

QMC Workshop 2021

Optimization of Pseudopotentials

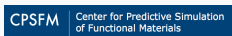
Week 6 / 9 November 2021

Gani Annaberdiyev, aannabe@ncsu.edu

North Carolina State University

https://github.com/QMCPACK/qmc_workshop_2021

Funding: U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division, as part of the Computational Materials Sciences Program and Center for Predictive Simulation of Functional Materials.



- DMC is an accurate many-body method that can estimate the correlation energy very accurately ($\sim 99\%$ of correlation energy in diamond Si using single-ref [2]).
- However, it needs to solve the correct, accurate Hamiltonian to reliably make predictions.
- An accurate, relativistic pseudopotential/ECP is therefore desired with the following features:
 1. **Many-body construction:** ECP must reference correlated, relativistic all-electron (AE) data, and must be optimized using an accurate many-body method.
 2. **Tested and Universal:** Usable in any method: mean-field or many-body methods, similar to original AE Hamiltonian.

Conventional ECP Optimization Approaches

For a chosen subset S of atomic spectrum, generate relativistic all-electron (AE) reference states using HF (mean-field). Two main approaches are:

1. Energy-consistency of the atomic spectrum gaps:

$$\mathcal{E}_{\text{HF}}^2 = \sum_{i \in S} \left(\Delta E_{\text{HF}}^{\text{ECP}} - \Delta E_{\text{HF}}^{\text{AE}} \right)_i^2 \quad (1)$$

2. Shape-consistency:

$$\mathcal{N}^2 = \sum_{\ell} \left[\left(\epsilon_{\ell}^{\text{ECP}} - \epsilon_{\ell}^{\text{AE}} \right)^2 + \left(N_{\ell}^{\text{ECP}} - N_{\ell}^{\text{AE}} \right)^2 + \left(V_{\ell}^{\text{ECP}} - V_{\ell}^{\text{AE}} \right)^2 + \left(S_{\ell}^{\text{ECP}} - S_{\ell}^{\text{AE}} \right)^2 \right] \quad (2)$$

where ϵ_{ℓ} : eigenvalues, $N_{\ell}^{\text{AE}} = \int_0^{R_{\ell}} (r^{\ell+1} \phi_{\ell}^{\text{AE}}(r))^2 dr$, $V_{\ell}^{\text{AE}} = \phi_{\ell}^{\text{AE}}(R_{\ell})$, $S_{\ell}^{\text{AE}} = \frac{d}{dr} \phi_{\ell}^{\text{AE}}(r)|_{R_{\ell}}$, R_{ℓ} : cutoff radius ($r^{\frac{4}{5}} \phi_{\ell}^{\text{AE}}$).

A new approach: go beyond mean-field HF, and consider electron-electron correlations.

1. AE reference data: scalar relativistic, fully-correlated (CC, CV, VV) CCSD(T).
2. Many-body method is used:

$$\mathcal{E}_{\text{CCSD(T)}}^2 = \sum_{i \in S} \left(\Delta E_{\text{CCSD(T)}}^{\text{ECP}} - \Delta E_{\text{CCSD(T)}}^{\text{AE}} \right)_i^2 + \omega \mathcal{N}^2 \quad (3)$$

3. Simple form parametrized with few Gaussians.
4. More emphasis on testing/transferability, especially on compressed XH and XO bond lengths.

Deviations between AE and ECP atomic spectrum energies

AE Reference: CCSD(T) with all-electrons correlated.

$$\text{MAD} = \frac{1}{N_s} \sum_{s=1}^{N_s} \left| \Delta E_s^{\text{ECP}} - \Delta E_s^{\text{AE}} \right| \quad (4)$$

BFD: Burkatzki-Filippi-Dolg DHF ECPs for QMC [5, 6]

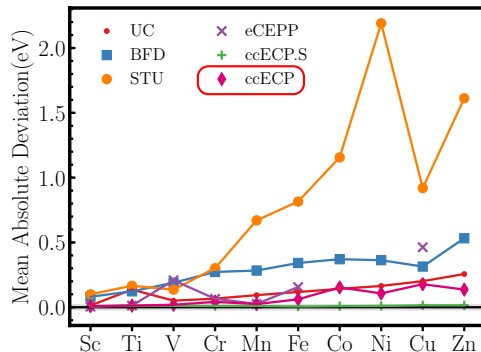
STU: Stuttgart group DHF ECPs [9]

eCEPP: Trail-Needs correlated ECPs for QMC [11]

ccECP: New correlation-consistent ECPs [3]

UC: AE, uncorrelated core CCSD(T)

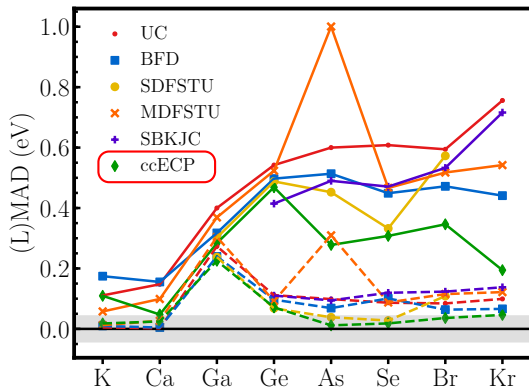
ccECP shows significant improvement over older ECPs.



Annaberdiev et al, J. Chem. Phys. 149, 134108 (2018) [3]

QMC Workshop 2021

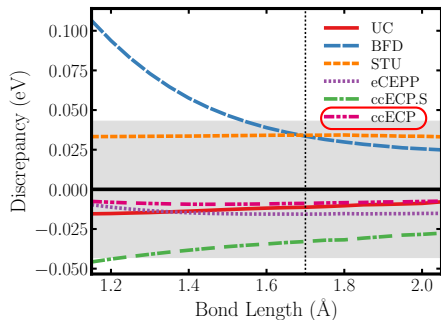
- Many states are included: neutral excitations, EA, IPs down to the bare core.
 - Accurate in low-lying excitations (LMAD) and for large set of states (MAD).
 - Better iso-spectrality in all ccECPs: H-Kr.
- Training data was optimized better. How does this carry to more complicated systems?
- Following slides demonstrate better transferability on fidelity of molecular bonding curves.



Wang et al, J. Chem. Phys. 151, 144110 (2019) [12]

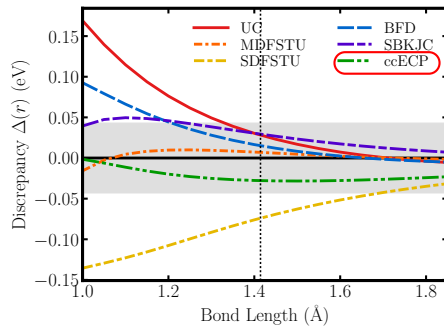
Discrepancy between AE and ECP monohydride binding energies (σ bonds)

VH molecule



Annaberdiyev et al, J. Chem. Phys. 149, 134108 (2018) [3]

BrH molecule

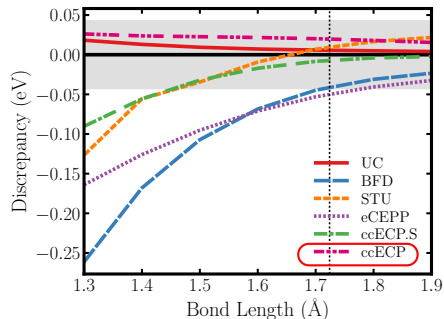


Wang et al, J. Chem. Phys. 151, 144110 (2019) [12]

QMC Workshop 2021

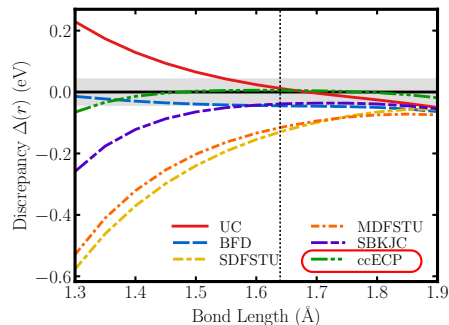
Discrepancy between AE and ECP monoxide binding energies (π bonds)

CuO molecule



Annaberdiyev et al, J. Chem. Phys. 149, 134108 (2018) [3]

SeO molecule

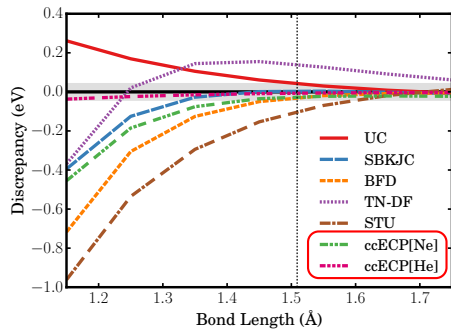


Wang et al, J. Chem. Phys. 151, 144110 (2019) [12]

QMC Workshop 2021

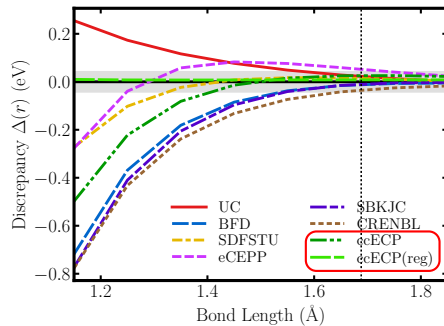
Various core sizes are available for ccECP

SiO molecule



Bennett et al, J. Chem. Phys. 149, 104108 (2018) [4]

LiO molecule



Wang et al, J. Chem. Phys. 151, 144110 (2019) [12]

QMC Workshop 2021

DMC singlet/triplet excitations for Si systems agree well with experiments.

TABLE IV. Si₂H₆ total energies (Ha) and vertical excitation gaps (eV) using various methods. Experimental geometry was used for this molecule.²²

	GS (¹ A _{1g})	EX (¹ E _{1u})		EX (³ A _{1g})	
Method	Total (hartree)	Total (hartree)	Gap (eV)	Total (hartree)	Gap (eV)
CISD/RHF ^a	−11.3287	−10.8183	13.889	−11.0825	6.699
CISD/CAS(14e [−] , 13o) ^b	−11.3400	−11.0623	7.557		
RCCSD(T)/RHF ^c	−11.3766(3)			−11.1308(5)	6.69(2)
CCSDT(Q)/RHF ^d	−11.3782(3)			−11.1336(5)	6.66(2)
DMC/RHF	−11.3708(2)	−11.0855(2)	7.763(8)	−11.1215(2)	6.784(8)
DMC/PBE ^e	−11.3725(2)	−11.0934(2)	7.595(8)	−11.1248(2)	6.740(8)
Lehtonen <i>et al.</i> ^f			7.61		
Experiment ⁴¹			7.6		≈6.7 ^g
Experiment ³⁹			7.56		

DMC Benchmarks of ccECP

DMC IP and EA of variety of molecular systems were tested by an *independent* groups.

The molecular systems tested:

System	
C ⁺	PH ⁺
C	PH
C ⁻	PH ⁻
N ⁺	PH ₂ ⁺
N	PH ₂
O ⁺	PH ₂ ⁻
O	S ⁺
O ⁻	S
OH ⁺	S ⁻
OH	SH ⁺
OH ⁻	SH
O ₂ ⁺	SH ⁻
O ₂	S ₂ ⁺
O ₂ ⁻	S ₂
Si ⁺	S ₂ ⁻
Si	Cl ⁺
Si ⁻	Cl
P ⁺	Cl ⁻
P	Cl ₂ ⁺
P ⁻	Cl ₂
	Cl ₂ ⁻

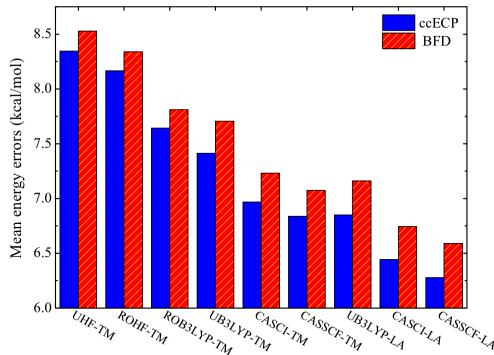


FIG. 5. Mean errors in DMC energies using ccECP and BFD potentials with different trial wave functions and nonlocal-treating strategies.

- The library is hosted on GitHub:
<https://github.com/QMCPACK/pseudopotentiallibrary>
- AREP ccECPs for H-Kr elements are available now.
- (aug)-cc-p(C)VnZ [$n = D - 6$] basis-sets are available.
- Commonly used formats are available: MOLPRO, GAMESS, NWCHEM, PYSCF.
- Also, UPF files are available for use in plane wave basis codes.
- Heavier elements with SOC will be added later, attend Week-8 to learn how to run with SOREP.

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.

H																						He
Li	Be											B	C	N	O	F	Ne					
Na	Mg											Al	Si	P	S	Cl	Ar					
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr					
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	Hg	Tl	Pb	Bi	Po	At					
Cs	Ba	La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu	Be					
Fr	Ra	Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	Nd	Ta	Bi					

Copper

RRKJ_PRJ_93_075143	TM_PRB_93_075143	eCEPP	ccECP
Rappe-Rabe-Kastras-Joannopoulos potential from J. T. Krogel et al. Physical Review B 93 075143 (2016)	Trouiller-Martins potential from J. T. Krogel et al. Physical Review B 93 075143 (2016)	eCEPP from J. R. Trail and R. J. Needs Journal of Chemical Physics 146, 204107 (2017)	ccECP from A. Annaberdiyev et al. Journal of Chemical Physics 149, 134108 (2018)
Cu.opt.uspf	Cu.tm.uspf	Cu.awfn	Cu.cc-pCVTZ_gamess
Cu.opt.xml	Cu.tm.xml	Cu.data	Cu.cc-pCVTZ_molpro

See Week 3 and Week 4 material for introduction to NEXUS.

Update workshop example files:

```
cd $HOME/apps/qmcpack  
git pull
```

```
cd $HOME/qmc_workshop_2021  
git pull
```

Enter week 6 example directory:

```
cd week6_pseudopotentials/02_ccECP_slides_runs/Sc_atom/workstation/
```

```
ls  
geom.xyz  pseudo  runs  scf_template.py  workflow.py
```

```
ls runs/  
c4q  gs_noj  optJ12  optJ123  qmc_no  qmc_v0  qmc_v1  scf
```

```
ls pseudo/  
download.sh  Sc.ccECP.xml
```

```
vim pseudo/download.sh
```

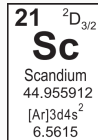
```
1 #!/usr/bin/env bash  
2  
3 atom1="Sc"  
4  
5 wget https://pseudopotentiallibrary.org/recipes/${atom1}/ccECP/${atom1}.ccECP.xml
```

```
vim geom.xyz
```

```
1 1  
2  
3 Sc  0.0000  0.0000  0.0000
```

vim scf_template.py

```
1 #!/usr/bin/env python3
2
3 import sys, os
4 from pyscf import scf, gto
5 import numpy as np
6 from urllib.request import urlretrieve
7
8 ### Set the current working directory
9 cwd = os.getcwd()
10 pplib = "http://pseudopotentiallibrary.org/recipes"
11 pptype = "ccECP"
12
13 ### Retrieve basis and ECP files from pseudopotentiallibrary.org
14 atom1 = "Sc"
15 basetype1 = "cc-pVTZ"
16 basfile1 = "{0}.{1}.nwchem".format(atom1,basetype1)
17 ecfile1 = "{0}.{1}.nwchem".format(atom1,pptype)
18 xmlfile1 = "{0}.{1}.xml".format(atom1,pptype)
19 urlretrieve("{0}/{1}/{2}/{3}".format(pplib,atom1,pptype,basfile1), filename=basfile1)
20 urlretrieve("{0}/{1}/{2}/{3}".format(pplib,atom1,pptype,ecfile1), filename=ecfile1)
21 urlretrieve("{0}/{1}/{2}/{3}".format(pplib,atom1,pptype,xmlfile1), filename=xmlfile1)
22
23 ###~~~~ Build the molecule ~~~~
24
25 # Nexus expands this with Mole info
26 $system
27
28 with open(os.path.join(cwd,basfile1)) as f:
29     bas1 = f.read()
30 with open(os.path.join(cwd,ecfile1)) as f:
31     ec1 = f.read()
32 mol.basis = {atom1: gto.basis.parse(bas1)}
33 mol.ecp = {atom1: gto.basis.parse_ecp(ec1)}
34 mol.build()
```



```
35
36 #~~~Run HF on molecule~~~
37 mf = scf.R0HF(mol)
38 mf.irrep_nelec = {
39 'Ag' : (3,2), # s, dz^2, dx^2-y^2
40 'B3u' : (1,1), # x 1
41 'B1u' : (1,1), # z 0
42 'B2u' : (1,1), # y -1
43 'B2g' : (0,0), # xz 1
44 'B3g' : (0,0), # yz -1
45 'B1g' : (0,0), # xy -2
46 'Au' : (0,0) # xyz
47 }
48 mf.max_cycle=100
49 mf.chkfile='scf.chkfile'
50 en=mf.kernel()
51 mf.analyze()
52
```

vim workflow.py

```
1 #!/usr/bin/env python3
2
3 from nexus import settings, job, run_project, obj
4 from nexus import generate_physical_system
5 from nexus import generate_pyscf
6 from nexus import generate_convert4qmc
7 from nexus import generate_qmcpack, loop, linear, vmc, dmc
8
9 # Obtain the core count of the local machine (lab only)
10 import os
11 cores = os.cpu_count()
12
13 settings(
14     pseudo_dir = 'pseudo',
15     results    = '.',
16     runs       = 'runs',
17     status_only = 0,
18     generate_only = 0,
19     sleep      = 10,          # In seconds
20     machine    = 'ws'+str(cores),
21 )
22
23 system = generate_physical_system(
24     structure = 'geom.xyz',
25     Sc        = 11,          # 11 valence electrons
26     net_spin  = 1,           # 2S
27     net_charge = 0,
28 )
29
```

```
29
30 sims = []
31
32 ## Perform SCF
33 scf = generate_pyscf(
34     identifier = 'scf', # Log output goes to scf.out
35     path       = 'scf', # Directory to run in
36     job        = job(serial=True, app='python3'),
37     system     = system,
38     template   = 'scf_template.py',      # pyscf template file
39     mole       = obj(),                  # used for the Mole() object
40     verbose    = 4,
41     symmetry   = 'D2h',
42 ),
43     save_qmc   = True,                  # save wfn data for qmcpack
44 )
45 sims.append(scf)
46
47 ##### convert orbitals to QMCPACK format
48 c4q = generate_convert4qmc(
49     identifier = 'c4q',
50     path       = 'c4q',
51     job        = job(cores=1),
52     no_jastrow = True,
53     dependencies = (scf, 'orbitals'),
54 )
55 sims.append(c4q)
56
57 ### collect dependencies relating to orbitals
58 orbdeps = [(c4q, 'particles'), # pyscf changes particle positions
59            (c4q, 'orbitals'), ]
60
```



```
python workflow.py
```

```
...
```

```
ls runs/scf/
```

```
Sc.ccECP.nwchem Sc.ccECP.xml Sc.cc-pVTZ.nwchem scf.chkfile scf.err
```

```
scf.h5 scf.out scf.py scf.sbatch.in scf.struct.xyz sim_scf
```

```
vim runs/scf/scf.py
```

```
22
23 ###~~~~ Build the molecule ~~~~
24
25 # Nexus expands this with Mole info
26
27 ### generated system text ###
28 from pyscf import gto as gto_loc
29 mol = gto_loc.Mole()
30 mol.verbose = 4
31 mol.atom = '''
32      Sc    0.00000000    0.00000000    0.00000000
33      '''
34 mol.unit = 'A'
35 mol.charge = 0
36 mol.spin = 1
37 mol.symmetry = 'D2h'
38 mol.build()
39 ### end generated system text ###
40
```

```
68
69 ### generated conversion text ###
70 from PyscfToQmcpack import savetoqmcpack
71 savetoqmcpack(mol,mf,'scf')
72 ### end generated conversion text ###
73
```

```

164 cycle= 15 E= -46.121844325316 delta E= 2.48e-06 |g|= 0.0001 |ddm|= 0.0232
165 HOMO (Ag) = -0.0512860808024703 LUMO (B1u) = 0.0212213370518054
166 cycle= 16 E= -46.1218430105355 delta E= 1.31e-06 |g|= 0.000361 |ddm|= 0.0316
167 HOMO (Ag) = -0.051165418814432 LUMO (B1u) = 0.021223988441713
168 cycle= 17 E= -46.1218430689892 delta E= -5.85e-08 |g|= 0.000251 |ddm|= 0.00115
169 HOMO (Ag) = -0.0511178095270498 LUMO (B1u) = 0.0212239829283067
170 cycle= 18 E= -46.1218430876165 delta E= -1.86e-08 |g|= 0.000204 |ddm|= 0.00084
171 HOMO (Ag) = -0.0509412957047003 LUMO (B1u) = 0.0212268278098667
172 cycle= 19 E= -46.1218431223565 delta E= -3.47e-08 |g|= 4.31e-05 |ddm|= 0.00101
173 HOMO (Ag) = -0.0508978866963989 LUMO (B1u) = 0.021226681589223
174 cycle= 20 E= -46.1218431246405 delta E= -2.28e-09 |g|= 3.64e-06 |ddm|= 0.00103
175 HOMO (Ag) = -0.0508900896655348 LUMO (B1u) = 0.0212268078866954
176 cycle= 21 E= -46.1218431242121 delta E= 4.28e-10 |g|= 5.44e-06 |ddm|= 0.00042
177 HOMO (Ag) = -0.0508976447814269 LUMO (B1u) = 0.0212266453759735
178 Extra cycle E= -46.1218431242235 delta E= -1.14e-11 |g|= 4.05e-06 |ddm|= 1.69e-05
179 converged SCF energy = -46.1218431242235
180 **** SCF Summaries ****
181 Total Energy = -46.121843124223474
182 Nuclear Repulsion Energy = 0.000000000000000
183 One-electron Energy = -76.798902125225936
184 Two-electron Energy = 30.677059001002462
185 Wave-function symmetry = Ag
186 occupancy for each irrep: Ag B1g B2g B3g Au B1u B2u B3u
187 double occ 2 0 0 0 0 1 1 1
188 single occ 1 0 0 0 0 0 0 0
189 **** MO energy ****
190
191 MO #1 (Ag #1) energy= -2.59527820108142 | Rootaan | alpha | beta | -2.56611299757682 occ= 2
192 MO #2 (B1u #1) energy= -1.58555599442954 | -1.5925394874459 | -1.57857250141318 occ= 2
193 MO #3 (B3u #1) energy= -1.57875667542816 | -1.63032348524351 | -1.5271898656128 occ= 2
194 MO #4 (B2u #1) energy= -1.57875438901967 | -1.63033557807 | -1.52717319996934 occ= 2
195 MO #5 (Ag #2) energy= -0.210835058082764 | -0.215125004312978 | -0.206545111853038 occ= 2
196 MO #6 (Ag #3) energy= -0.0508976447814269 | -0.346452079424369 | 0.244658966793811 occ= 1
197 MO #7 (B1u #2) energy= 0.0212266453759735 | 0.02094525158588 | 0.0215080391660666 occ= 0
198 MO #8 (B3u #2) energy= 0.024412614689053 | 0.0228037087932995 | 0.0260215205848067 occ= 0
199 MO #9 (B2u #2) energy= 0.0244135828037429 | 0.0228041642565865 | 0.0260230013508989 occ= 0
200 MO #10 (Ag #4) energy= 0.0638801098953974 | 0.0633192197670475 | 0.0644410000219459 occ= 0
201 MO #11 (B2g #1) energy= 0.0642676989705254 | 0.0638575172384601 | 0.0646778807025906 occ= 0

```

```

251 MO #61 (B3u #9) energy= 2.38408239635703 | 2.35
252 MO #62 (B2u #9) energy= 2.38408249146256 | 2.35
253 MO #63 (Ag #16) energy= 5.42588680466658 | 5.42
254 ** Mulliken pop on meta-lowdin orthogonal AOs **
255 ** Mulliken pop **
256 pop of 0 Sc 3s 1.99988
257 pop of 0 Sc 4s 1.99833
258 pop of 0 Sc 5s 0.00000
259 pop of 0 Sc 6s 0.00000
260 pop of 0 Sc 7s 0.00000
261 pop of 0 Sc 3px 1.99781
262 pop of 0 Sc 3py 1.99781
263 pop of 0 Sc 3pz 1.99833
264 pop of 0 Sc 4px 0.00083
265 pop of 0 Sc 4py 0.00083
266 pop of 0 Sc 4pz 0.00063
267 pop of 0 Sc 5px 0.00114
268 pop of 0 Sc 5py 0.00114
269 pop of 0 Sc 5pz 0.00091
270 pop of 0 Sc 6px 0.00006
271 pop of 0 Sc 6py 0.00006
272 pop of 0 Sc 6pz 0.00007
273 pop of 0 Sc 7px 0.00003
274 pop of 0 Sc 7py 0.00003
275 pop of 0 Sc 7pz 0.00002
276 pop of 0 Sc 3dxy 0.00000
277 pop of 0 Sc 3dyz 0.00000
278 pop of 0 Sc 3dz^2 0.00145
279 pop of 0 Sc 3dxz 0.00000
280 pop of 0 Sc 3dx2-y2 0.92126
281 pop of 0 Sc 4dxy 0.00000
282 pop of 0 Sc 4dyz 0.00000
283 pop of 0 Sc 4dz^2 0.00029
284 pop of 0 Sc 4dxz 0.00000
285 pop of 0 Sc 4dx2-y2 0.06730
286 pop of 0 Sc 5dxy 0.00000
287 pop of 0 Sc 5dyz 0.00000
288 pop of 0 Sc 5dz^2 0.00004
289 pop of 0 Sc 5dxz 0.00000

```

```
vim workflow.py
```

```
61
62 ### Run VMC without Jastrow for sanity check
63 gs_noj = generate_qmcpack(
64     identifier = 'gs_noj',
65     path       = 'gs_noj',
66     job        = job(cores=cores),
67     system     = system,
68     pseudos    = ['Sc.ccECP.xml'],
69     jastrows   = [],
70     calculations = [
71         vmc(
72             walkers = 1,
73             warmupsteps = 20,
74             blocks = 5,
75             steps = 50,
76             substeps = 2,
77             timestep = 0.7,
78         ),
79     ],
80     dependencies = orbdeps,
81 )
82 sims.append(gs_noj)
83
```

```
cd runs/gs_noj/
```

```
qmca -q ekpvar -e 3 --sac *.scalar.*
```

```
gs_noj series 0
LocalEnergy      = -46.12177 +/- 0.00072 1.3
Variance         = 4.493 +/- 0.024 1.3
Kinetic          = 18.7966 +/- 0.0027 1.0
LocalPotential   = -64.9184 +/- 0.0028 1.0
AcceptRatio      = 0.498003 +/- 0.000029 1.0
```

SCF energy = -46.121843 Ha

```
vim workflow.py
```

```
84 ##### OPTIMIZATION methods
85 linopt1 = linear(
86     energy           = 0.0,
87     unreweightedvariance = 1.0,
88     reweightedvariance = 0.0,
89     samples          = int(3e3),
90     substeps         = 2,
91     steps            = 20,
92     blocks           = 20,
93     nonlocalpp       = True,
94     usedrift         = True,
95     minmethod        = 'OneShiftOnly',
96     minwalkers       = 1e-3,
97     timestep         = 1.0,
98 )
99 linopt2 = linopt1.copy()
100 linopt2.minwalkers = 0.10
101 linopt2.samples = linopt1.samples*2
102
103 linopt3 = linopt2.copy()
104 linopt3.unreweightedvariance = 0.0
105 linopt3.reweightedvariance = 0.10
106 linopt3.energy = 0.90
107 linopt3.minwalkers = 0.30
108 linopt3.samples = linopt2.samples*2
109
110 ##### optimize 1, 2-body Jastrow
111 optJ12 = generate_qmcpack(
112     identifier = 'optJ12',
113     path       = 'optJ12',
114     job        = job(cores=cores),
115     system     = system,
116     J2         = True,           # 2-body B-spline Jastrow
117     J1_rcut    = 12.0,          # Bohr cutoff for J1
118     J2_rcut    = 12.0,          # Bohr cutoff for J2
119     psuedos    = ['Sc.ccECP.xml'],
120     calculations = [
121         loop(max=5, qmc=linopt1),
122         loop(max=5, qmc=linopt2),
123         loop(max=5, qmc=linopt3),
124     ],
125     dependencies = orbdeps,
126 )
127 sims.append(optJ12)
```

```
cd runs/optJ12
```

```
qmca -q ev -e 1 --sac *.scalar.*
```

		LocalEnergy		Variance	ratio		
optJ12	series 0	-46.138778 +/- 0.002003	1.0	3.076381 +/- 0.006039	1.0	0.0667	
optJ12	series 1	-46.419171 +/- 0.001346	2.2	0.606835 +/- 0.002252	1.0	0.0131	
optJ12	series 2	-46.452039 +/- 0.000896	1.0	0.625210 +/- 0.003446	1.0	0.0135	
optJ12	series 3	-46.458866 +/- 0.000884	1.0	0.654501 +/- 0.005287	1.0	0.0141	
optJ12	series 4	-46.461309 +/- 0.001270	1.0	0.663231 +/- 0.005007	1.2	0.0143	
optJ12	series 5	-46.464978 +/- 0.000693	1.0	0.665643 +/- 0.003202	1.0	0.0143	
optJ12	series 6	-46.466492 +/- 0.000964	1.0	0.670662 +/- 0.004712	1.0	0.0144	
optJ12	series 7	-46.467072 +/- 0.000584	1.0	0.669408 +/- 0.003340	1.0	0.0144	
optJ12	series 8	-46.469110 +/- 0.000763	1.0	0.686031 +/- 0.013676	1.0	0.0148	
optJ12	series 9	-46.469448 +/- 0.000838	2.5	0.680276 +/- 0.005176	1.0	0.0146	
optJ12	series 10	-46.468701 +/- 0.000573	1.0	0.675707 +/- 0.002290	1.4	0.0145	
optJ12	series 11	-46.469053 +/- 0.000484	1.0	0.678638 +/- 0.002415	1.0	0.0146	
optJ12	series 12	-46.469395 +/- 0.000546	1.7	0.676652 +/- 0.001945	1.0	0.0146	
optJ12	series 13	-46.469334 +/- 0.000357	1.0	0.680689 +/- 0.003857	1.0	0.0146	
optJ12	series 14	-46.470180 +/- 0.000619	1.0	0.683609 +/- 0.007195	4.6	0.0147	

QMC Workshop 2021

`vim workflow.py`

```
128 ##### optimize 3-body Jastrow
129 optJ123 = generate_qmcpack(
130     identifier = 'optJ123',
131     path       = 'optJ123',
132     job        = job(cores=cores),
133     system     = system,
134     J3         = True,
135     J3_rcut    = 12.0,
136     pseudos    = ['Sc.ccECP.xml'],
137     calculations = [
138         loop(max=10, qmc=linopt3),
139     ],
140     dependencies = orbdeps+[(optJ12, 'jastrow')],
141 )
142 sims.append(optJ123)
144
```

`cd runs/optJ123`

`qmca -q ev -e 1 --sac *.scalar.*`

		LocalEnergy		Variance	ratio	
optJ123	series 0	-46.467934 +/- 0.000419	1.0	0.672600 +/- 0.002882	1.0	0.0145
optJ123	series 1	-46.489434 +/- 0.000408	1.0	0.578860 +/- 0.008589	1.0	0.0125
optJ123	series 2	-46.490477 +/- 0.000430	1.0	0.656540 +/- 0.029653	1.0	0.0141
optJ123	series 3	-46.491878 +/- 0.000578	1.1	0.656817 +/- 0.006533	1.0	0.0141
optJ123	series 4	-46.492373 +/- 0.000553	1.0	0.681497 +/- 0.012269	1.0	0.0147
optJ123	series 5	-46.492559 +/- 0.000996	3.2	0.673054 +/- 0.008965	1.3	0.0145
optJ123	series 6	-46.492327 +/- 0.000518	1.0	0.666388 +/- 0.008933	1.0	0.0143
optJ123	series 7	-46.493250 +/- 0.000335	1.0	0.683002 +/- 0.008520	1.0	0.0147
optJ123	series 8	-46.493760 +/- 0.000440	1.0	0.682064 +/- 0.015371	1.0	0.0147
optJ123	series 9	-46.493525 +/- 0.000588	1.0	0.680150 +/- 0.005577	1.1	0.0146

vim workflow.py

```
145 t_moves = ['no', 'v0', 'v1']
146 locality_run = {}
147
148 ### run DMC with QMCPACK
149 MY_TARGET_WALKERS = 16 # Small number for presentation purposes. Needs to be at least a few thousand walkers.
150
151 for moves in t_moves:
152     name = 'qmc_' + moves
153
154     locality_run[moves] = generate_qmcpack(
155         identifier = name,
156         path = name,
157         job = job(cores=cores),
158         system = system,
159         pseudos = ['Sc.ccECP.xml'],
160         jastrows = [],
161         calculations = [
162             vmc(
163                 #walkers = 1,
164                 walkers = int(MY_TARGET_WALKERS/cores), # Per MPI
165                 warmupsteps = 20,
166                 blocks = 5,
167                 steps = 50,
168                 substeps = 2,
169                 timestep = 0.5,
170             ),
171             dmc(targetwalkers = MY_TARGET_WALKERS, # Total walkers
172                 timestep = 0.02,
173                 warmupsteps = int(5/0.02),
174                 blocks = 5,
175                 steps = int(1.0/0.02),
176                 nonlocalmoves = moves,
177                 checkpoint = 5
178             ),
```

```
179             dmc(targetwalkers = MY_TARGET_WALKERS, # Total walkers
180                 timestep = 0.01,
181                 warmupsteps = int(2/0.01),
182                 blocks = 5,
183                 steps = int(1.0/0.01),
184                 nonlocalmoves = moves,
185                 checkpoint = 5
186             ),
187             dmc(targetwalkers = MY_TARGET_WALKERS, # Total walkers
188                 timestep = 0.005,
189                 warmupsteps = int(1/0.005),
190                 blocks = 5,
191                 steps = int(1.0/0.005),
192                 nonlocalmoves = moves,
193                 checkpoint = 5
194             ),
195             dmc(targetwalkers = MY_TARGET_WALKERS, # Total walkers
196                 timestep = 0.0025,
197                 warmupsteps = int(1/0.0025),
198                 blocks = 5,
199                 steps = int(1.0/0.0025),
200                 nonlocalmoves = moves,
201                 checkpoint = 5
202             ),
203         ],
204         dependencies = orbdeps+[optjl123,'jastrow']],
205     )
206     sims.append(locality_run[moves])
207
208 run_project(sims)
209
```

For more info see:

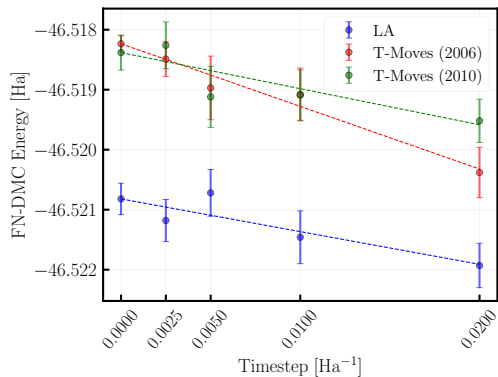
<https://qmcpack.readthedocs.io/en/develop/methods.html#dmc-driver>

```
ls runs/  
c4q gs_noj optJ12 optJ123  
qmc_no qmc_v0 qmc_v1 scf
```

```
cd runs/qmc_v1/
```

```
qmca -q ev -e 3 *scalar*
```

		LocalEnergy		Variance		ratio
qmc_v1	series 0	-46.488935	+/- 0.000314	0.575068	+/- 0.003038	0.0124
qmc_v1	series 1	-46.519518	+/- 0.000364	0.593276	+/- 0.003472	0.0128
qmc_v1	series 2	-46.519089	+/- 0.000422	0.584251	+/- 0.002304	0.0126
qmc_v1	series 3	-46.519124	+/- 0.000513	0.578059	+/- 0.001688	0.0124
qmc_v1	series 4	-46.518257	+/- 0.000394	0.574395	+/- 0.001310	0.0123



Locality Treatment	Execution Time (min)
LA ("no") [10]	18.3
T-Moves (2006) ("v0") [7]	21.0
T-Moves (2010) ("v1") [8]	26.6

Table 1: Comparison of execution times for various locality treatments. NERSC, 8 KNL nodes.

Checking the results against reference data

Table 17. Most Accurate Total Energies for ccECP[Ne] K–Zn Elements along with Fixed-Node DMC Energies with Single-Reference HF Trial Wave Functions^a

atom	state	"exact" (Ha)	DMC/HF (Ha)	ϵ (mHa)	η
K	(² S)	−28.25243(25)	−28.2394(2)	13.0(3)	4.1(1)
Ca	(¹ S)	−36.72897(46)	−36.7055(2)	23.5(5)	6.2(1)
Sc	(² D)	−46.55704(81)	−46.5202(4)	36.8(9)	8.4(2)
Ti	(³ F)	−58.09263(76)	−58.0458(2)	46.8(8)	9.6(2)
V	(⁴ F)	−71.44178(59)	−71.3829(2)	58.9(6)	10.8(1)
Cr	(⁷ S)	−86.64109(33)	−86.5876(2)	53.5(4)	9.03(7)
Mn	(⁶ S)	−103.8919(10)	−103.8260(3)	66(1)	10.2(2)
Fe	(⁵ D)	−123.38804(93)	−123.3100(3)	78(1)	10.5(1)
Co	(⁴ F)	−145.1541(10)	−145.0709(3)	83(1)	10.1(1)
Ni	(³ F)	−169.3912(12)	−169.2973(6)	94(1)	10.3(1)
Ni	(³ D)	−169.3932(12)	−169.3056(6)	88(1)	9.0(1)
Cu	(² S)	−196.4038(10)	−196.3178(3)	86(1)	8.1(1)
Zn	(¹ S)	−226.3699(18)	−226.2775(4)	92(2)	8.4(2)

Annaberdiyev et al, J. Chem. Theory Comput. 2020, 16, 3, 1482–1502 [1]

Another reason to use T-Moves: Avoiding instabilities

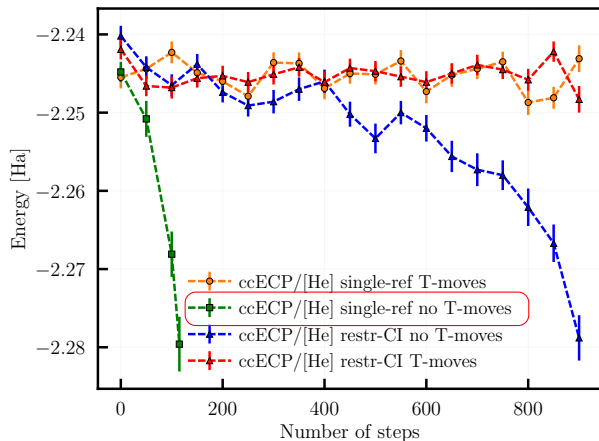









Figure: FNDMC energies vs number of steps (in time steps of 0.001 Ha^{-1}) for BeH_2 molecule. Be atom is represented by ccECP/[He].





Key takeaway message: The quality of pseudopotentials require thorough testing for use in production calculations.




So far, we have only focused on ECPs with *averaged* spin-orbit effects. Calculations with explicit spin-orbit terms will be covered in Week 8.


`$HOME/qmc_workshop_2021/week8_spin-orbit_qmc/`

-  A. Annaberdiyev, C. A. Melton, M. C. Bennett, G. Wang, and L. Mitas.
Accurate Atomic Correlation and Total Energies for Correlation Consistent Effective Core Potentials.
Journal of Chemical Theory and Computation, 16(3):1482–1502, Mar. 2020.
-  A. Annaberdiyev, G. Wang, C. A. Melton, M. C. Bennett, and L. Mitas.
Cohesion and excitations of diamond-structure silicon by quantum Monte Carlo: Benchmarks and control of systematic biases.
Physical Review B, 103(20):205206, May 2021.
-  A. Annaberdiyev, G. Wang, C. A. Melton, M. C. Bennett, L. Shulenburger, and L. Mitas.
A new generation of effective core potentials from correlated calculations: 3d transition metal series.
The Journal of Chemical Physics, 149(13):134108, Oct. 2018.

-  M. C. Bennett, G. Wang, A. Annaberdiyev, C. A. Melton, L. Shulenburger, and L. Mitas.
A new generation of effective core potentials from correlated calculations: 2nd row elements.
The Journal of Chemical Physics, 149(10):104108, Sept. 2018.
-  M. Burkatzki, C. Filippi, and M. Dolg.
Energy-consistent pseudopotentials for quantum Monte Carlo calculations.
The Journal of Chemical Physics, 126(23):234105, June 2007.
-  M. Burkatzki, C. Filippi, and M. Dolg.
Energy-consistent small-core pseudopotentials for 3d-transition metals adapted to quantum Monte Carlo calculations.
The Journal of Chemical Physics, 129(16):164115, Oct. 2008.
-  M. Casula.
Beyond the locality approximation in the standard diffusion Monte Carlo method.
Physical Review B, 74(16):161102, Oct. 2006.

-  M. Casula, S. Moroni, S. Sorella, and C. Filippi.
Size-consistent variational approaches to nonlocal pseudopotentials: Standard and lattice regularized diffusion Monte Carlo methods revisited.
The Journal of Chemical Physics, 132(15):154113, Apr. 2010.
-  M. Dolg, U. Wedig, H. Stoll, and H. Preuss.
Energy-adjusted ab initio pseudopotentials for the first row transition elements.
The Journal of Chemical Physics, 86(2):866–872, Jan. 1987.
-  L. Mitáš, E. L. Shirley, and D. M. Ceperley.
Nonlocal pseudopotentials and diffusion Monte Carlo.
The Journal of Chemical Physics, 95(5):3467–3475, Sept. 1991.
-  J. R. Trail and R. J. Needs.
Shape and energy consistent pseudopotentials for correlated electron systems.
The Journal of Chemical Physics, 146(20):204107, May 2017.

-  G. Wang, A. Annaberdiyev, C. A. Melton, M. C. Bennett, L. Shulenburger, and L. Mitas.
A new generation of effective core potentials from correlated calculations: 4s and 4p main group elements and first row additions.
The Journal of Chemical Physics, 151(14):144110, Oct. 2019.
-  G. Wang, A. Annaberdiyev, and L. Mitas.
Binding and excitations in SixHy molecular systems using quantum Monte Carlo.
The Journal of Chemical Physics, 153(14):144303, Oct. 2020.
-  H. Zhou, A. Scemama, G. Wang, A. Annaberdiyev, B. Kincaid, M. Caffarel, and L. Mitas.
A quantum Monte Carlo study of systems with effective core potentials and node nonlinearities.
arXiv:2109.08653 [physics], Sept. 2021.

-  X. Zhou, H. Zhao, T. Wang, and F. Wang.
Diffusion quantum Monte Carlo calculations with a recent generation of effective core potentials for ionization potentials and electron affinities.
Physical Review A, 100(6):062502, Dec. 2019.