbf{R}'; \tau) \exp(-|\mathbf{S}' - \mathbf{S} - \tau_s \mathbf{v}_s(\mathbf{S})|^2 / 2\tau_s)$$



Standard Drift/Diffusion and Branching Greens Functions. Only acts on particle positions



Drift/Diffusion Greens function for the collective spin variables

To generate the spin dynamics, we introduce a “spin kinetic energy” to the Hamiltonian. This generates the same Green’s function as standard position drift/diffusion

$$H_{\text{DMC}} \rightarrow H_{\text{DMC}} + \sum_i T_i^s \quad T_i^s = -\frac{1}{2\mu_s} \left[\frac{\partial^2}{\partial s_i^2} + 1 \right]$$

By construction, this does not contribute to the local energy since the spin kinetic energy isn’t physical

$$\sum_i T_i^s \prod_j \psi_j(\mathbf{r}_i, s_i) = 0$$

RELATIVISTIC EFFECTIVE CORE POTENTIALS

$$W^{\text{REP}} = \sum_{\ell,j,m_j} W_{\ell j}(r) |\ell j m_j\rangle \langle \ell j m_j|$$

Rather than work directly with the spin spherical harmonics projectors, in QMCPACK we decouple the ECP into an averaged relativistic effective potential (AREP) and a spin-orbit relativistic effective potential.

The AREP looks just like our normal pseudopotentials, and the SOREP contains the spin-orbit part of the Hamiltonian.

$$\begin{aligned} W^{\text{REP}} &= W^{\text{AREP}} + W^{\text{SOREP}} \\ W^{\text{AREP}} &= W_L(r) + \sum_{\ell,m} W_{\ell}^{\text{AREP}}(r) |\ell m\rangle \langle \ell m| \\ W^{\text{SOREP}} &= \sum_{\ell} W_{\ell}^{\text{SOREP}}(r) \sum_{m,m'} |\ell m\rangle \langle \ell m | \vec{\ell} \cdot \vec{s} | \ell m' \rangle \langle \ell m' | \end{aligned}$$

In QMCPACK, we include both the AREP and SOREP terms in the pseudopotential xml files

RELATIVISTIC EFFECTIVE CORE POTENTIALS

$$W^{\text{REP}} = \sum_{\ell,j,m_j} W_{\ell j}(r) |\ell j m_j\rangle \langle \ell j m_j|$$

Rather than work directly with the spin spherical harmonics projectors, in QMCPACK we decouple the ECP into an averaged relativistic effective potential (AREP) and a spin-orbit relativistic effective potential.

The AREP looks just like our normal pseudopotentials, and the SOREP contains the spin-orbit part of the Hamiltonian.

$$\begin{aligned} W^{\text{REP}} &= W^{\text{AREP}} + W^{\text{SOREP}} \\ W^{\text{AREP}} &= W_L(r) + \sum_{\ell,m} W_{\ell}^{\text{AREP}}(r) |\ell m\rangle \langle \ell m| \\ W^{\text{SOREP}} &= \sum_{\ell} W_{\ell}^{\text{SOREP}}(r) \sum_{m,m'} |\ell m\rangle \langle \ell m| \vec{\ell} \cdot \vec{s} |\ell m'\rangle \langle \ell m'| \end{aligned}$$

In QMCPACK, we include both the AREP and SOREP terms in the pseudopotential xml files

In the quantum chemistry community, REPs are typically specified as AREP + SOREP. For DFT potentials (Troullier-Martins, RKKJ, Kerker, etc.) these are typically specified as $W_{\ell j}^{\text{REP}}(r)$. There is a straightforward relation between the REP and AREP+SOREP

$$\begin{pmatrix} W_{\ell}^{\text{AREP}} \\ W_{\ell}^{\text{SOREP}} \end{pmatrix} = \begin{pmatrix} \frac{\ell+1}{2\ell+1} & \frac{\ell}{2\ell+1} \\ \frac{-2}{2\ell+1} & \frac{1}{2\ell+1} \end{pmatrix} \begin{pmatrix} W_{\ell, j=\ell+1/2}^{\text{REP}} \\ W_{\ell, j=\ell-1/2}^{\text{REP}} \end{pmatrix}$$

IMPLEMENTATION OF SPIN-ORBIT CONTRIBUTION

$$\frac{W^{\text{SO}}\Psi}{\Psi} = \sum_I \sum_i \int ds'_i \sum_{\alpha} w_{\alpha} \sum_{\ell} v_{\ell}^{\text{SO}}(r_{iI}) \frac{\Psi(\dots(\mathbf{q}_{iI}^{\alpha}, s'_i)\dots)}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \sum_{m,m'} Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell m'}(r_{iI} \mathbf{u}^{\alpha}) \langle \ell m s_i | \ell \cdot \mathbf{s} | \ell m' s'_i \rangle$$

$$\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$$

$$r_{iI} = |\mathbf{r}_{iI}|$$

$$\mathbf{q}_{iI}^{\alpha} = \mathbf{R}_I + r_{iI} \mathbf{u}^{\alpha}$$

This expression is evaluated in a similar way to normal pseudopotentials in QMC. We use quadrature points \mathbf{q} , and evaluate directly evaluate the wave function ratios. We use directly evaluate complex spherical harmonics rather than simplify into Legendre polynomials.

Current implementation is to do a Simpon's rule integration for the spin variable. This is inefficient and is a target for future optimization

$$\Lambda = \langle \ell m s_i | \ell \cdot \mathbf{s} | \ell m' s'_i \rangle = \langle \ell m | \ell_{\mu} | \ell m' \rangle \langle s_i | s_{\mu} | s'_i \rangle$$

$$\langle s_i | s_{\mu} | s'_i \rangle = \begin{cases} \cos(s_i + s'_i), & \mu = x \\ \sin(s_i + s'_i), & \mu = y \\ i \sin(s_i - s'_i), & \mu = z \end{cases}$$

$$\langle \ell m | \ell_{\mu} | \ell m' \rangle = \begin{cases} \frac{1}{2} \left(\sqrt{\ell(\ell+1) - m'(m'+1)} \delta_{m,m'+1} + \sqrt{\ell(\ell+1) - m'(m'-1)} \delta_{m,m'-1} \right), & \mu = x \\ \frac{i}{2} \left(\sqrt{\ell(\ell+1) - m'(m'-1)} \delta_{m,m'-1} - \sqrt{\ell(\ell+1) - m'(m'+1)} \delta_{m,m'+1} \right), & \mu = y \\ m' \delta_{m,m'}, & \mu = z \end{cases}$$

The angular momentum matrix elements are straightforward. The spin matrix elements are obtained by expressing the spin operators into our continuous representation.

DEVELOPMENT OF CORRELATION CONSISTENT ECPS WITH SOC

In addition to an accurate treatment of SOC within DMC, DMC is only as accurate as the quality of the effective Hamiltonian used. If the underlying pseudopotential doesn't reflect the physics of the all-electron atom, DMC won't be reliable.

Extend development of **ccECPs** to include spin-orbit terms.

First optimize the AREP portion, which includes relativistic terms

$$\mathcal{O}_{\text{AREP}}^2 = \sum_{s \in S} w_s (\Delta E_{\text{AE}}^{(s)} - \Delta E_{\text{ECP}}^{(s)})^2 + \sum_{i \in L} w_i (\epsilon_{\text{AE}}^{(i)} - \epsilon_{\text{ECP}}^{(i)})^2$$

Correlation Energy Consistency:
Use CCSD(T) with DKH Hamiltonian
to get scalar relativistic effects for
multiple atomic states

Partial Norm-conservation:
Single particle eigenvalues (HF), part of norm conservation objective function.
Found to be helpful in transferability.

Pseudopotential Library

A community website for pseudopotentials/effective core potentials developed for high accuracy correlated many-body methods such as quantum Monte Carlo and quantum chemistry.

$$\begin{aligned}
 W^{\text{REP}} &= W^{\text{AREP}} + W^{\text{SOREP}} \\
 W^{\text{AREP}} &= W_L(r) + \sum_{\ell,m} W_\ell^{\text{AREP}}(r) |\ell m\rangle\langle\ell m| \\
 W^{\text{SOREP}} &= \sum_\ell W_\ell^{\text{SOREP}}(r) \sum_{m,m'} |\ell m\rangle\langle\ell m| \vec{\ell} \cdot \vec{s} |\ell m'\rangle\langle\ell m'|
 \end{aligned}$$

DEVELOPMENT OF CORRELATION CONSISTENT ECPS WITH SOC

In addition to an accurate treatment of SOC within DMC, DMC is only as accurate as the quality of the effective Hamiltonian used. If the underlying pseudopotential doesn't reflect the physics of the all-electron atom, DMC won't be reliable.

Extend development of **ccECPs** to include spin-orbit terms.

Next optimize the SOREP portion, which introduces the Spin-orbit correction

$$\mathcal{O}_{\text{SOREP}}^2 = \sum_{s \in S'} w_s (\Delta E_{\text{AE}}^{(s)} - \Delta E_{\text{ECP}}^{(s)})^2 + \sum_{m \in M} w_m (\Delta E_{\text{AE}}^{(m)} - \Delta E_{\text{ECP}}^{(m)})^2$$

Energy differences with respect to ground state of various charge states (EA, IP, IPI, IPII, IPIII, etc) using X2C Hamiltonian and COSCI

Energy differences of various $^{2S+1}L_J$ multiplets using X2C Hamiltonian at the COSCI level.

Pseudopotential Library

A community website for pseudopotentials/effective core potentials developed for high accuracy correlated many-body methods such as quantum Monte Carlo and quantum chemistry.

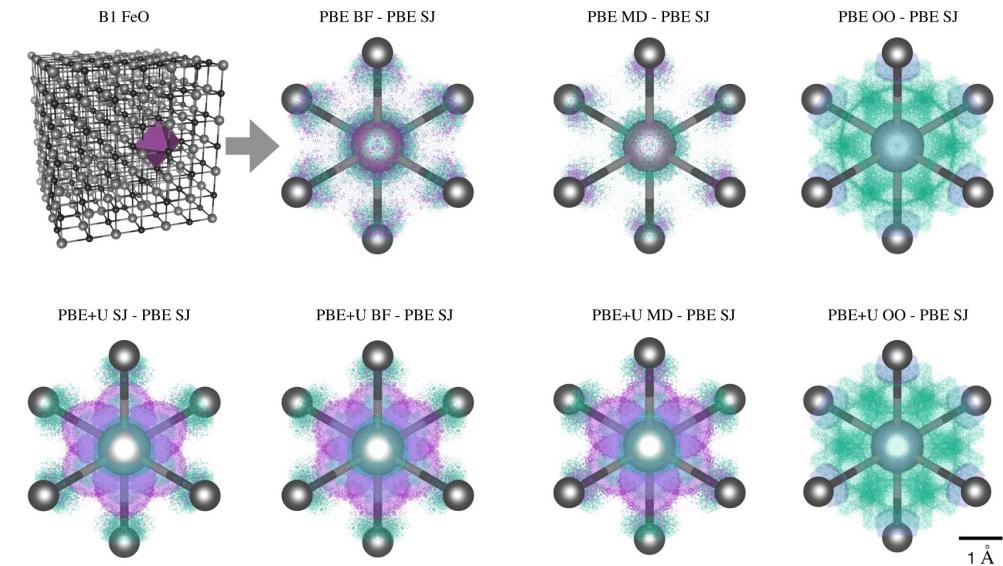
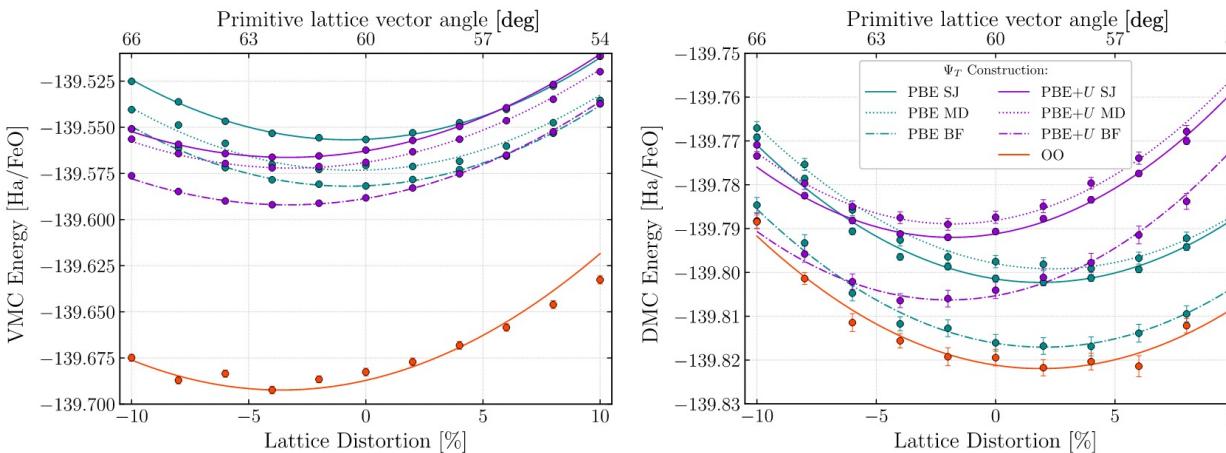
$$\begin{aligned} W^{\text{REP}} &= W^{\text{AREP}} + W^{\text{SOREP}} \\ W^{\text{AREP}} &= W_L(r) + \sum_{\ell,m} W_\ell^{\text{AREP}}(r) |\ell m\rangle\langle\ell m| \\ W^{\text{SOREP}} &= \sum_\ell W_\ell^{\text{SOREP}}(r) \sum_{m,m'} |\ell m\rangle\langle\ell m| \vec{\ell} \cdot \vec{s} |\ell m'\rangle\langle\ell m'| \end{aligned}$$

Reach out to Ben Kincaid for new ccECPs with SOC.

Also reach out for help
converting existing
pseudopotentials to XML/UPF
format

OPTIMIZATION OF SPINORS TO REDUCE FIXED-PHASE ERROR

Orbital optimization is a key capability for solids to reduce the nodal surface error. This has already been demonstrated in QMCPACK for FeO (see J.P. Townsend *et al.*, PRB, **102**, 155151 (2020))



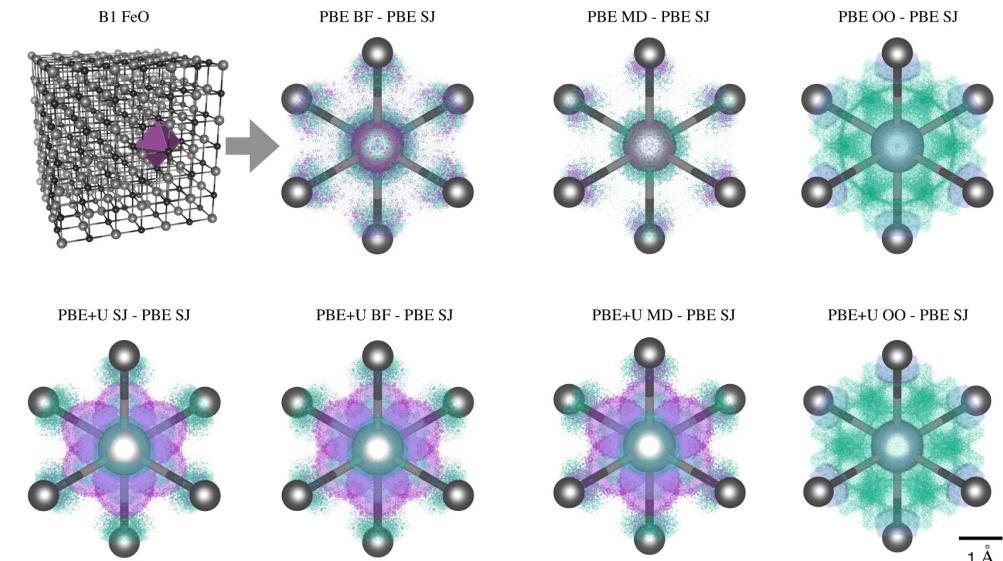
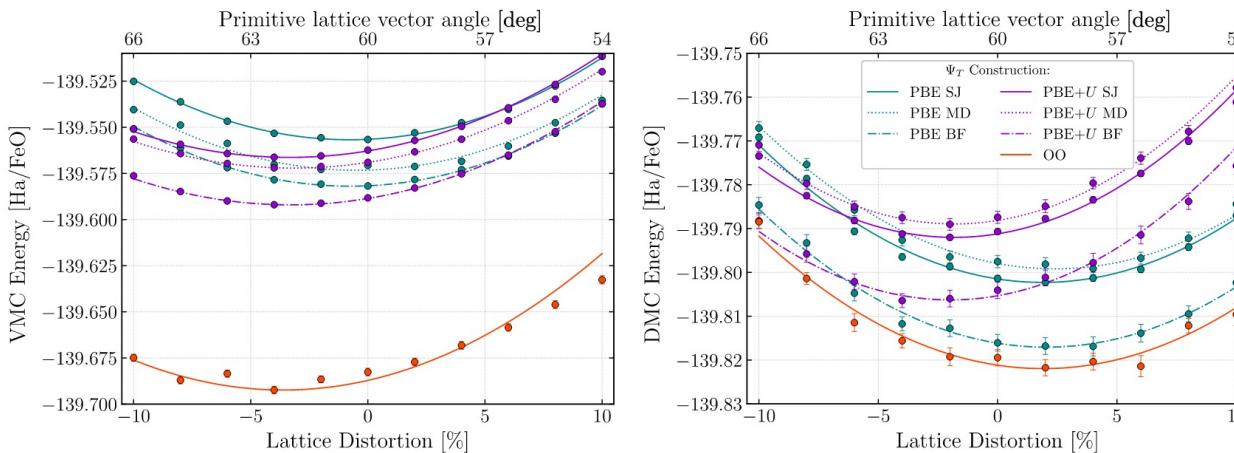
These calculations were performed using a development branch from Berkley which has not been fully enabled into mainline QMCPACK. This is a key capability for systematic improvability

$$|\Psi(\alpha, \mathbf{c}, \kappa)\rangle = \hat{J}(\alpha) e^{\hat{\kappa}(\kappa)} \sum_{I=1}^{N_{\text{CSF}}} c_I |C_I\rangle,$$

$$|\phi_k\rangle = e^{\hat{\kappa}(\kappa)} |\phi_k^0\rangle = \sum_l (e^\kappa)_{kl} |\phi_l^0\rangle,$$

OPTIMIZATION OF SPINORS TO REDUCE FIXED-PHASE ERROR

Orbital optimization is a key capability for solids to reduce the nodal surface error. This has already been demonstrated in QMCPACK for FeO (see J.P. Townsend *et al.*, PRB, **102**, 155151 (2020))



Orbital rotation for our standard SPOSets (LCAOs, splines) has already been implemented.

Generalization to spinors is trivial. All that is needed is to add an API for the apply rotation to the SpinorSet. Internally, the implementation just applies the *same* rotation matrix to each “up” and “down” SPOSet.

$$\begin{aligned}
 |\tilde{\psi}_i\rangle &= \sum_j \alpha_{ij} |\psi_j\rangle \\
 &= \sum_j \alpha_{ij} \left(|\phi_j^\uparrow\rangle |\chi^\uparrow\rangle + |\phi_j^\downarrow\rangle |\chi^\downarrow\rangle \right) \\
 &= \sum_j \left(\alpha_{ij} |\phi_j^\uparrow\rangle \right) |\chi^\uparrow\rangle + \left(\alpha_{ij} |\phi_j^\downarrow\rangle \right) |\chi^\downarrow\rangle
 \end{aligned}$$

OTHER OBSERVABLES

Magnetization Density: Measures spin vector per unit volume across the simulation cell.

Useful for studying noncollinear spin, spin spirals, etc.

FT of this gives the magnetic form factor which is directly comparable to neutron diffraction expts.

$$\mathbf{m}_c = \int d\mathbf{X} |\Psi(\mathbf{X})|^2 \int_{\Omega_c} d\mathbf{r} \sum_i \delta(\mathbf{r} - \hat{\mathbf{r}}_i) \int_0^{2\pi} \frac{ds'_i}{2\pi} \frac{\Psi(\dots \mathbf{r}_i s'_i \dots)}{\Psi(\dots \mathbf{r}_i s_i \dots)} \langle s_i | \hat{\sigma} | s'_i \rangle$$

$$\begin{aligned} n(\mathbf{r}) &= n^{\uparrow\uparrow}(\mathbf{r}) + n^{\downarrow\downarrow}(\mathbf{r}) \\ m_x(\mathbf{r}) &= n^{\uparrow\downarrow}(\mathbf{r}) + n^{\downarrow\uparrow}(\mathbf{r}) \\ m_y(\mathbf{r}) &= i(n^{\uparrow\downarrow}(\mathbf{r}) - n^{\downarrow\uparrow}(\mathbf{r})) \\ m_z(\mathbf{r}) &= n^{\uparrow\uparrow}(\mathbf{r}) - n^{\downarrow\downarrow}(\mathbf{r}) \end{aligned}$$

Reduced Density Matrices:

Currently in QMCPACK, we have a working version of the one-body reduced density matrix for normal wave functions.

Actively working on adapting it to work with out spinor representation.

Plans to utilize this for a variety of applications, including some topological materials.

Also have plans to implement the 2-RDM, which will enable a wide variety of applications, including downfolding to model Hamiltonians

$$n_1(i, j) = \int d\mathbf{X} \Psi_{\text{FP}}^*(\mathbf{X}) \Psi_T(\mathbf{X}) \left[\sum_n \int d\mathbf{x}'_n \frac{\Psi_T(\mathbf{x}'_n, \mathbf{X}_n)}{\Psi_T(\mathbf{x}_n, \mathbf{X}_n)} \psi_i^*(\mathbf{x}'_n) \psi_j(\mathbf{x}_n) \right]$$

TUTORIAL: SOC FOR SOLID TIN



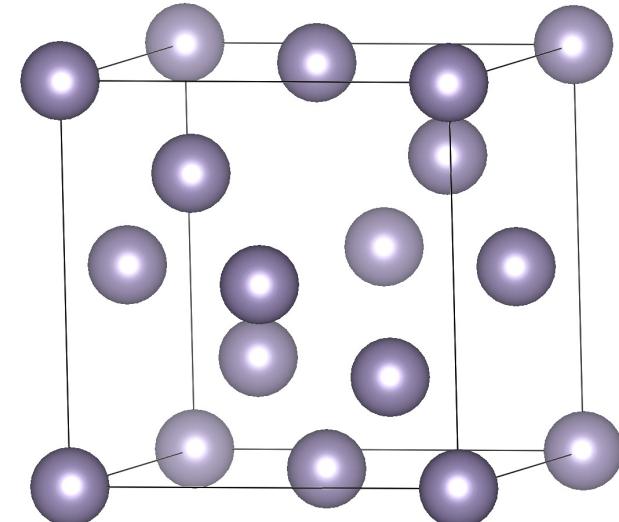
Sn

ccECP

... .

Sn.ccECP.AREP.upf.
Sn.ccECP.SOREP.upf
Sn.ccECP.AREP.xml
Sn.ccECP.SOREP.xml

- Averaged Rel. for QE
- SOC for QE
- Average Rel. for QMCPACK
- SOC for OMCPACK



- Simple example demonstrating how to drive SOC calculations in QMCPACK using Nexus
 - Will go through standard workflow of running QE, wave function conversion, optimization, and VMC/DMC using batched algorithm
 - Note: If interested in running molecular systems, see tutorial from [2021 QMCPACK Workshop](#)

PHYSICAL SYSTEM - NEXUS

```
system = generate_physical_system(
    units      = 'A',
    axes       = ''
        0.0000000000          3.2456000031          3.2456000031
        3.2456000031          0.0000000000          3.2456000031
        3.2456000031          3.2456000031          0.0000000000
    '',
    elem_pos  = ''
        Sn  0.0000000000          0.0000000000          0.0000000000
        Sn  1.622799993           1.622799993           1.622799993
    '',
    kgrid     = (1,1,1),
    Sn       = 4,
)
```

Nothing special needed in the system to specify SOC/Spinor calculation
Set up as you normally would

SCF STEP WITH QE

```

pwscf_common_arguments = obj(
    identifier = 'scf',
    input_type = 'generic',
    job        = job(cores=8, app=pwscf),
    calculation = 'scf',
    ecutwfc    = 200,
    conv_thr    = 1e-08,
    system      = system,
    kgrid       = (4,4,4),
    kshift      = (1,1,1),
)
pwscf_woSOC_arguments = obj(
    path      = 'scf/woSOC',
    pseudos  = ['Sn.ccECP.AREP.upf'],
)
pwscf_wSOC_arguments = obj(
    path      = 'scf/wSOC',
    pseudos  = ['Sn.ccECP.SOREP.upf'],
    noncolin = True,
    lspinorb = True,
)

```

- Entire soc.py file is set up to show the *common* arguments between SOC and non-SOC, as well as the specific arguments needed for SOC and non SOC respectively
- For QE without SOC, must use the AREP UPF i.e. pseudopotential file without SOC terms.
- To enable SOC in QE, you need **both** a SOC pseudopotential file (i.e. Sn.ccECP.SOREP.upf) as well as the keywords noncolin and lspinorb

```

if withSOC:
    pwscf_args = obj(**pwscf_common_arguments, **pwscf_wSOC_arguments)
else:
    pwscf_args = obj(**pwscf_common_arguments, **pwscf_woSOC_arguments)
scf = generate_pwscf(**pwscf_args)

```

WAVEFUNCTION CONVERSION

- Currently, pw2qmcpack.x does **not** support spinor wave functions.
- Need to use convertpw4qmc executable for now, built in QMCPACK. To use, QE needs to be compiled with HDF5 support (i.e add -D QE_ENABLE_HDF5=1 to cmake). Then simply run

```
> /path/to/convertpw4qmc pwscf_output/pwscf.save/data-file-schema.xml
```

from inside the QE run directory

- Also supported in nexus

```
conv = generate_convertpw4qmc(  
    identifier = 'conv',  
    path       = 'nscf/wSOC',  
    job        = job(cores=1, app=cpw4qmc),  
    dependencies = (nscf, 'orbitals'),  
)
```

SPIN ORBIT QMC

- To indicate a spinor calculation, the most important change in the input file is in the specification of the particle set

```
<particleset name="e" random="yes" randomsrc="ion0" spinor="yes">
    <group name="u" size="10" mass="1.0">
        <parameter name="charge"          > -1           </parameter>
        <parameter name="mass"            > 1.0          </parameter>
    </group>
</particleset>
```

- Only need one type of electron group, since there is no longer a distinction between 'u' and 'd'
- Just add *spinor='yes'* tag

SPIN ORBIT QMC

- The wave function specification needs to point to a spinor HDF5 file from convertpw4qmc

```
<qmcsystem>
  <wavefunction name="psi0" target="e">
    <sposet_builder name="spo_builder" type="bspline" href="eshdf.h5"
      tilematrix="100010001" twistnum="0" source="ion0" size="10">
      <sposet type="bspline" name="myspo" size="10">
        <occupation mode="ground"/>
      </sposet>
    </sposet_builder>
    <determinantset>
      <slaterdeterminant>
        <determinant id="det" group="u" sposet="myspo" size="10"/>
      </slaterdeterminant>
    </determinantset>
    <jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
      <correlation elementType="0" size="8" cusp="0.0">
        <coefficients id="e0" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
    <jastrow type="Two-Body" name="J2" function="bspline" print="yes">
      <correlation speciesA="u" speciesB="u" size="8" cusp="-0.5">
        <coefficients id="uu" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
  </wavefunction>
</qmcsystem>
```

SPIN ORBIT QMC

- The wave function specification needs to point to a spinor HDF5 file from convertpw4qmc
- Only need one determinant, since there is not a decoupled up/down determinant anymore

```

<qmcsystem>
  <wavefunction name="psi0" target="e">
    <sposet_builder name="spo_builder" type="bspline" href="eshdf.h5"
      tilematrix="100010001" twistnum="0" source="ion0" size="10">
      <sposet type="bspline" name="myspo" size="10">
        <occupation mode="ground"/>
      </sposet>
    </sposet_builder>
    <determinantset>
      <slaterdeterminant>
        <determinant id="det" group="u" sposet="myspo" size="10"/>
      </slaterdeterminant>
    </determinantset>
    <jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
      <correlation elementType="0" size="8" cusp="0.0">
        <coefficients id="e0" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
    <jastrow type="Two-Body" name="J2" function="bspline" print="yes">
      <correlation speciesA="u" speciesB="u" size="8" cusp="-0.5">
        <coefficients id="uu" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
  </wavefunction>
</qmcsystem>
```

SPIN ORBIT QMC

- The wave function specification needs to point to a spinor HDF5 file from convertpw4qmc
- Only need one determinant, since there is not a decoupled up/down determinant anymore
- For two-body Jastrow, we also only need one between our single electron group.
- Need to adjust cusp to -0.5 since the same-spin Jastrow is technically being used.

```
<qmcsystem>
  <wavefunction name="psi0" target="e">
    <sposet_builder name="spo_builder" type="bspline" href="eshdf.h5"
      tilematrix="100010001" twistnum="0" source="ion0" size="10">
      <sposet type="bspline" name="myspo" size="10">
        <occupation mode="ground"/>
      </sposet>
    </sposet_builder>
    <determinantset>
      <slaterdeterminant>
        <determinant id="det" group="u" sposet="myspo" size="10"/>
      </slaterdeterminant>
    </determinantset>
    <jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
      <correlation elementType="0" size="8" cusp="0.0">
        <coefficients id="e0" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
    <jastrow type="Two-Body" name="J2" function="bspline" print="yes">
      <correlation speciesA="u" speciesB="u" size="8" cusp="-0.5">
        <coefficients id="uu" type="Array">
        </coefficients>
      </correlation>
    </jastrow>
  </wavefunction>
</qmcsystem>
```

See discussion of cusps in [doi:10.1063/1.4954726](https://doi.org/10.1063/1.4954726)

SPIN ORBIT QMC

When driving with nexus,
we only need to add the
'spinor = yes' argument

That takes care of the
cusp single determinant,
the number of Jastrow
factors, the cusps in the
Jastrow, and the particle
set input for you

```
opt_common_arguments = obj(  
    identifier      = 'opt',  
    job             = qmcjob,  
    input_type      = 'basic',  
    system          = system,  
    J1              = True,  
    J2              = True,  
    driver          = 'batched',  
    dependencies    = (conv, 'orbitals'),  
    calculations    = [loop(**loop_arguments  
)  
opt_woSOC_arguments = obj(  
    path     = 'optJ2_batched/woSOC',  
    pseudos = ['Sn.ccECP.AREP.xml'],  
)  
opt_wSOC_arguments = obj(  
    path     = 'optJ2_batched/wSOC',  
    pseudos = ['Sn.ccECP.SOREP.xml'],  
    spinor   = 'yes',  
)
```

SPIN ORBIT QMC

For VMC/DMC with dynamic spins, we need to use a SOC xml pseudopotential, and also use the spinor=yes flag.

The other important input parameter is the spin_mass, which sets the rate of sampling of the spins relative to the spatial timestep.

VMC spin sampling can use the default spin_mass of 1

For DMC, the correct limit is $\mu_s \rightarrow \infty$

$$H_{\text{DMC}} \rightarrow H_{\text{DMC}} + \sum_i T_i^s$$

$$T_i^s = -\frac{1}{2\mu_s} \left[\frac{\partial^2}{\partial s_i^2} + 1 \right]$$

```
def createSpinorQMC(spinmass):
    #for VMC, just using default spinmass so not setting it
    vmc_input = obj(walkers_per_rank = 56,
                    steps          = 50,
                    blocks         = 30,
                    usedrift       = True,
                    timestep       = 0.75
                )
    dmc_input = obj(nonlocalmoves   = False,
                    blocks          = 30,
                    walkers_per_rank = 56,
                    spin_mass        = spinmass
                )
    qmc_arguments = obj(
        path           = 'qmc_batched/wSOC/spinmass_{}'.format(spinmass),
        identifier     = 'qmcc',
        job            = qmcjob,
        input_type     = 'basic',
        system          = system,
        J2             = True,
        qmc            = 'vmc',
        driver          = 'batched',
        spinor          = 'yes',
        dependencies   = [(conv,'orbitals'),
                           (opt,'jastrow')],
        pseudos         = ['Sn.ccECP.SOREP.xml'],
        calculations   = [vmc(**vmc_input),
                          dmc(timestep = 0.1600, steps = 100, **dmc_input),
                          dmc(timestep = 0.0400, steps = 25, **dmc_input),
                          dmc(timestep = 0.0100, steps = 100, **dmc_input),
                          dmc(timestep = 0.0025, steps = 400, **dmc_input)]
    )
    return generate_qmcpack(**qmc_arguments)
```

SPIN ORBIT QMC

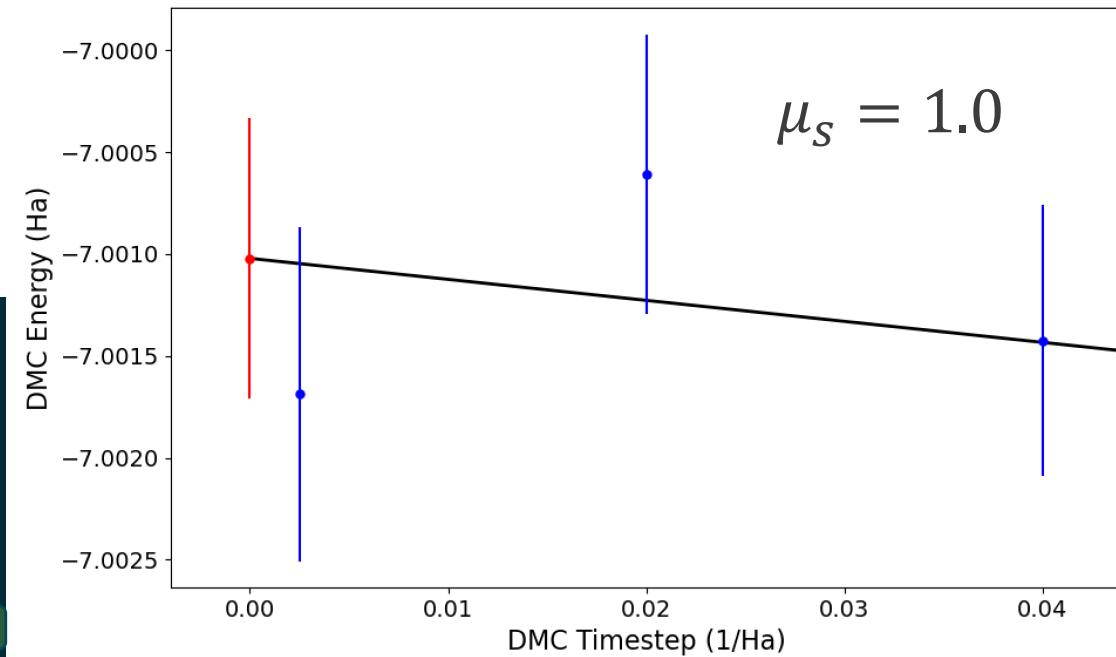
- We scan the spin masses to investigate the impact on the DMC energy
- For each spin mass, we do a normal timestep convergence
- Contribution of SOECP is incorporated into the total energy, and can be found from *qmca*

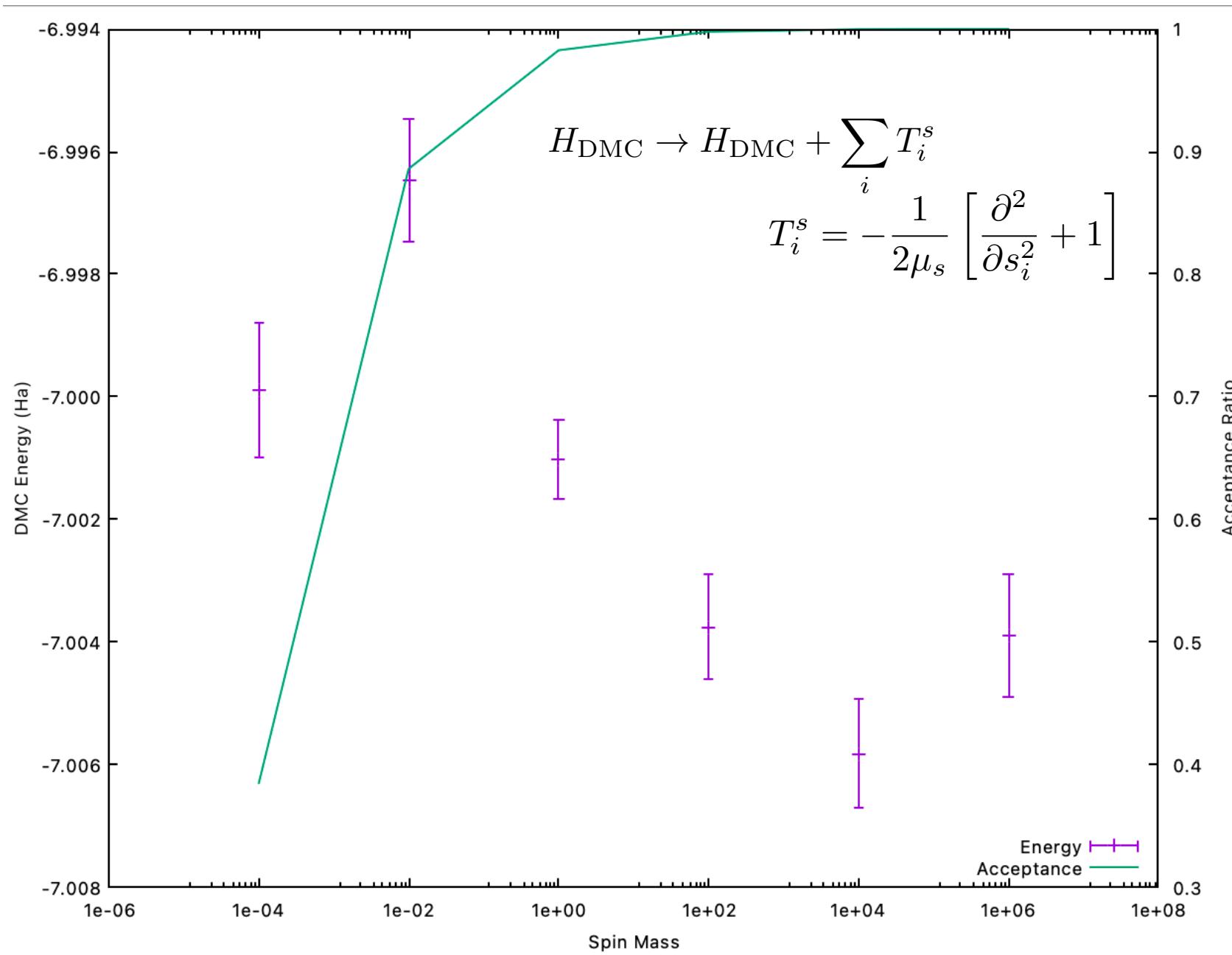
opt	series	9
LocalEnergy	=	-6.98551 +/- 0.00048
Variance	=	0.07503 +/- 0.00041
Kinetic	=	2.6999 +/- 0.0029
LocalPotential	=	-9.6854 +/- 0.0029
ElecElec	=	-1.7743 +/- 0.0017
LocaleCP	=	-1.5283 +/- 0.0035
NonLocaleCP	=	0.6533 +/- 0.0037
IonIon	=	-7.03 +/- 0.00
LocalEnergy_sq	=	48.8724 +/- 0.0068
SOECP	=	-0.00976 +/- 0.00013
BlockWeight	=	13440.00 +/- 0.00
BlockCPU	=	2.41122 +/- 0.00021
AcceptRatio	=	0.75316 +/- 0.00020
Efficiency	=	4879.01 +/- 0.00
TotalTime	=	241.12 +/- 0.00
TotalSamples	=	1344000 +/- 0

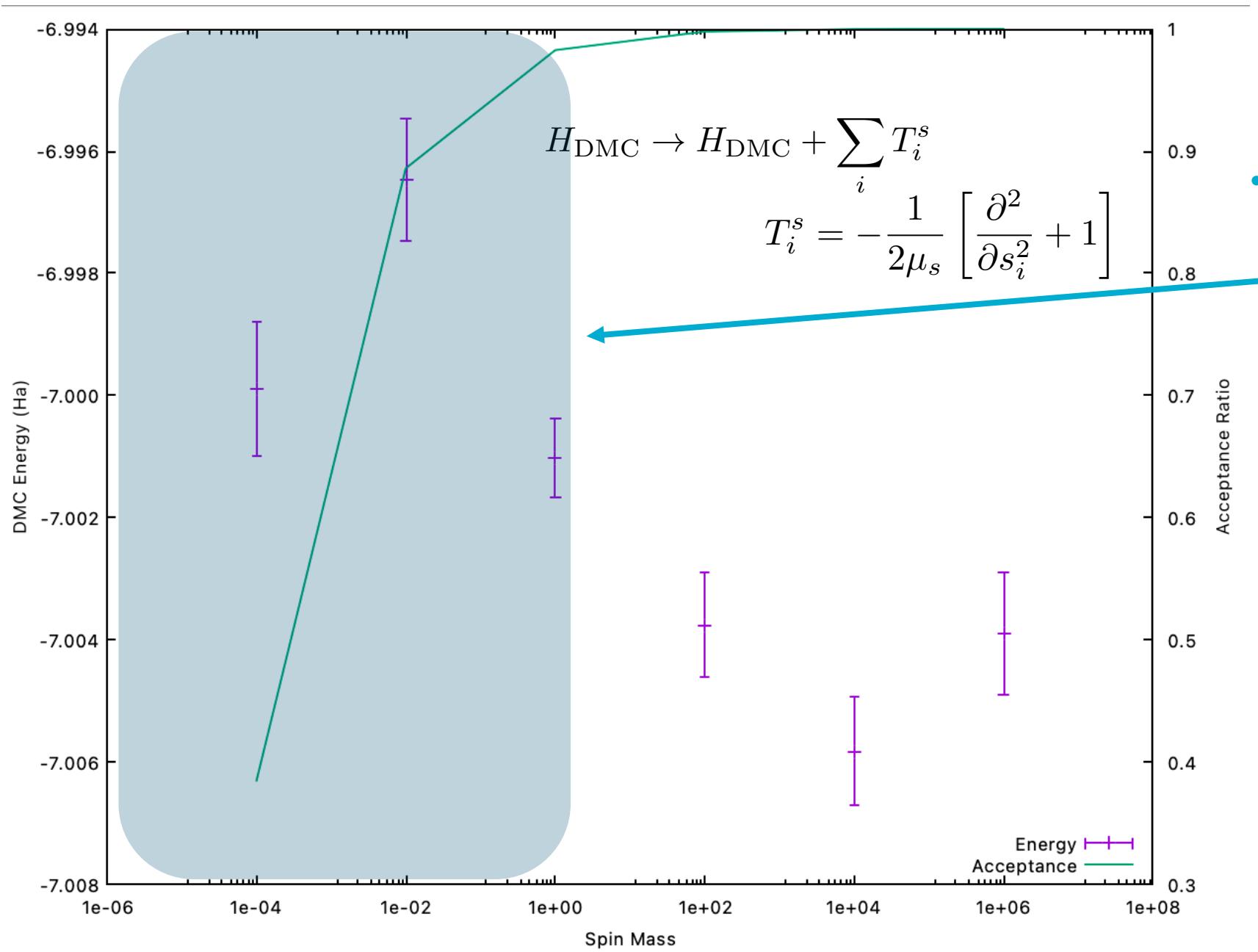
```

if withSOC:
    for spinmass in [1e-04, 1e-02, 1e00, 1e02, 1e04, 1e06]:
        qmc = createSpinorQMC(spinmass)
else:
    qmc = createQMC()

```







- Too small of a spin mass in DMC ($\mu_s < 1$) produce higher energies, and it can also be seen as a very poor acceptance ratio in DMC

- Too small of a spin mass in DMC ($\mu_s < 1$) produce higher energies, and it can also be seen as a very poor acceptance ratio in DMC
- Once the spin mass is large enough ($\mu_s > 100$), energy is essentially converged and acceptance ratio $\alpha > 0.99$
- Should be checked!

$$H_{\text{DMC}} \rightarrow H_{\text{DMC}} + \sum_i T_i^s$$
$$T_i^s = \frac{1}{2\mu_s} \left[\frac{\partial^2}{\partial s_i^2} + 1 \right]$$

- Too small of a spin mass in DMC ($\mu_s < 1$) produce higher energies, and it can also be seen as a very poor acceptance ratio in DMC
- Once the spin mass is large enough ($\mu_s > 100$), energy is essentially converged and acceptance ratio $\alpha > 0.99$
- Should be checked!

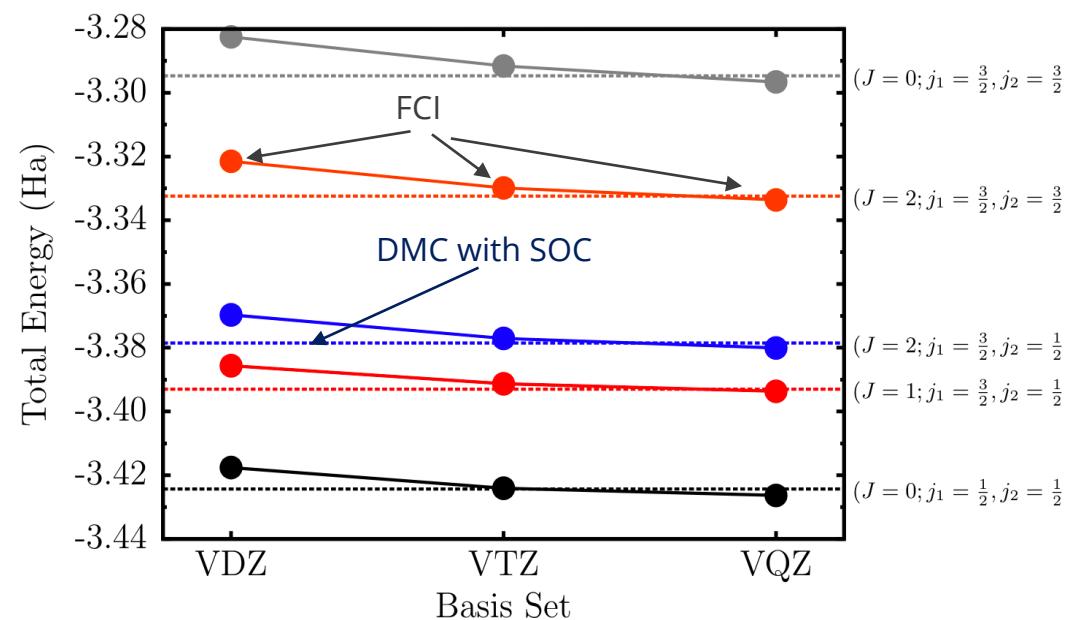


SUMMARY

- ❖ Spin-Orbit interactions are critical to be able to accurately describe heavy element materials, topological properties, etc. Accurate treatments of both electron correlation and relativistic effects can be obtained using Quantum Monte Carlo methods
- ❖ Variational and Diffusion Monte Carlo with dynamic spin and SOC has been implemented in QMCPACK, and is already incorporated into the batched code.
- ❖ Recent Nexus support enables simple SOC calculations with minimal changes
- ❖ Beyond energy observables are already supported like magnetization density and the 1RDM in spinor basis.
- ❖ Preliminary support for orbital optimization has also been implemented (see orbital optimization talk later)

EXAMPLES FOR ATOMIC / MOLECULAR SYSTEMS

Total Energies of Pb with Large Core REP

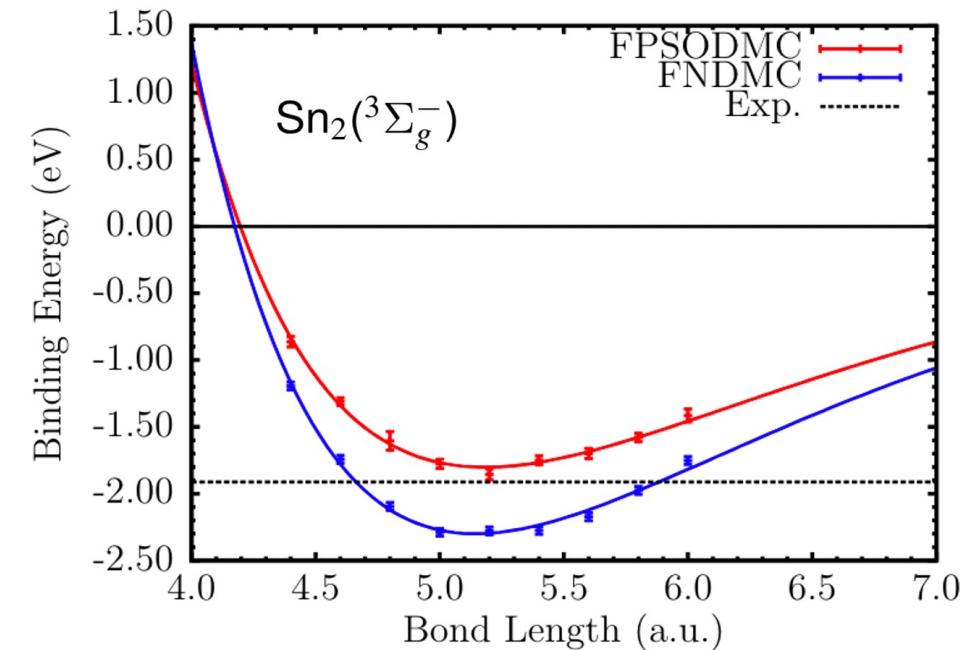


C.A. Melton *et al.*, PRA, 93, 2016

DMC comparison to exact FCI calculations for the same Hamiltonian (i.e. same ECP for both FCI and DMC)

DMC is able to reproduce the total energies and splittings for **all** excited states.

Binding energy of Sn_2 impact of SOC



C.A. Melton *et al.*, JCP, 144, 2016

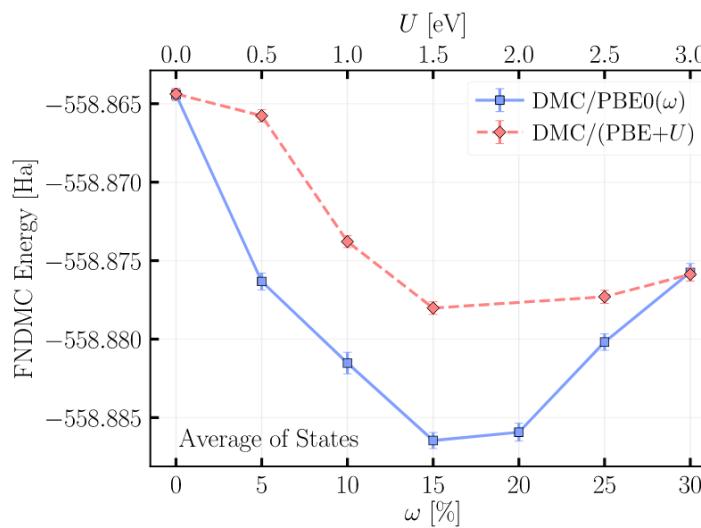
Comparison between DMC both with and without SOC.

Note that standard DMC would overbind the Sn_2 molecule by roughly 0.5 eV if SOC is neglected. DMC with SOC agrees with experiment

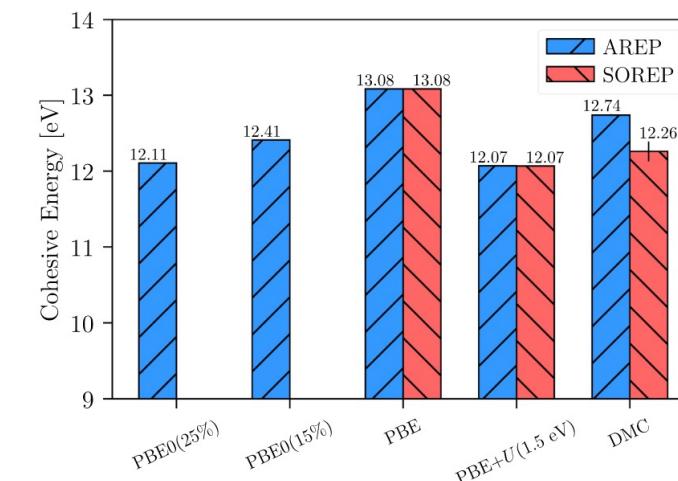
FIRST APPLICATION TO EXTENDED SYSTEMS

DMC with SOC was used to study the layered $\alpha\text{-RuCl}_3$

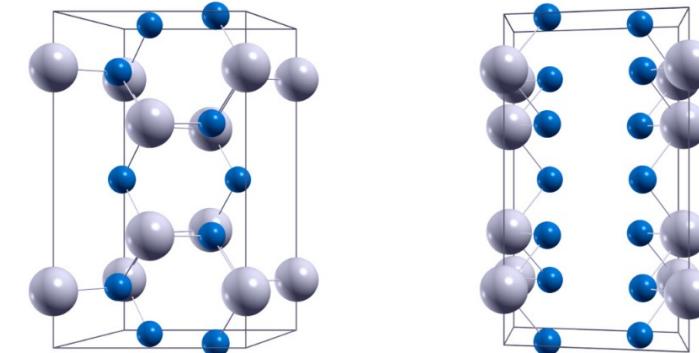
Stuttgart relativistic ECPs for Ru and ccECP for Cl. Used PBE+U and PBE0 to generate trial wave functions from Quantum Espresso.



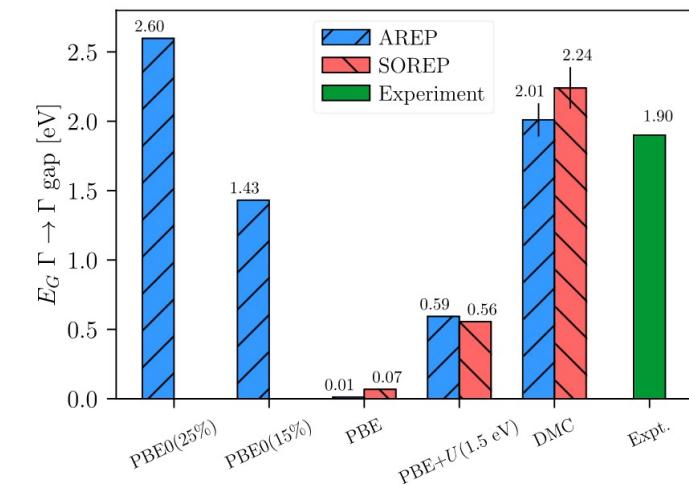
E_N , $E_{N\pm 1}$, $E^{\Gamma \rightarrow \Gamma}$ states considered.
Averaged to find best compromise in nodal surface to calculate gaps



DMC find sizeable correction to cohesive energy and optical gap coming from SOC. Better agreement with experiment than various DFT methods



Annaberdiyev, G. et al., Phys. Rev. B, 106, 075127 (2022)



OPTIMIZATION OF THE SPIN-ORBIT HAMILTONIAN EVALUATION

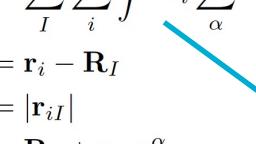
Current implementation of SOC

Hamiltonian is inefficient. We evaluate the expression (right) and replace the spin integral with a Simpson's rule.

Essentially, for every spin point we also have to do the quadrature point sum for the angular integral.

$$\frac{W^{\text{SO}}\Psi}{\Psi} = \sum_I \sum_i \int ds'_i \sum_{\alpha} w_{\alpha} \sum_{\ell} v_{\ell}^{\text{SO}}(r_{iI}) \frac{\Psi(\dots(\mathbf{q}_{iI}^{\alpha}, s'_i)\dots)}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \sum_{m,m'} Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell,m'}(r_{iI} \mathbf{u}^{\alpha}) \langle \ell m s_i | \ell \cdot \mathbf{s} | \ell m' s'_i \rangle$$

$\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$
 $r_{iI} = |\mathbf{r}_{iI}|$
 $\mathbf{q}_{iI}^{\alpha} = \mathbf{R}_I + r_{iI} \mathbf{u}^{\alpha}$



$$\int_0^{2\pi} ds f(s) \approx \frac{2\pi}{3N} \sum_{i=1}^{N/2} [f(s_{2i-2}) + 4f(s_{2i-1}) + f(s_{2i})]$$

OPTIMIZATION OF THE SPIN-ORBIT HAMILTONIAN EVALUATION

Current implementation of SOC

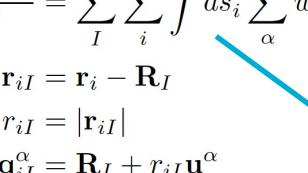
Hamiltonian is inefficient. We evaluate the expression (right) and replace the spin integral with a Simpson's rule.

Essentially, for every spin point we also have to do the quadrature point sum for the angular integral.

More efficient: Exploit the fact that Ψ can be written as a (multi-) determinant of spinors with jastrows. Then we can express WF ratio using inverse transpose slater matrix and the single particle spinors

$$\frac{W^{\text{SO}}\Psi}{\Psi} = \sum_I \sum_i \int ds'_i \sum_\alpha w_\alpha \sum_\ell v_\ell^{\text{SO}}(r_{iI}) \frac{\Psi(\dots(\mathbf{q}_{iI}^\alpha, s'_i)\dots)}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \sum_{m,m'} Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell,m'}(r_{iI} \mathbf{u}^\alpha) \langle \ell m s_i | \ell \cdot \mathbf{s} | \ell m' s'_i \rangle$$

$\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$
 $r_{iI} = |\mathbf{r}_{iI}|$
 $\mathbf{q}_{iI}^\alpha = \mathbf{R}_I + r_{iI} \mathbf{u}^\alpha$



$$\int_0^{2\pi} ds f(s) \approx \frac{2\pi}{3N} \sum_{i=1}^{N/2} [f(s_{2i-2}) + 4f(s_{2i-1}) + f(s_{2i})]$$

$$\frac{\Psi(\dots(\mathbf{q}_{iI}^\alpha), s'_i)\dots}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \rightarrow \frac{J(\dots \mathbf{q}_{iI}^\alpha \dots)}{J(\dots \mathbf{r}_i \dots)} \sum_n C_{ni}(\mathbf{r}_i, s_i) \psi_n(\mathbf{r}_{iI}^\alpha, s'_i)$$

$$\sum_{i,I} \sum_\alpha w_\alpha \sum_{\ell,m,m'} v_\ell^{\text{SO}}(r_{iI}) Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell,m'}(r_{iI} \mathbf{u}^\alpha) \frac{J(\dots \mathbf{q}_{iI}^\alpha \dots)}{J(\dots \mathbf{r}_i \dots)} \sum_n C_{ni}(\mathbf{r}_i, s_i) \int \frac{ds'_i}{2\pi} \psi_n(\mathbf{q}_{iI}^\alpha, s'_i) \Lambda_{\ell,m,m',s_i,s'_i}$$

OPTIMIZATION OF THE SPIN-ORBIT HAMILTONIAN EVALUATION

Current implementation of SOC

Hamiltonian is inefficient. We evaluate the expression (right) and replace the spin integral with a Simpson's rule.

Essentially, for every spin point we also have to do the quadrature point sum for the angular integral.

More efficient: Exploit the fact that Ψ can be written as a (multi-) determinant of spinors with jastrows. Then we can express WF ratio using inverse transpose slater matrix and the single particle spinors

By expanding the spinors as into the spin components, we can analytically integrate the spin integral and removes the unnecessary numerical integration.

In QMCPACK, historically the TrialWaveFunction interface was designed to be form agnostic, i.e. not straightforward to exploit the properties of (multi)-Slater Determinants and get access to the orbitals from the interface. This limitation prevents this optimization, but will be addressed in the future.

$$\frac{W^{\text{SO}}\Psi}{\Psi} = \sum_I \sum_i \int ds'_i \sum_\alpha w_\alpha \sum_\ell v_\ell^{\text{SO}}(r_{iI}) \frac{\Psi(\dots(\mathbf{q}_{iI}^\alpha, s'_i)\dots)}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \sum_{m,m'} Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell,m'}(r_{iI} \mathbf{u}^\alpha) \langle \ell m s_i | \ell \cdot \mathbf{s} | \ell m' s'_i \rangle$$

$\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$
 $r_{iI} = |\mathbf{r}_{iI}|$
 $\mathbf{q}_{iI}^\alpha = \mathbf{R}_I + r_{iI} \mathbf{u}^\alpha$

$$\int_0^{2\pi} ds f(s) \approx \frac{2\pi}{3N} \sum_{i=1}^{N/2} [f(s_{2i-2}) + 4f(s_{2i-1}) + f(s_{2i})]$$

$$\frac{\Psi(\dots(\mathbf{q}_{iI}^\alpha, s'_i)\dots)}{\Psi(\dots(\mathbf{r}_i, s_i)\dots)} \rightarrow \frac{J(\dots \mathbf{q}_{iI}^\alpha \dots)}{J(\dots \mathbf{r}_i \dots)} \sum_n C_{ni}(\mathbf{r}_i, s_i) \psi_n(\mathbf{r}_{iI}^\alpha, s'_i)$$

$$\sum_{i,I} \sum_\alpha w_\alpha \sum_{\ell,m,m'} v_\ell^{\text{SO}}(r_{iI}) Y_{\ell,m}^*(\mathbf{r}_{iI}) Y_{\ell,m'}(r_{iI} \mathbf{u}^\alpha) \frac{J(\dots \mathbf{q}_{iI}^\alpha \dots)}{J(\dots \mathbf{r}_i \dots)} \sum_n C_{ni}(\mathbf{r}_i, s_i) \int \frac{ds'_i}{2\pi} \psi_n(\mathbf{q}_{iI}^\alpha, s'_i) \Lambda_{\ell,m,m',s_i,s'_i}$$

$$\frac{1}{2} \left(\varphi^\uparrow(\mathbf{q}_{iI}^\alpha) e^{is_i} \left[(-\ell_x^{\ell,m,m'} + i\ell_y^{\ell,m,m'}) e^{-2is_i} + \ell_z^{\ell,m,m'} \right] + \varphi^\downarrow(\mathbf{q}_{iI}^\alpha) e^{-is_i} \left[(\ell_x^{\ell,m,m'} - i\ell_y^{\ell,m,m'}) e^{2is_i} \right] - \ell_z^{\ell,m,m'} \right)$$

SO-CCECPs SHOW IMPROVED ACCURACY

Significant improvement using ccECPs over existing AREP/SOREP potentials

Previous ECPs (AREP only) show significant overbinding in CCSD(T). Improving the AREP with our ccECP improves both the bond length and binding energy.

Including SOC in DMC still shows significant overbinding using previous ECPs. By using our improved ccECPs, obtain excellent agreement with experiment.

**Experimental bond length
and binding energy**

