

# QMCPACK Workshop 2021

## Molecular Calculations

Week 4 / October 26<sup>th</sup> ,2021

Anouar Benali, [benali@anl.gov](mailto:benali@anl.gov)

[https://github.com/QMCPACK/qmc\\_workshop\\_2021](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.

# Important notes

- For full VMC and DMC background theory please refer to relevant literature and/or lecture from the week#2 of the workshop

[https://github.com/QMCPACK/qmc\\_workshop\\_2021/blob/master/week2\\_qmc\\_introduction/week2\\_qmc\\_introduction.pdf](https://github.com/QMCPACK/qmc_workshop_2021/blob/master/week2_qmc_introduction/week2_qmc_introduction.pdf)

YouTube: [https://www.youtube.com/channel/UCdca2X8NEbjX\\_oYv60vS4gA](https://www.youtube.com/channel/UCdca2X8NEbjX_oYv60vS4gA) (QMCPACK Channel)

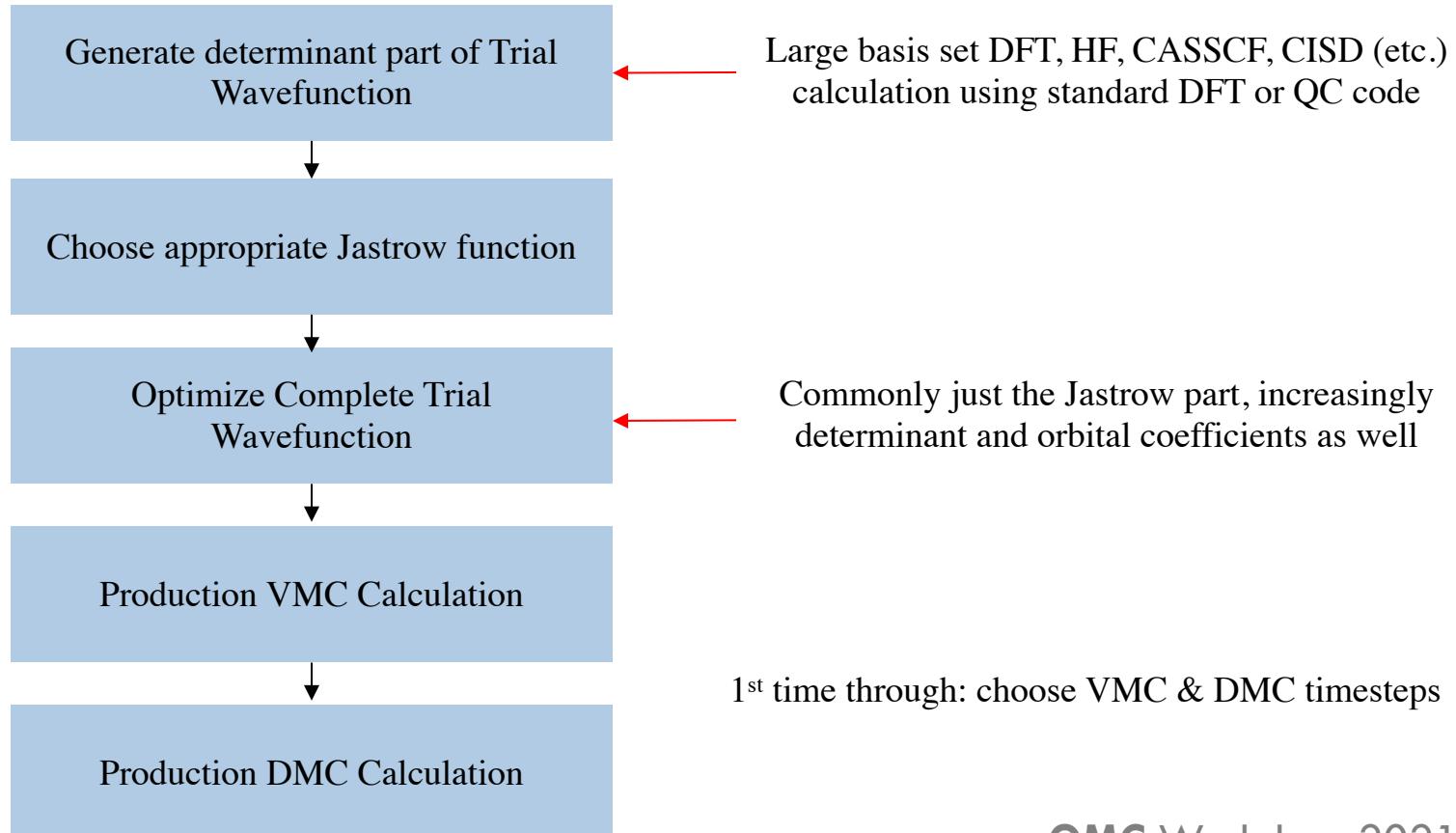
- For more details about Statistics and Nexus, refer to material from the week #3 of the workshop

[https://github.com/QMCPACK/qmc\\_workshop\\_2021/blob/master/week3\\_stats\\_and\\_nexus/week3\\_stats\\_nexus\\_vfinal.pdf](https://github.com/QMCPACK/qmc_workshop_2021/blob/master/week3_stats_and_nexus/week3_stats_nexus_vfinal.pdf)

- For more details about QMCPACK, refer to the Manual!!!

<https://qmcpack.readthedocs.io/en/develop/>

# Target of the lab: Run through QMC workflow for molecules



# The Fixed-Node Diffusion Monte Carlo method

Projector Quantum Monte Carlo methods, such as DMC, are based on the time-dependent evolution of wavefunctions and the imaginary time Schrodinger equation. Any initial state  $|\psi\rangle$ , that is not orthogonal to the ground state  $|\phi_0\rangle$ , will evolve to the ground state in the long time limit

$$\frac{\partial |\psi\rangle}{\partial \tau} = -\hat{H}|\psi\rangle, \quad \lim_{\tau \rightarrow \infty} |\psi(\tau)\rangle = c_0 e^{-\epsilon_0 \tau} |\phi_0\rangle$$

To avoid reaching the bosonic solution, we introduce a guiding/trial function  $\psi_G(\mathbf{R})$ , which closely approximates the ground state.

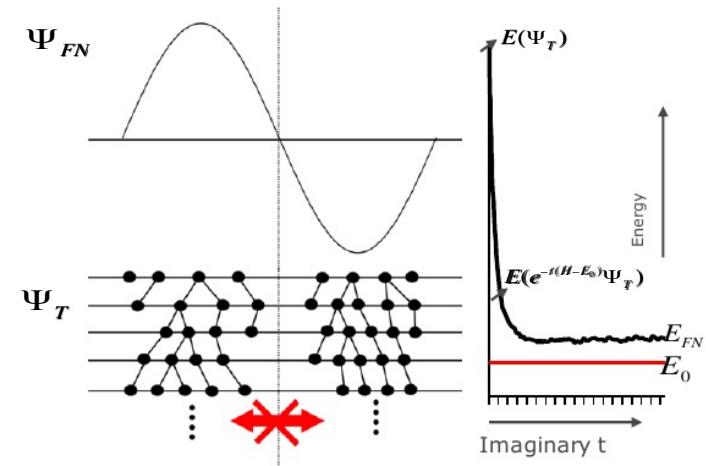
$$f(\mathbf{R}, \tau) = \psi_G(\mathbf{R})\psi(\mathbf{R}, \tau)$$

DMC is variational, and upper bound to the exact nodal surface

$$-\frac{\delta f(\mathbf{R}, \tau)}{\delta \tau} = \left[ \sum_{i=1}^N -\frac{1}{2} \nabla_i^2 f(\mathbf{R}, \tau) \right] - \nabla \cdot \left[ \frac{\nabla \psi(\mathbf{R})}{\psi(\mathbf{R})} f(\mathbf{R}, \tau) \right] + (E_L(\mathbf{R}) - E_T) f(\mathbf{R}, \tau)$$

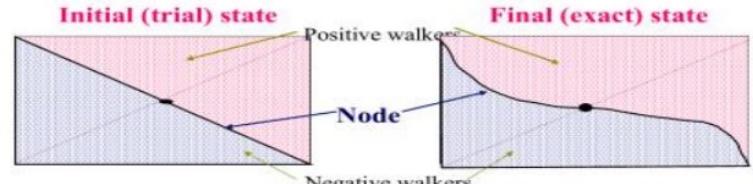
Diffusion                      Drift                      Branching

Quality of the calculation depends on the quality of nodal surface. Can be systematically improved (by increasing complexity of the guiding function)



## Model fermion problem: Particle in a box

Symmetric potential:  $V(\mathbf{r}) = V(-\mathbf{r})$   
Antisymmetric state:  $f(\mathbf{r}) = -f(-\mathbf{r})$



# Single Particle Orbitals

## Linear Combination of Atomic Orbitals (LCAO)

Generate a trial Wavefunction (DFT, HF, MCSCF, CASSCF, sCI, etc..)

$$D_k^\sigma = \begin{vmatrix} \phi_1(r_1) & \cdots & \phi_1(r_{N^\sigma}) \\ \vdots & \ddots & \vdots \\ \phi_{N^\sigma}(r_1) & \cdots & \phi_{N^\sigma}(r_{N^\sigma}) \end{vmatrix} \quad \begin{aligned} \phi_i &= \sum_l^{l=N_b} C_l^i \Phi_l \\ &\text{Single-particle orbitals} \end{aligned}$$

QMCPACK implements linear combination of atomic orbitals (LCAO) and Gaussian basis sets in (non)periodic boundary conditions.

$$\phi(\mathbf{r}) = R_l(r)Y_{lm}(\theta, \phi)$$

where  $Y_{lm}(\theta, \Phi)$  is a spherical harmonic, l and m are the angular momentum and its z component, and r,  $\theta$ ,  $\Phi$  are spherical coordinates.

Various codes are supported natively (through Nexus):

- PySCF, Gamess, Quantum Package, Quantum Espresso (Molecule in a box – but not LCAO)

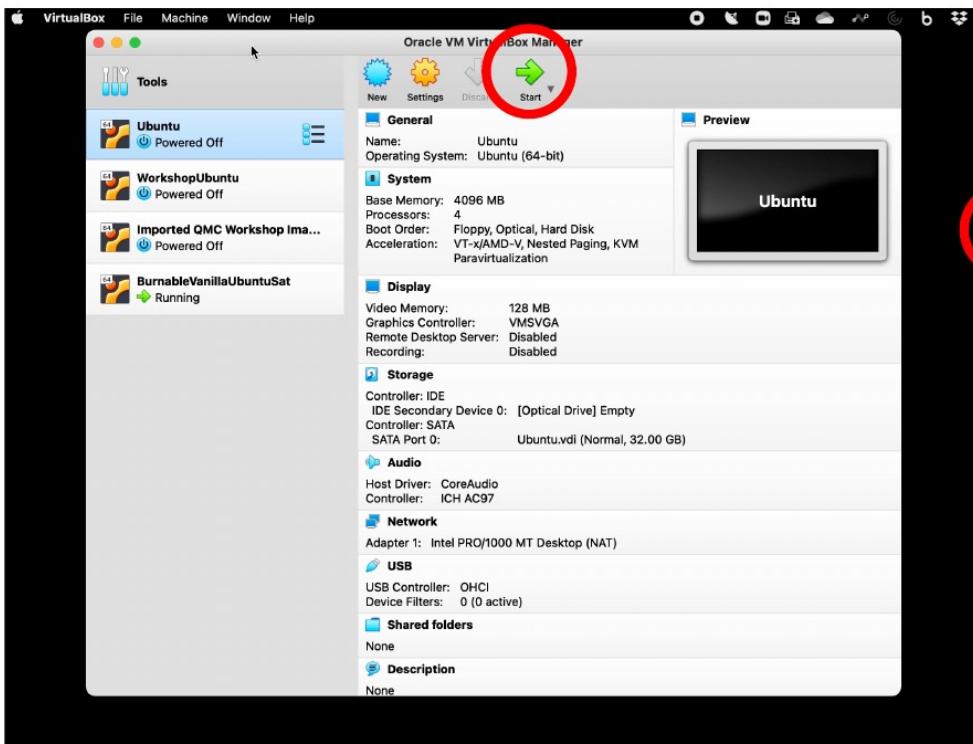
Large number of codes supported through converters (molden2qmc: <https://github.com/QMCPACK/molden2qmc>

- TURBOMOLE, PSI4, CFOUR, ORCA, DALTON, MOLPRO, NWChem, QCHEM  
(generate a Molden output through this files, then calling molden2qmc as described in the tutorial)

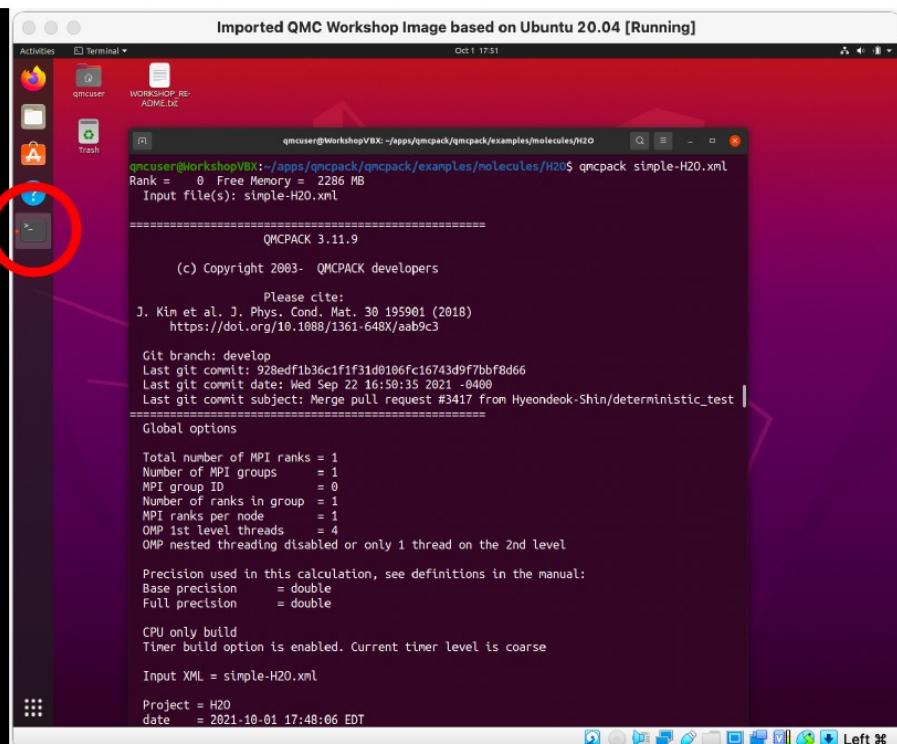
**Workshop Week 5: QMC for solids, Use of plane wave SPOs. Can also run “molecule in a box”**

# Starting the Lab: Virtual Machine

Start the Virtual Machine



Open a Terminal



# Starting the Lab: File Update & Navigation

Update workshop example files:

```
cd $HOME/qmc_workshop_2021  
git pull  
cd workshop_image  
sh ubuntu_setup.sh
```

Enter week 4 lab directory:

```
cd $HOME/qmc_workshop_2021/week4_molecules
```

We will make intensive use of the Nexus framework. If you missed week3 tutorial, I recommend reviewing it on our YouTube channel, and running the lab before running this one!

# Questions we answer in this lab:

- How to choose a trial wavefunction/method
- How to choose a basis set
- How to choose/optimize the Jastrow function
- How to select and correct for time steps in DMC
- How to manage population bias in DMC
- How to reduce the error bar in DMC
- How to systematically reduce the fixed-node approximation

# Application example: Beryllium dimer

AIP The Journal of Chemical Physics



SUBMIT YOUR ARTICLE

HOME BROWSE INFO FOR AUTHORS COLLECTIONS

SIGN UP FOR ALERTS

Home > The Journal of Chemical Physics > Volume 143, Issue 8 > 10.1063/1.4929351

Full . Published Online: 26 August 2015 Accepted: August 2015



PREV NEXT

## Quantum Monte Carlo calculation of the binding energy of the beryllium dimer

J. Chem. Phys. 143, 084116 (2015); <https://doi.org/10.1063/1.4929351>

Michael J. Deible, Melody Kessler, Kevin E. Gasperich, and Kenneth D. Jordan

View Affiliations

PDF

ABSTRACT

FULL TEXT

FIGURES

CITED BY

TOOLS

SHARE

METRICS

### TOPICS

- Slater determinant
- Chemical elements
- Heterocyclic compounds
- Correlation-consistent basis sets
- Complete-active-space
- Self consistent field methods
- Configuration interaction
- Hybrid density functional calculations
- Density functional theory
- Monte Carlo methods

### ABSTRACT

The accurate calculation of the binding energy of the beryllium dimer is a challenging theoretical problem. In this study, the binding energy of  $\text{Be}_2$  is calculated using the diffusion Monte Carlo (DMC) method, using single Slater determinant and multiconfigurational trial functions. DMC calculations using single-determinant trial wave functions of orbitals obtained from density functional theory calculations overestimate the binding energy, while DMC calculations using Hartree-Fock or CAS(4,8), complete active space trial functions significantly underestimate the binding energy. In order to obtain an accurate value of the binding energy of  $\text{Be}_2$  from DMC calculations, it is necessary to employ trial functions that



TABLE I. Total energies of Be and  $\text{Be}_2$  and the  $\text{Be}_2$  dissociation energy computed with DMC using various trial functions.

Trial function <sup>a</sup>	Total energy (a.u.)		
	Be <sup>b</sup>	$\text{Be}_2$	$D_e$ ( $\text{cm}^{-1}$ )
HF/QZ- <i>g</i>	-14.657 30(4)	-29.317 89(6)	724(21)
LDA/QZ- <i>g</i>	-14.657 21(4)	-29.319 77(7)	1174(25)
PBE/QZ- <i>g</i>	-14.657 31(5)	-29.319 60(8)	1094(26)
BLYP/QZ- <i>g</i>	-14.657 25(4)	-29.319 56(8)	1113(26)
B3LYP/QZ- <i>g</i>	-14.657 27(3)	-29.319 46(8)	1079(23)
PBE0/QZ- <i>g</i>	-14.657 28(3)	-29.319 07(8)	992(21)
BH&HLYP/QZ- <i>g</i>	-14.657 26(5)	-29.318 91(7)	966(26)
BD/QZ- <i>g</i>	-14.657 18(4)	-29.318 72(7)	955(24)
CAS(4,8)/QZ- <i>fg</i> <sup>c</sup>	-14.667 23(1)	-29.337 07(3)	573(8)
CAS(4,16)/QZ- <i>fg</i> <sup>c</sup>	-14.667 30(1)	-29.338 32(3)	819(8)
Ext. CAS(4,16)/QZ- <i>fg</i>	-14.667 30(1)	-29.338 41(2)	838(7)
CAS(4,16)/QZ- <i>g</i> <sup>c</sup>	-14.667 27(2)	-29.338 38(3)	845(8)
Ext. CAS(4,16)/QZ- <i>g</i>	-14.667 27(2)	-29.338 45(2)	857(9)
CI/QZ- <i>g</i> <sup>c</sup>	-14.667 25(1)	-29.338 48(2)	873(6)
Ext. CI/QZ- <i>g</i>	-14.667 25(1)	-29.338 64(2)	908(6)
Experimental <sup>d</sup>	-14.667 356	-29.338 97	934.9(4)

<sup>a</sup>QZ refers to the cc-pVQZ basis set. The “-*g*” and “-*fg*” indicate, respectively, that the *g* functions and *f* and *g* functions were omitted from the basis sets. Ext. refers to CAS and CI results extrapolated to the full configuration space for the active orbital list as described in the text.

<sup>b</sup>The DMC energies of the Be atom calculated using various single determinant trial functions should agree. The spread of the energies in the table is the result of statistical errors and the use of a finite (0.001 a.u.) time step.

<sup>c</sup>0.001 threshold on CI coefficients for retained configurations.

<sup>d</sup>The experimental  $D_e$  value for  $\text{Be}_2$  is from Ref. 5. The non-relativistic energy of the Be atom is from Ref. 60.

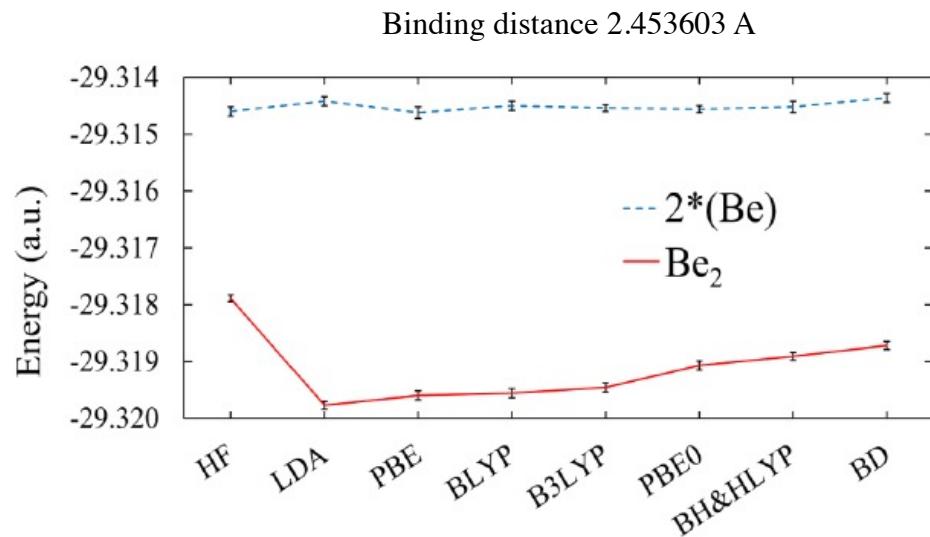


FIG. 3. DMC energies of twice the beryllium atom and the dimer for several single-determinant trial wave functions.

# Outline

## I- Single determinant, all electrons calculations:

- Setting up an all-electrons Calculation with Nexus (DFT)
  - Generating multiple Trial wavefunction (HF, DFT:LDA, GGA, Hybrids)
  - Effects of basis-sets on the trial wavefunction (cc-pcvXz | X=D,T,Q)
- QMC Calculations
  - Wavefunction conversion
  - Cusp Correction
  - Jastrow Optimization
  - Effect of a Jastrow function on VMC and DMC calculation
- Controlling the DMC approximations
  - Time-step Error
  - Population bias
  - Reducing error bars
- Dependence of DMC to the trial wavefunction (SD-DMC)



Total 8e<sup>-</sup> (Be:4e<sup>-</sup>)

Experimental<sup>1</sup> total energy:-29.33897 Ha

Bond length<sup>1</sup> : 2.453603 Å

# Setting up an all-electrons Calculation with Nexus (DFT)

Varying the trial wavefunction (modifying the nodes – SCF study)

$$\Psi_T(R) = J(R)\Psi_{AS}(R) = e^{J_1+J_2+\dots} \sum_k^M C_k D_k^\uparrow(\phi) D_k^\downarrow(\phi)$$

Nodes (HF, DFT, Basis-set etc..)

We will see how close to the exact solution, some mean-field solutions are:

- We will try HF, DFT{LDA, PBE, PBE0, SCAN}
- For each method/functional, we will use 3 basis-sets: cc-pvXz | X=d,t,q

# Using PySCF

File located in: ~/week4\_molecules/00\_dft/Be2.py

- Using python3
- From PySCF Importing modules for density fitting (DF), self consistent calculations (SCF) and density Functional theory (DFT)
- Importing Gaussian type orbitals (GTO)
- Defining Atoms and coordinates (different ways to do)
- Defining a basis set, units, charge and spin.
- Defining the method:
  - RHF Restricted Hartree Fock
  - ROHF Restricted Open-shell Hartree Fock
  - UHF unrestricted Hartree Fock
  - RKS Restricted Kohn-Sham (DFT, requires specifying XC functional)
  - ROKS and UKS: Restricted Open-shell Kohn-Sham and Unrestricted Kohn-Sham (DFT, requires specifying XC functional)
- Running simulation (mf.kernel)
- Save to QMCPACK format:
  - Outside VM requires to add the location of QMCPACK script PyscfToQMCPACK.py to your PYTHONPATH
  - export PYTHONPATH=/mypath/qmcpack/src/QMCTools/:\$PYTHONPATH

```
#!/usr/bin/env python3

import numpy as np
from numpy import array

## generated pyscfimport text ##
from pyscf import df, scf, dft
## end generated pyscfimport text ##

## generated system text ##
from pyscf import gto as gto_loc
mol = gto_loc.Mole()
mol.verbose = 5
mol.atom = '''
    Be   0.00000000  0.00000000  0.00000000
    Be   2.45360300  0.00000000  0.00000000
    '''
mol.basis = 'cc-pvtz'
mol.unit = 'A'
mol.charge = 0
mol.spin = 0
mol.symmetry = True
mol.build()
## end generated system text ##

## generated calculation text ##
mf = scf.ROHF(mol).density_fit()
mf.max_cycle=200
mf.level_shift=0.0
mf.tol = '1e-10'
e_scf = mf.kernel()
## end generated calculation text ##

## generated conversion text ##
from PyscfToQmcpack import savetoqmcpack
savetoqmcpack(mol,mf,'scf')
## end generated conversion text ##
```

# Using Nexus

- We need to import the module we will need for the simulation:
  - PySCF, physical system, multiple QMC sections we will need.
- Settings:
  - Path for the result
  - Seconds to sleep before checking if job is completed
  - Machine: type of workstation (Nexus can submit jobs to HPC platforms such as Theta at ANL or Summit at ORNL)
- Physical system:
  - Can read geometry from: xyz file, CIF file, raw coordinates, POSCAR file, etc..
- SCF run:
  - Recognizes all methods and “most” pyscf parameters. (please contact developers if a parameter is missing)
  - Note that the results will be written in the *path* section

```
grep "converged SCF energy =" runs/Be2/cc-pvtz/HF/SCF/scf.out
converged SCF energy = -29.1338009143752 Ha
```

File located in: week4\_molecules/00\_dft/nx\_Be2\_scf.py

```
#!/usr/bin/env python3

from nexus import settings, job, run_project, obj
from nexus import generate_physical_system
from nexus import generate_pyscf

# Obtain the core count of the local machine (lab only)
import os
cores = os.cpu_count()

settings(
    results      = '',
    sleep        = 3,
    machine     = 'ws'+str(cores),
)

system = generate_physical_system(
    structure = 'Be2.xyz',           # Be2 geometry located in the Be2.xyz file
    )

# perform Hartree Fock in all electron and a cc-pvtz basis set
scf = generate_pyscf(
    identifier = 'scf',            # log output goes to scf.out
    path       = 'Be2/cc-pvtz/HF/SCF', # directory to run in
    job        = job(serial=True, app='python3'), # pyscf must run serially
    system     = system,
    mole       = obj(                # used to make Mole() inputs
        basis   = 'cc-pvtz',
        symmetry = True,
        verbose  = 5,
    ),
    calculation = obj(
        method      = 'ROHF',          # Restricted Orbital Hartree Fock
        df_fitting  = True,           # Density fitting
        max_cycle   = 200,             # Max SCF cycles
        level_shift = 0.0,             # Mixing orbitals for convergence
        tol         = '1e-10',          # Accuracy needed for convergence
    ),
    save_qmc   = True,              # Save the orbital to QMCPACK format
)

run_project()
```

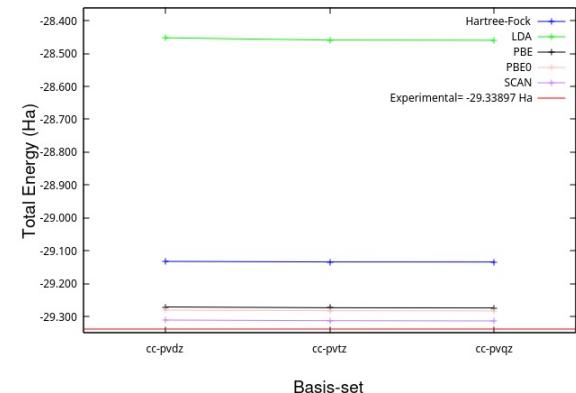
# Plotting/analyzing your data (gnuplot)

- We will use a simple script to extract the energies from the output files using the qmca binary and plot them using gnuplot
- Script name is Extract\_data\_scf.sh (will parse the output and put the energies in a correct format)
- Type:

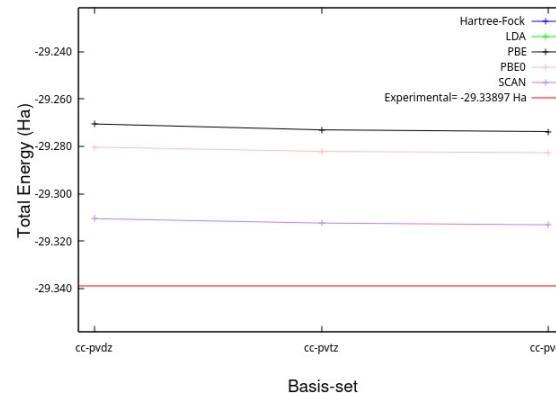
```
sh Extract_data_scf.sh  
gnuplot Be2_TotEnergy_SCF.gp; gnuplot Be2_EnergyDiff_SCF.gp
```

#Method	HF	LDA	PBE	PBE0	SCAN
cc-pvdz	-29.1323298263028	-28.4524632697256	-29.2706072146257	-29.2803582209562	-29.3105780453077
cc-pvtz	-29.1338009143752	-28.4592510398411	-29.2730701036598	-29.2822041815256	-29.3124577762556
cc-pvqz	-29.1340587491698	-28.4598638073636	-29.2737871836446	-29.2827419339206	-29.3131579603851

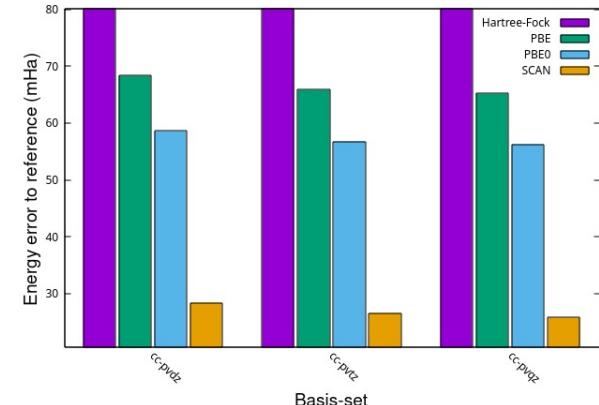
Be<sub>2</sub> Total Energy as a function of the method and basis-set



Be<sub>2</sub> Total Energy as a function of the method and basis-set



Be<sub>2</sub> Total Energy as a function of the method and basis-set



- Large dependence on the functional

- SCAN particularly good but still not exact (25mHa)

# QMC Calculations

1- Choice of an Anti Symmetric wavefunction

$$\Psi_T(R) = J(R) \Psi_{AS}(R) = e^{J_1 + J_2 + \dots} \sum_k^M C_k D_k^\uparrow(\phi) D_k^\downarrow(\phi)$$

2- Converting/extracting QMC data from PySCF output

3- Correcting orbitals for electron-nuclei cusp (all electron calculations only): **Variance analysis**

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions: **Variance Analysis**

5- Run Diffusion Monte Carlo

- Correct for time-step error
- Control population bias
- Reducing error bars

# QMC Calculations

## 1- Choice of an Anti Symmetric wavefunction

$$\Psi_T(R) = J(R)\Psi_{AS}(R) = e^{J_1+J_2+\dots} \sum_k^M C_k D_k^\uparrow(\phi) D_k^\downarrow(\phi)$$

Nexus will check the directory path; if data is available and status is successful, the module will be skipped

This will generate an HDF5 file containing all data needed by QMCPACK

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc0.py

```
#! /usr/bin/env python3
from nexus import settings, job, run_project, obj
from nexus import ppset
from nexus import generate_physical_system
from nexus import generate_pyscf
from nexus import generate_convert4qmc
from nexus import generate_cusp_correction
from nexus import generate_qmcpack

# Obtain the core count of the local machine (lab only)
import os
cores = os.cpu_count()

settings(
    results      = '',
    sleep        = 3,
    machine     = 'ws'+str(cores),
)

system = generate_physical_system(
    structure = 'Be2.xyz',                      # Be2 geometry located in the Be2.xyz file
    )

XC=["SCAN"]
MyBasis=["cc-pvtz"]
for x in XC:
    for y in MyBasis:
        # perform DFT
        scf = generate_pyscf(
            identifier = 'scf',
            path       = 'Be2/'+y+'/'+x+'/SCF',          # directory to run in
            job        = job(serial=True,app='python3'), # pyscf must run serially
            system     = system,
            mole       = obj(
                basis   = y,
                symmetry = True,
                verbose  = 5,
            ),
            calculation = obj(
                method   = 'ROKS',
                df_fitting = True,
                max_cycle = 200,
                level_shift = 0.0,
                tol       = '1e-10',
                xc        = x,
            ),
            save_qmc  = True,                            # Save the orbital to QMCPACK format
        )
        #Loop around multiple XC functionals
        #Loop around multiple basis sets
```

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc0.py

1- Choice of an Anti Symmetric wavefunction

2- Converting/extracting QMC data from PySCF output

- Nexus needs to know that it is an all electron calculation and we will want to correct for the electron-nuclei cusp, and where are the orbitals (scf tag).

```
# convert orbitals to QMCPACK format
c4q = generate_convert4qmc(
    identifier      = 'c4q',
    path            = 'Be2/' + y + '/' + x + '/SCF',           # directory to run in
    job             = job(cores=1),
    add_cusp        = True,
    dependencies   = (scf, 'orbitals'),                      # Make sure to add Cusp Correction scheme
)                                                               # Create a dependency to DFT success
```

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc0.py

1- Choice of an Anti Symmetric wavefunction

2- Converting/extracting QMC data from PySCF output

3- Correcting orbitals for electron-nuclei cusp (all electron calculations only)

- At the nuclei, Gaussian basis sets are unable to describe the cusps in the single-particle orbitals (zero gradient at the nuclei on which they are centered).
- In VMC/DMC, the potential energy contribution to  $E_L$  diverges as  $-Z/r$  when  $r \sim 0$ . We need equal and opposite divergence in local kinetic energy.
- Using Gaussian Basis set, the kinetic energy is finite at the nucleus and  $E_L$  diverges.
- The solution is to modify each molecular orbitals to obey the cusp condition at each nucleus; replacement of the orbital inside some small radius by a well-behaved form.

```
# calculate cusp correction
cc = generate_cusp_correction(
    identifier = 'cusp',
    path       = 'Be2/' + y + '/' + x + '/SCF',           # directory to run in
    job        = job(cores=cores),
    system     = system,
    dependencies = (c4q, 'orbitals'),                   #Create dependency to converter
)
```

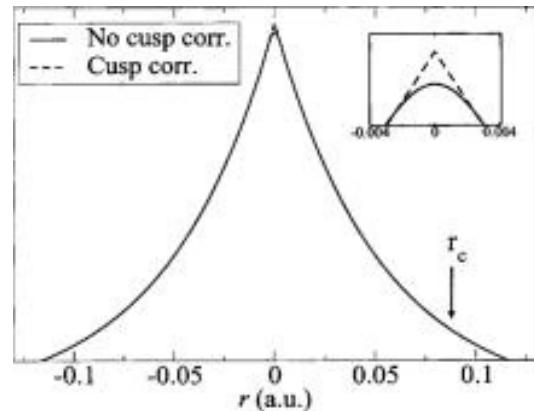


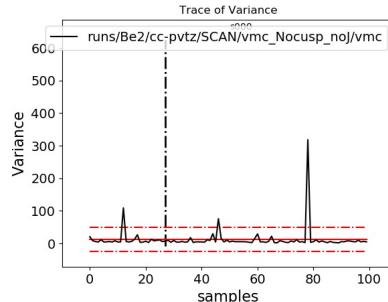
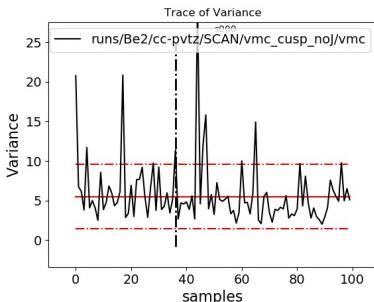
FIG.1 The 1s orbital of the Ne atom expanded in a Gaussian basis set with and without the cusp correction

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc0.py

## 3- Correcting orbitals for electron-nuclei cusp (all electron calculations only)

Effect of Cusp correction can be seen by running both ***nx\_Be2\_qmc0.py*** and ***nx\_Be2\_qmc0\_NoCusp.py*** files



```
# run VMC with cusp Correction and no Jastrow function
qmc = generate_qmcpack(
    identifier = 'vmc',
    path       = 'Be2/' + y + '/' + x + '/vmc_cusp_noJ', # directory to run in
    job        = job(cores=cores),
    system     = system,
    jastrows   = [],
    qmc        = 'vmc',
    seed        = 42,
    warmupsteps = 10,
    blocks      = 100,
    steps       = 10,
    timestep    = 0.1,
    dependencies = orbdeps,
)
run_project()
```

```
qmca -q ev runs/Be2/cc-pvtz/SCAN/vmc_Nocusp_noJ/*scalar.dat runs/Be2/cc-pvtz/SCAN/vmc_cusp_noJ/*.scalar.dat
```

```
runs/Be2/cc-pvtz/SCAN/vmc_Nocusp_noJ/vmc  series 0  -29.129717 +/- 0.048947
runs/Be2/cc-pvtz/SCAN/vmc_cusp_noJ/vmc    series 0  -29.105796 +/- 0.031162
```

LocalEnergy	Variance	ratio
series 0 -29.129717 +/- 0.048947	12.191076 +/- 4.310170	0.4185
series 0 -29.105796 +/- 0.031162	5.519650 +/- 0.503832	0.1896

Significant effect on the variance!!

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc0.py

- 1- Choice of an Anti Symmetric wavefunction
- 2- Converting/extracting QMC data from PySCF output
- 3- Correcting orbitals for electron-nuclei cusp (all electron calculations only)
- \*- Intermediate step

```
# collect dependencies relating to orbitals
orbdeps = [(c4q,'particles'),
            (c4q,'orbitals' ),  
           (cc,'cuspcorr' )] # pyscf changes particle positions
```

All non-QMC dependencies are now known to nexus and will be carried through various nexus blocks through the tag *orbdeps*

# QMC Calculations

More details in QMCPACK manual

1- Choice of an Anti Symmetric wavefunction

2- Converting/extracting QMC data from PySCF output

3- Correcting orbitals for electron-nuclei cusp (all electron calculations only)

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

$$\Psi_T(R) = J(R)\Psi_{AS}(R) = e^{J_1+J_2+\dots} \sum_k^M C_k D_k^\uparrow(\phi) D_k^\downarrow(\phi)$$

$$J_1 = \int_I^{ion0} \sum_i^e u_{ab}(|r_i - R_I|)$$

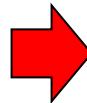
$$J_2 = \sum_i^e \sum_{j>i}^e u_{ab}(|r_i - r_j|)$$

$$u_{\sigma\sigma'I}(r_{\sigma I}, r_{\sigma' I}, r_{\sigma\sigma'}) = \sum_{\ell=0}^{M_{el}} \sum_{m=0}^{M_{el}} \sum_{n=0}^{M_{ee}} \gamma_{\ell m n} r_{\sigma I}^\ell r_{\sigma' I}^m r_{\sigma\sigma'}^n \times \left(r_{\sigma I} - \frac{r_c}{2}\right)^3 \Theta\left(r_{\sigma I} - \frac{r_c}{2}\right) \times \left(r_{\sigma' I} - \frac{r_c}{2}\right)^3 \Theta\left(r_{\sigma' I} - \frac{r_c}{2}\right)$$

- $u_{ab}$  is an interpolating 1D B-spline (tricubic spline in a linear grid) between zero distance and  $r_{cut}$ .
- The cusp condition as  $r_i$  approaches  $r_j$  is set by the relative spin of the electrons
- MeI and Mee are the maximum polynomial orders of the electron-ion and electron-electron distances, respecttively,  $\{\gamma_{lmn}\}$  are the optimizable parameters (modulo constraints),  $r_c$  is a cutoff radius, and  $r_{cut}$  are the distances between electrons or ions  $a$  and  $b$ . i.e. The correlation function is only a function of the interparticle distances and not a more complex function of the particle positions,  $r$ . As indicated by the  $\Theta$  functions, correlations are set to zero beyond a distance of  $r_c/2$  in either of the electron-ion distances and the largest meaningful electron-electron distance is  $r_c$ .

# QMC Calculations

Change of file



File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We optimize the J1 and J2 parameters

```
# optimize 2-body Jastrow
optJ2 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'Be2/' + yt + '/' + xt + '/optJ2',           # directory to run in
    job             = job(cores=cores),
    system          = system,
    J2              = True,           # 2-body B-spline Jastrow
    J1_rcut         = 6.0,            # 6 Bohr cutoff for J1
    J2_rcut         = 8.0,            # 8 Bohr cutoff for J2
    seed            = 42,             # Fix the seed (lab only)
    qmc             = 'opt',          # Wavefunction optimization run
    minmethod       = 'oneshift',     # Energy minimization
    init_cycles     = 4,              # 4 iterations allowing larger parameter changes
    cycles          = 8,              # 8 production iterations
    warmupsteps     = 10,
    blocks          = 20,
    steps           = 3,
    timestep        = 0.1,
    init_minwalkers = 0.1,
    minwalkers      = 0.5,
    samples          = 25600,          # VMC samples per iteration
    dependencies    = orbdeps,
)
```

runs/Be2/cc-pvtz/SCAN/optJ2/opt.in.xml

```
<jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
    <correlation elementType="Be" size="12" rcut="6.0" cusp="0.0">
        <coefficients id="eBe" type="Array">
0 0 0 0 0 0 0 0 0 0 0 0
        </coefficients>
    </correlation>
</jastrow>
<jastrow type="Two-Body" name="J2" function="bspline" print="yes">
    <correlation speciesA="u" speciesB="u" size="16" rcut="8.0">
        <coefficients id="uu" type="Array">
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        </coefficients>
    </correlation>
    <correlation speciesA="u" speciesB="d" size="16" rcut="8.0">
        <coefficients id="ud" type="Array">
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        </coefficients>
    </correlation>
</jastrow>
```

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We optimize the J1 and J2 parameters (stochastic optimization via VMC; lower energy/variance means better parameters)

```
cd runs/Be2/cc-pvtz/SCAN/optJ2/
qmca -q ev *.scalar.dat | sort -k4
```

	LocalEnergy	Variance	ratio
opt series 0	-29.152415 +/- 0.009964	2.094150 +/- 0.020768	0.0718
opt series 2	-29.253657 +/- 0.004868	0.413195 +/- 0.005872	0.0141
opt series 1	-29.259555 +/- 0.007212	0.375314 +/- 0.006928	0.0128
opt series 5	-29.264656 +/- 0.006250	0.461682 +/- 0.006575	0.0158
opt series 6	-29.265019 +/- 0.004714	0.463221 +/- 0.008596	0.0158
opt series 9	-29.266346 +/- 0.008183	0.492715 +/- 0.005705	0.0168
opt series 11	-29.267270 +/- 0.004427	0.494253 +/- 0.004432	0.0169
opt series 8	-29.267562 +/- 0.005051	0.503567 +/- 0.007467	0.0172
opt series 3	-29.268296 +/- 0.006235	0.446392 +/- 0.005157	0.0153
opt series 4	-29.268767 +/- 0.006669	0.457073 +/- 0.007368	0.0156
opt series 7	-29.270653 +/- 0.004499	0.481380 +/- 0.004861	0.0164
opt series 10	-29.271025 +/- 0.004992	0.497263 +/- 0.005199	0.0170

```
<?xml version="1.0"?>
<qmcsystem>
  <wavefunction name="psi0" target="e">
    <determinantset type="MolecularOrbital" href="../SCF/scf.h5" source="ion0" transform="yes" name="LCAOBSet" cuspCorrection="yes">
      <slaterdeterminant>
        <determinant id="updet" size="4" cuspInfo="../SCF/updet.cuspInfo.xml">
          <occupation mode="ground">
            <coefficient size="60" spindataset="0">
              </occupation>
            </coefficient>
          </determinant>
        <determinant id="downdet" size="4" cuspInfo="../SCF/downdet.cuspInfo.xml">
          <occupation mode="ground">
            <coefficient size="60" spindataset="0">
              </occupation>
            </coefficient>
          </determinant>
        </slaterdeterminant>
      </determinantset>
      <jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
        <correlation elementType="Be" size="12" rcut="6.0" cusp="0.0">
          <coefficients id="eBe" type="Array" -0.1941756812 -0.85382965 -0.7409253541 -0.639286145 -0.5468855822 -0.4575510581
-0.3672189612 -0.2742578036 -0.1200012388 -0.06098851466</coefficients>
        </correlation>
      </jastrow>
      <jastrow type="Two-Body" name="J2" function="bspline" print="yes">
        <correlation species="u" species="u" size="16" rcut="8.0">
          <coefficients id="uu" type="Array" 0.286494837 -0.2002724141 0.155101472 0.1186741914 0.09128333231 0.07007304412 0.05148561169 0.
03365197704 0.15515622838 -0.008369036969 -0.01378501028 -0.02668143697 -0.03437987653 -0.0273251815 -0.0193908639 -0.01321158763</coefficients>
        </correlation>
        <correlation species="u" species="d" size="16" rcut="8.0">
          <coefficients id="ud" type="Array" 0.6973392418 0.5437740857 0.4535660295 0.3935979623 0.3382767646 0.2900516436 0.2437848858 0.1
993285683 0.1571844165 0.1192336806 0.08739279614 0.05667843986 0.03326895371 0.02371499997 0.0157137336 0.009161590998</coefficients>
        </correlation>
      </jastrow>
    </wavefunction>
  </qmcsystem>
```

- Lowest energy achieved at series 010, best parameters optimized in iteration 009: **opt.s009.opt.xml**
- Note the reduction of he variance as the energy goes down.
- Ratio variance/energy gives good judgment on the variance ( lower than 0.02 is a very good ratio)

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We optimize the J3 parameters after J1 and J2 are optimized

```
# optimize 3-body Jastrow
optJ3 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'Be2/'+'y+'+'/'+xx+'/optJ3',           # directory to run in
    job             = job(cores=cores),
    system          = system,
    J3              = True,          # 3-body B-spline Jastrow
    seed            = 42,            # Fix the seed (lab only)
    qmc             = 'opt',          # Wavefunction optimization run
    minmethod       = 'oneshift',   # Energy minimization
    init_cycles     = 4,             # 4 iterations allowing larger parameter changes
    cycles          = 8,             # 8 production iterations
    warmupsteps     = 10,
    blocks          = 20,
    steps           = 5,
    timestep        = 0.1,
    init_minwalkers = 0.1,
    minwalkers      = 0.5,
    samples          = 25600,         # VMC samples per iteration
    dependencies    = orbdeps+[(optJ2,'jastrow')],
)
```

runs/Be2/cc-pvtz/SCAN/optJ3/opt.in.xml

```
<jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
  <correlation elementtype="Be" size="12" rcut="6.0" cusp="0.0">
    <coefficients id="eBe" type="Array">
      -1.036692517 -0.9491756812 -0.85382965 -0.7409253541 -0.639286145 -0.5468855822
      -0.4575510581 -0.3672189612 -0.2742578036 -0.1941760504 -0.1200012388 -0.06098051466
    </coefficients>
  </correlation>
</jastrow>
<jastrow type="Two-Body" name="J2" function="bspline" print="yes">
  <correlation speciesA="u" speciesB="u" size="16" rcut="8.0">
    <coefficients id="uu" type="Array">
      0.286494837 0.2002724141 0.155101472 0.1186741914 0.09128333231 0.07007304412
      0.05148561169 0.03365197704 0.01551622838 -0.000369036969 -0.01378501028
      -0.02668143697 -0.03437087653 -0.0273251815 -0.0193908639 -0.01321158763
    </coefficients>
  </correlation>
  <correlation speciesA="u" speciesB="d" size="16" rcut="8.0">
    <coefficients id="ud" type="Array">
      0.6973392418 0.543740857 0.4535660295 0.3935979623 0.3382767646 0.2900516436
      0.2437848858 0.1993285683 0.1571844165 0.1192336866 0.08739279614 0.05667843986
      0.03326895371 0.02371499997 0.0157137336 0.009161590998
    </coefficients>
  </correlation>
</jastrow>
<jastrow type="eeI" name="J3" function="polynomial" print="yes" source="ion0">
  <correlation ispecies="Be" especies1="u" especies2="u" isize="3" esize="3" rcut="5.0">
    <coefficients id="uuBe" type="Array" optimize="yes"/>
  </correlation>
  <correlation ispecies="Be" especies1="u" especies2="d" isize="3" esize="3" rcut="5.0">
    <coefficients id="udBe" type="Array" optimize="yes"/>
  </correlation>
</jastrow>
</wavefunction>
```

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We optimize the **J1, J2 AND J3** parameters

```
cd runs/Be2/cc-pvtz/SCAN/optJ3/
qmca -q ev *.scalar.dat | sort -k4
```

	LocalEnergy	Variance	ratio
opt series 3	-29.268254 +/- 0.007506	0.244048 +/- 0.003614	0.0083
opt series 0	-29.269403 +/- 0.004200	0.494745 +/- 0.005474	0.0169
opt series 11	-29.276331 +/- 0.004007	0.267680 +/- 0.004092	0.0091
opt series 8	-29.277155 +/- 0.005009	0.264350 +/- 0.004955	0.0090
opt series 9	-29.279309 +/- 0.004174	0.268449 +/- 0.008893	0.0092
opt series 2	-29.281779 +/- 0.004483	0.246311 +/- 0.003313	0.0084
opt series 4	-29.282044 +/- 0.004457	0.252980 +/- 0.003455	0.0086
opt series 1	-29.284109 +/- 0.005932	0.234446 +/- 0.005228	0.0080
opt series 6	-29.285399 +/- 0.009696	0.262893 +/- 0.008862	0.0090
opt series 10	-29.286640 +/- 0.003885	0.263126 +/- 0.002738	0.0090
opt series 7	-29.287853 +/- 0.003120	0.254136 +/- 0.004671	0.0087
opt series 5	-29.288128 +/- 0.004488	0.247736 +/- 0.003564	0.0085

```
?xml version="1.0"?
<qmcSystem>
  <wavefunction name="psi0" target="e">
    <determinant type="MolecularOrbital" href="../SCF/scf.h5" source="ion0" transform="yes" name="LCAOBSet" cuspCorrection="yes">
      <slaterdeterminant>
        <determinant id="updet" size="4" cuspInfo="../SCF/updet.cuspInfo.xml">
          <occupation mode="ground">
            <coefficient size="60" spinDataset="0">                                         </occupation>
          </determinant>
        <determinant id="downdet" size="4" cuspInfo="../SCF/downdet.cuspInfo.xml">
          <occupation mode="ground">
            <coefficient size="60" spinDataset="0">                                         </occupation>
          </determinant>
        </slaterdeterminant>
      </determinantset>
      <jastrow type="One-Body" name="J1" function="bspline" source="ion0" print="yes">
        <correlation elementType="Be" size="12" rcut="6.0" cusp="0.0" >
          <coefficients id="uBe" type="Array" >-1.299983246 -1.183412295 -1.089573911 -0.8924886749 -0.7590911094 -0.539570117
5 -0.4614140423 -0.3606319321 -0.2747886299 -0.1932344795 -0.1203174429 -0.06144056575</coefficients>
        </correlation>
      </jastrow>
      <jastrow type="Two-Body" name="J2" function="bspline" print="yes">
        <correlation species="u" species="u" size="16" rcut="8.0" >
          <coefficients id="uBe" type="Array" >0.31429523083 0.1776826558 0.1338254218 0.1822842678 0.08805437084 0
.06236669886 0.0444036629 0.02761499318 0.01192522851 -0.001028569266 -0.01398351476 -0.0222915965 -0.01730293957 -0.0124356745 -0
.008518647734</coefficients>
        </correlation>
        <correlation species="u" species="d" size="16" rcut="8.0" >
          <coefficients id="uBd" type="Array" >0.05978082065 0.7517309829 0.5699293731 0.4339576349 0.3350611063 0.2820777488 0
2423013536 0.2002422849 0.160596697 0.124562161 0.09420381054 0.06560106389 0.04332410727 0.03395272133 0.0242932223 0.01493720409
        </coefficients>
        </correlation>
      </jastrow>
      <jastrow type="eEl" name="J3" function="polynomial" print="yes" source="ion0">
        <correlation ispecies="Be" especies="u" ispecies="u" especies2="d" isize="3" esize="3" rcut="5.0" >
          <coefficients id="uube" type="Array" optimize="yes" >-0.0426845037 -1.500259898 -0.0124943211 0.001271344892 -
0.004010698539 0.0029923357 0.006468551918 0.001398783433 0.01788785613 -0.001625400806 0.001264699615 -0.01196958723 -0.00632652
5316 0.0009183695881 0.002194138537 0.0008426467211 0.0005102280486 0.002346602199 0.003363102911 0.00420270529 0.00250303643 0
.00472935109 -0.02514671044 -0.0008406753613 -0.003043595073 0.001623060765</coefficients>
        </correlation>
        <correlation ispecies="Be" especies="u" especies2="d" isize="3" esize="3" rcut="5.0" >
          <coefficients id="udBe" type="Array" optimize="yes" >-0.01276239878 -0.002606274218 0.01349269463 -0.002045781279 -
0.002332224448 -0.00211862742 -0.01211862742 -0.003148054288 0.02993132779 0.02642747618 0.002991168074 -0.01866694899 -0.01686
34921 -0.00431317429 0.01136595698 0.001804653231 -0.001181381798 0.007250884821 0.01486607966 0.008665078969 -0.00827841117 -0.0
02529213093 -0.001825210376 0.009459478566 -0.004456789435 0.00103412212</coefficients>
      </correlation>
    </wavefunction>
  </qmcSystem>
```

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We compare energies from VMC\_NoJ, VMC\_2J, VMC\_3J

		LocalEnergy	Variance	ratio
runs/Be2/cc-pvtz/SCAN/vmc_NoJ/vmc	series 0	-29.122630 +/- 0.008140	6.293397 +/- 0.650270	0.2161
runs/Be2/cc-pvtz/SCAN/vmc_2J/vmc	series 0	-29.264403 +/- 0.002655	0.497976 +/- 0.002821	0.0170
runs/Be2/cc-pvtz/SCAN/vmc_3J/vmc	■ series 0	-29.281952 +/- 0.002211	0.262596 +/- 0.002101	0.0090

- Significant improvement of the variance and energy (better trial wavefunction) when using 1, 2 and 3 body Jastrow function
- Better trial wavefunction => **lower variance** => lower error bar (error bar ~4x smaller => needs 16x more CPU time to reach similar accuracy)

# QMC Calculations

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc1.py

4- Adding 1 body (electron-ion), 2 body(electron-electron) and 3 body (electron-electron-ion) Jastrow functions

- We compare energies from DMC\_NoJ, DMC\_2J, DMC\_3J

		LocalEnergy	Variance	ratio
runs/Be2/cc-pvtz/SCAN/dmc_NoJ/dmc	series 1	-29.322531 +/- 0.002778	5.032637 +/- 0.029377	0.1716
runs/Be2/cc-pvtz/SCAN/dmc_2J/dmc	series 1	-29.319212 +/- 0.000941	0.492822 +/- 0.001771	0.0168
runs/Be2/cc-pvtz/SCAN/dmc_3J/dmc	series 1	-29.318362 +/- 0.000797	0.258312 +/- 0.001667	0.0088

- In All electron calculations, adding a Jastrow function improves significantly the variance, but eventually energies converges to the same value

Speedup of ~8X  
20X better variance

- Keep in mind!!: Difference of energy between J2 and J3 is : 0.4 +/- 1.33 mHa      Same energies!!!  
Difference of energy between NoJ and J3 is : 4.1 +/- 2.9mHa      within 2 sigma

Questions ?

# Break

# Setting up a DMC Calculation

How to set up time step, number of blocks, number of steps and number of walkers

- Extrapolate the time-step
- Control population bias
  - Reducing error bars

# Setting up a DMC Calculation

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc2.py

Goal: Extrapolate the time-step

Change of file



```
#### DMC BLOCKS FOR TIME STEP EXTRAPOLATION

# run DMC with cusp Correction and 1,2 and 3 Body Jastrow function
qmc = generate_qmcpack(
    identifier      = 'dmc',
    seed            = 42,
    path            = 'Be2/ty+/' + 'tx+' + 'dmc_tstep',           # directory to run in
    job             = job(cores=cores),                         # Submit with the number of cores available
    system          = system,
    jastrows        = [],
    qmc             = 'dmc',                                     # dmc run
    vmc_samples     = 4096,                                     # Number of Samples (selected from a VMC step)
    warmupsteps    = 50,                                       # Number of Equilibration steps
    vmc_blocks     = 200,                                      # Number of VMC blocks (To generate the DMC samples)
    vmc_steps       = 20,                                       # Number of VMC steps (To generate DMC samples)
    vmc_timestep   = 0.3,                                      # VMC Timestep (To Generate DMC samples)
    timestep        = 0.01,                                     # DMC timestep
    timestep_factor = 0.5,                                     # Reduce by 1/2
    nimesteps       = 4,                                       # 4 times, i.e. [0.01, 0.005, 0.0025, 0.00125] timesteps
    steps           = 20,                                      # start with small number for large timesteps [autocorrelation]
    blocks          = 400,                                     # Number of DMC blocks
    dependencies    = orbdeps+[(optJ3,'jastrow')],           # Dependence (1B and 2B Jastrows)
)

run_project()
```

Will take some time to run on VM.

# Setting up a DMC Calculation

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc2.py

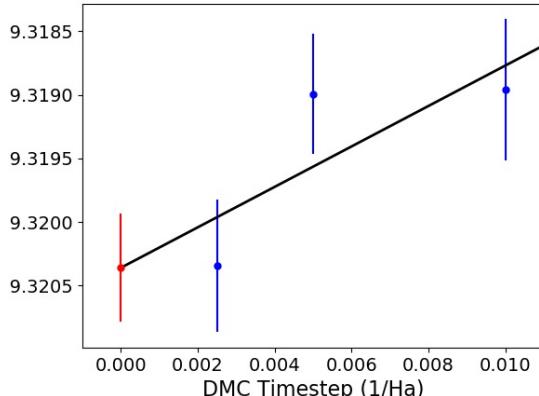
## Extrapolate Time Step

We evaluate autocorrelation times for each time step

```
qmc -q e *s00{1,2,3,4}*.scalar.dat --sac  
  
dmc series 1 LocalEnergy      = -29.319165 +/- 0.000422 2.7  
dmc series 2 LocalEnergy      = -29.319135 +/- 0.000471 4.3  
dmc series 3 LocalEnergy      = -29.320283 +/- 0.000544 4.7  
dmc series 4 LocalEnergy      = -29.319607 +/- 0.000396 3.3
```

```
qmc-fit ts -e 20 -b '2 4 4' -t '0.01 0.005 0.0025' *s00{1,2,3}*scalar*
```

```
fit function : linear  
fitted formula: (-29.32036 +/- 0.00043) + (0.159 +/- 0.066)*t  
intercept   : -29.32036 +/- 0.00043 Ha
```

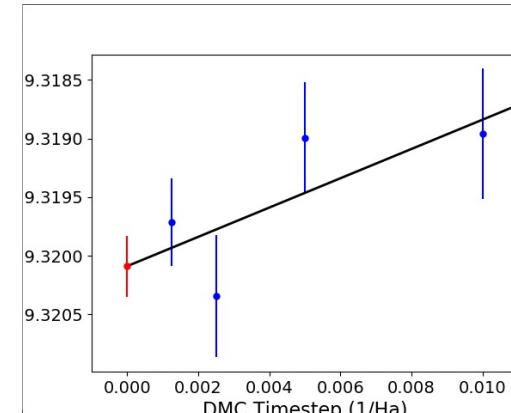


The extrapolation is performed by resampling the data at each timestep (jackknife resampling), making a fit to each resampled mean, and then observing the distribution of extrapolated zero time intercepts. We have reblocked the data by a factor of 8 (averaged 8 blocks into single new blocks) to remove autocorrelation.

Time steps=0.0025 Ha<sup>-1</sup>  
Steps =80

```
qmc-fit ts -e 20 -b '2 4 4 3' -t '0.01 0.005 0.0025 0.00125' *s00{1,2,3,4}*scalar*
```

```
fit function : linear  
fitted formula: (-29.32009 +/- 0.00026) + (0.125 +/- 0.050)*t  
intercept   : -29.32009 +/- 0.00026 Ha
```



# Setting up a DMC Calculation

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc3.py

Goal: Population bias

Change of file



```
#### DMC BLOCKS FOR POPULATION BIAS ANALYSIS
# run DMC with cusp Correction and 1,2 and 3 Body Jastrow function
pop=64
while pop<2049:
    qmc = generate_qmcpack(
        identifier      = 'dmc_pop'+str(pop),
        seed            = 42,
        path            = 'Be2/'+'y+'+'/'+x+'/'dmc_pop',           # directory to run in
        job             = job(cores=cores),                         # Submit with the number of cores available
        system          = system,
        jastrows        = [],
        qmc             = 'dmc',                                     # dmc run
        vmc_samples     = pop,                                      # Number of Samples (selected from a VMC step)
        warmupsteps    = 0,                                         # Number of Equilibration steps
        vmc_blocks     = 10,                                        # Number of VMC blocks (To generate the DMC samples)
        vmc_steps       = 1,                                         # Number of VMC steps (To generate DMC samples)
        vmc_timestep   = 0.3,                                       # VMC Timestep (To Generate DMC samples)
        timestep        = 0.01,                                       # DMC timestep
        steps           = 20480//pop,                                # number of step*population maintained constant to maintain error bar
        blocks          = 10,                                         # Number of DMC blocks
        dependencies    = orbdeps+[(optJ3,'jastrow')],           # Dependence (1B and 2B Jastrows)
    )
    pop=pop*2

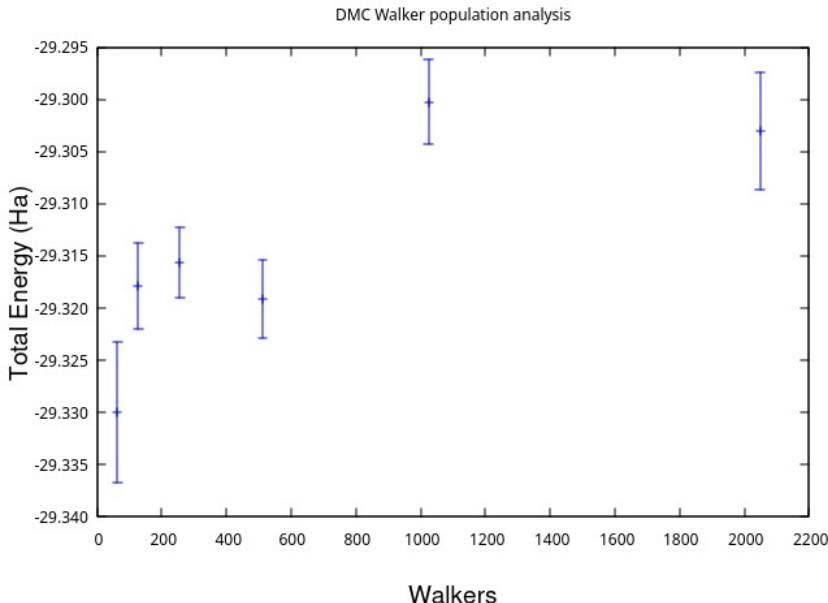
run_project()
```

We avoid equilibration time in DMC to show the population bias and loop through multiple walker configurations

# Setting up a DMC Calculation

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc3.py

## Goal: Population bias



- We will use a simple script to extract the dmc energies from each run with increased population and plot them using gnuplot
- Script name is Extract\_data\_qmc\_Pop.sh (will parse the output and put the energies in a correct format)
- Type:  
`sh Extract_data_qmc_Pop.sh`
- You can look at the script to generate the plots by typing:  
`cat Be2_dmc_pop_bias.gp`  
`gnuplot Be2_dmc_pop_bias.gp`

Using a population of 1024 seem to be enough to converge the energy  
Using 512 walkers would introduce a ~20mHa bias in the energy

# Setting up a DMC Calculation

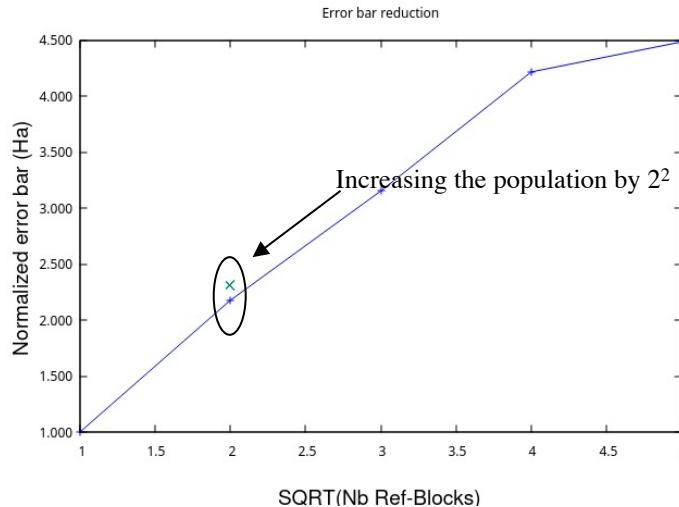
File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc4.py

Goal: Reducing the error bars

Change of file



To reduce the error bar by a factor N, multiply the population or the number of blocks by  $N^2$



Small computer -> reduce population and run longer

Large computer -> increase population and run shorter

```
#### DMC BLOCKS FOR error bar analysis - through simulation length (Block)
# run DMC with cusp Correction and 1,2 and 3 Body Jastrow function
init_blocks=100
for i in range(1,6):
    myblocks=init_blocks*i*i
    qmc = generate_qmcpack(
        identifier      = 'dmc_error'+str(i),
        seed            = 42,
        path            = 'Be2/ty+'/'tx+'/'dmc_error-bar',
        job             = job(cores=cores),
        system          = system,
        jastrows        = [],
        qmc             = 'dmc',
        vmc_samples     = 1024,
        warmupsteps    = 50,
        vmc_blocks     = 100,
        vmc_steps       = 10,
        vmc_timestep   = 0.1,
        timestep        = 0.00250,
        steps           = 80,
        blocks          = myblocks,
        dependencies    = orbdeps+[('optJ3','jastrow')],
    )
```

Error-bar: standard dev. of  $\sigma_e = \frac{\sigma_x}{\sqrt{M}}$  distribution of means

- Type:  
sh Extract\_data\_qmc\_Error-bar.sh

- You can look at the script to generate the plots by typing:

```
cat Be2_dmc_pop_bias.gp
gnuplot Be2_dmc_pop_bias.gp
```

# Running a DMC Calculation

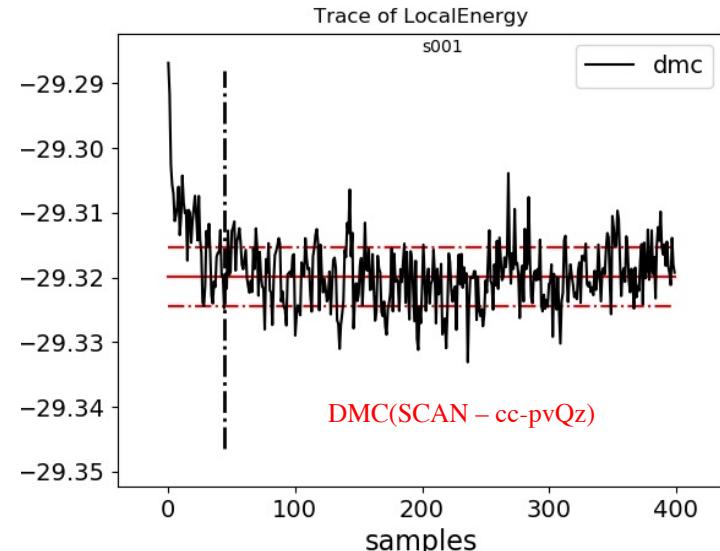
```
XC=["LDA","PBE","PBE0","SCAN"]
MyBasis=["cc-pvdz","cc-pvtz","cc-pvqz"]
for x in XC:
    for y in MyBasis:
        # perform DFT
```

```
# run DMC with cusp Correction and 1,2 and 3 Body Jastrow function
qmc = generate_qmcpack(
    identifier      = 'dmc',
    seed            = 42,
    path            = 'Be2/'+y+'/'+x+'/dmc',
    job             = job(cores=cores),
    system          = system,
    jastrows        = [],
    qmc             = 'dmc',
    vmc_samples     = 4096,
    warmupsteps    = 100,
    vmc_blocks     = 100,
    vmc_steps       = 10,
    vmc_timestep   = 0.3,
    timestep        = 0.00250,
    steps           = 80,
    blocks          = 400,
    dependencies   = orbdeps+[('optJ3','jastrow')],
```

```
qmca -q ev runs/Be2/cc-pvqz/SCAN/dmc/*.s001.scalar.dat
                                         LocalEnergy          Variance      ratio
runs/Be2/cc-pvqz/SCAN/dmc  series 1  -29.319851 +/- 0.000505  0.215959 +/- 0.000217  0.0074
```

File located in:  
week4\_molecules/01\_qmc\_SD/nx\_Be2\_qmc5.py

Change of file

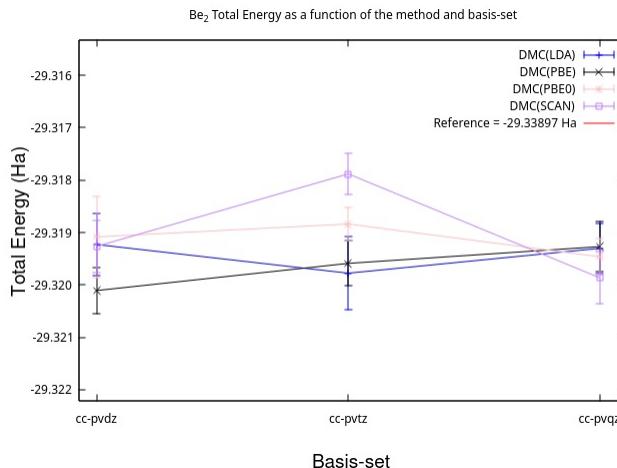


Small variance

Error of 19mHa compared to experimental

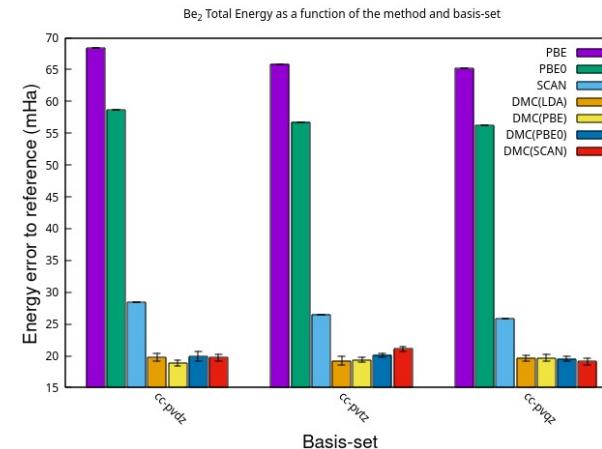
# Plotting/analyzing your data (gnuplot)

- We will use a simple script to extract the energies from the output files and plot them using gnuplot
- Script name is Extract\_data\_qmc.sh (will parse the output and put the energies in a correct format)
- Type:  
`sh Extract_data_qmc.sh`



All DMC energies within each other's error bar  
(almost no dependence to basis-set or starting point)

- You can generate the plot by typing:  
`gnuplot Be2_TotEnergy_QMC.gp`  
`gnuplot Be2_EnergyDiff_QMC.gp`



Systematic improvement of DMC over DFT

Still, ~19mHa off experimental, How to improve the result?

QMC Workshop 2021

Questions ?

# Setting up an ECP DMC Calculation

## Application to Be<sub>2</sub> molecule

The use of ECPs reduces the number of electrons to be evaluated explicitly

Same steps as all electron calculations but we need to redo study for:

How to set up time step, number of blocks, number of steps and number of walkers

- Extrapolate the time-step
- Control population bias
- Reducing error bars

Change in Nexus

```
settings(  
    pseudo_dir = 'pseudopotentials',  
    results     = '',  
    sleep       = 3,  
    machine     = 'ws'+str(cores),  
)  
  
ppset(  
    label      = 'ccecp',  
    qmcpack   = ['Be.ccECP.xml'],  
)  
  
system = generate_physical_system(  
    structure = 'Be2.xyz',  
    Be=2,  
)
```

Worth spending a few minutes on pointing out the difference between cc-ECP and all electron calculations

# Setting up a DMC Calculation

Effect of Jastrows on VMC and DMC energy and variance

All electrons VMC

	LocalEnergy	Variance	ratio
runs/Be2/cc-pvtz/SCAN/vmc_NoJ/vmc	series 0 -29.122630 +/- 0.008140	6.293397 +/- 0.650270	0.2161
runs/Be2/cc-pvtz/SCAN/vmc_2J/vmc	series 0 -29.264403 +/- 0.002655	0.497976 +/- 0.002821	0.0170
runs/Be2/cc-pvtz/SCAN/vmc_3J/vmc	series 0 -29.281952 +/- 0.002211	0.262596 +/- 0.002101	0.0090

All electrons DMC

	LocalEnergy	Variance	ratio
runs/Be2/cc-pvtz/SCAN/dmc_NoJ/dmc	series 1 -29.322531 +/- 0.002778	5.032637 +/- 0.029377	0.1716
runs/Be2/cc-pvtz/SCAN/dmc_2J/dmc	series 1 -29.319212 +/- 0.000941	0.492822 +/- 0.001771	0.0168
runs/Be2/cc-pvtz/SCAN/dmc_3J/dmc	series 1 -29.318362 +/- 0.000797	0.258312 +/- 0.001667	0.0088

$$\hat{H} = -\frac{1}{2}\nabla^2 + V_{ee} + V_{eI} + V_{II} \\ + V_{FN} + (\Psi_T^{-1}V_{eI}\Psi_T - V_{eI})$$

Fixed node  
constraint

Pseudopotential  
localization

File located in:

week4\_molecules/02\_qmc\_ccECP/nx\_Be2\_qmc0.py

Change of file



ccECP VMC

	LocalEnergy	Variance	ratio
runs/Be2/Calibration/vmc_NoJ/vmc	series 0 -1.906315 +/- 0.002627	0.141744 +/- 0.005848	0.0744
runs/Be2/Calibration/vmc_2J/vmc	series 0 -1.995480 +/- 0.001028	0.037666 +/- 0.001119	0.0189
runs/Be2/Calibration/vmc_3J/vmc	series 0 -1.999279 +/- 0.001005	0.042458 +/- 0.001270	0.0212

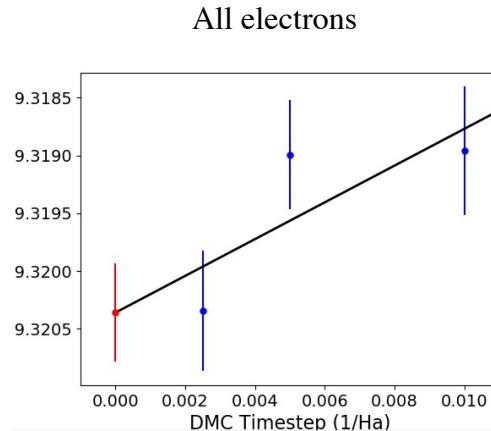
ccECP DMC

	LocalEnergy	Variance	ratio
runs/Be2/Calibration/dmc_NoJ/dmc	series 1 -2.000624 +/- 0.002397	0.085552 +/- 0.001534	0.0428
runs/Be2/Calibration/dmc_2J/dmc	series 1 -2.010426 +/- 0.000499	0.036481 +/- 0.000387	0.0181
runs/Be2/Calibration/dmc_3J/dmc	series 1 -2.012832 +/- 0.000589	0.043121 +/- 0.001206	0.0214

- The Jastrow will act on the local part of the pseudopotential (More details in lecture #6 of the Workshop) e.g DMC with and without Jastrows will not necessarily converge to the same energy.
- VMC and DMC calculations speed up by the use of Jastrows

# Setting up a DMC Calculation

Time step study



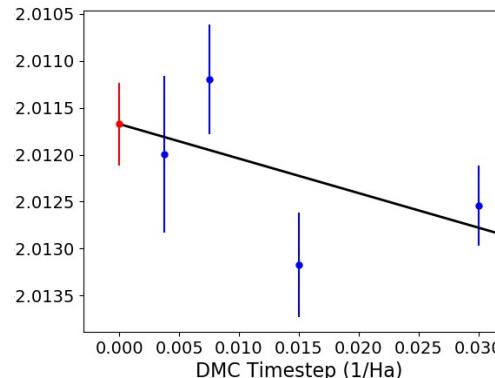
```
fit function : linear
fitted formula: (-29.32036 +/- 0.00043) + (0.159 +/- 0.066)*t
intercept   : -29.32036 +/- 0.00043 Ha
```

File located in:  
week4\_molecules/02\_qmc\_ccECP/nx\_Be2\_qmc1.py

Change of file



ccECP



```
fit function : linear
fitted formula: (-2.01167 +/- 0.00044) + (-0.037 +/- 0.021)*t
intercept   : -2.01167 +/- 0.00044 Ha
```

- When using ECPs, DMC is significantly less sensitive to the time step error (compared to all electrons calculations)
- The time step error is likely to benefit from cancellation of error when calculating energy differences

# Atomization Energy

All electrons

Experimental: -4.26mHa

ccECP

Single Atom reference:

DFT-SCAN = -14.649312 Ha

DMC-SCAN = -14.657269 +/- 0.000815 Ha

Be2 dimer total energy:

DFT-SCAN -29.3124577 Ha

DMC-SCAN = -29.32036 +/- 0.00043 Ha

Be2 atomization energy:

DFT-SCAN = -13.83 mHa

DMC-SCAN = - 5.82 +/- 0.9 mHa

Single Atom reference:

DFT-SCAN = -0.994018 Ha

DMC-SCAN = -1.00412 +/- 0.00042 Ha

Be2 dimer total energy:

DFT-SCAN -1.999310 Ha

DMC-SCAN = -2.01167 +/- 0.00044 Ha

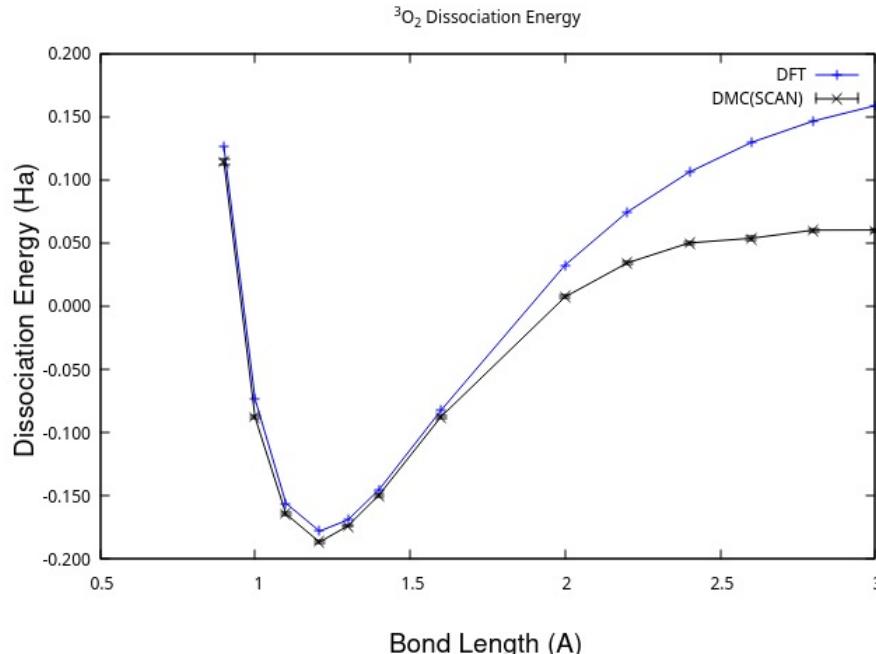
Be2 atomization energy:

DFT-SCAN = -11.27 mHa

DMC-SCAN = -3.43 +/- 0.65 mHa

# $^3\text{O}_2$ Dissociation curve

File located in:  
week4\_molecules/02\_qmc\_ccECP/nx\_O2\_qmc.py



Trial WF:DFT-SCAN / ccECP cc-pvtz  
Spin State: Triplet

$$E_d^{\text{exp}} = 0.18816 \text{ Ha}$$

$$E_d^{\text{DFT}} = 0.1805 \text{ Ha}$$

$$E_d^{\text{DMC}} = 0.1864 (5) \text{ Ha}$$

- DMC dissociation energy of  $\text{O}_2$  in very good agreement with experimental value
- Using ECPs (less electrons) and Nexus, it is possible to drive a full dissociation curve automatically  
(but verification is always necessary!)

Questions ?

# Examples of applications

# Systematic reduction of the fixed node error

The fixed node error can be systematically reduced for both solids and molecules by increasing the complexity of the trial wavefunction.

$$\Psi_T(R) = J(R)\Psi_{AS}(R) = e^{J_1 + J_2 + \dots} \boxed{\sum_k c_k \sum_q D_{k,q}^{\uparrow}(R^{\uparrow}) D_{k,q}^{\downarrow}(R^{\downarrow})} \longrightarrow \text{Multideterminant expansion}$$

## Correlation (Jastrow)

$$J_1 = \sum_i^N \sum_l^{N_{ions}} u_1(|r_i - r_l|)$$

$$J_2 = \sum_{i \neq j}^N u_2(|r_i - r_j|)$$

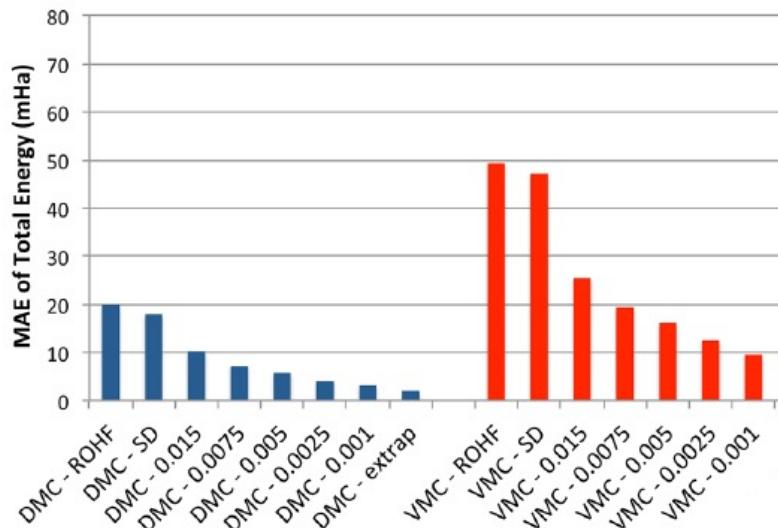
- Using advanced trial wavefunctions like CASSCF, or selected Configuration Interaction (sCI)
- Orbital optimization
- Using backflow
- Combination of all the above
- (..)

# Using multideterminants

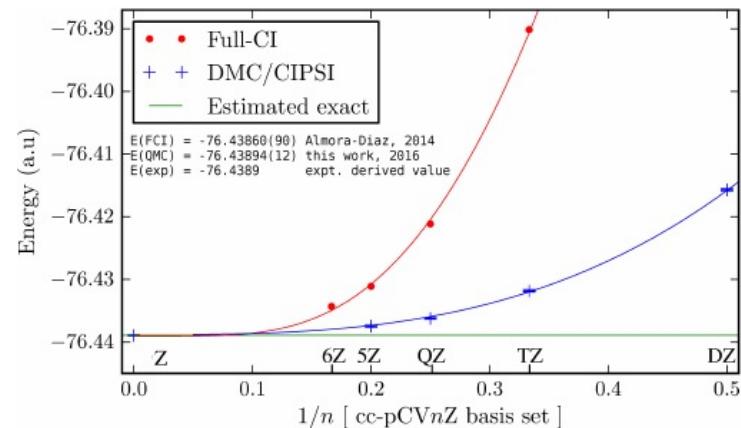
- sCI evolves in Hilbert space
- DMC evolves in real space

Strategy: Increase the complexity of the trial wavefunction using a multideterminant expansion

G1 test set



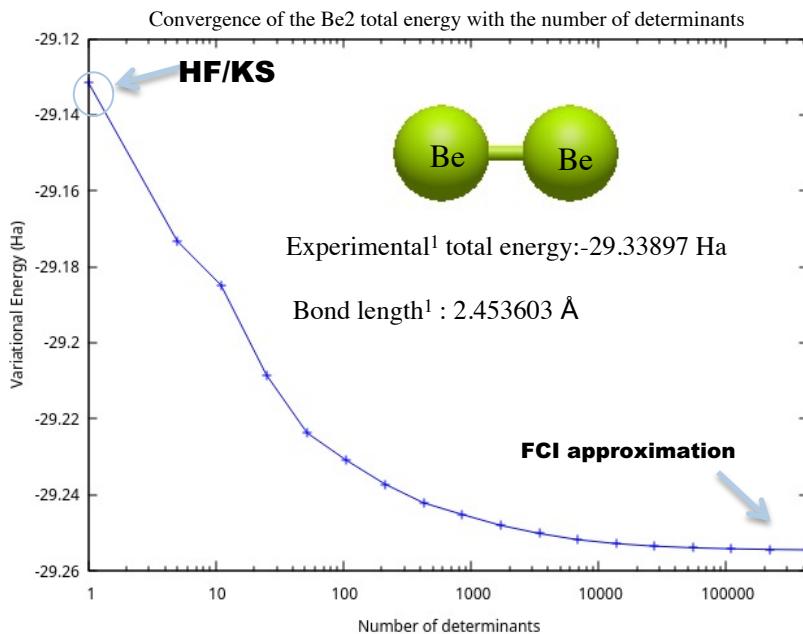
H<sub>2</sub>O



Morales et al. J. Chem. Theory Comput. 8, 2181–2188, (2012)

Caffarel et. al , J. Chem. Phys. 144, 151103 (2016); <http://dx.doi.org/10.1063/1.4947093>

# Using sCI (CIPSI flavor)



- Convergence at 0.1 mHa with 0.45M determinants
- Selected CI (CIPSI) systematically reduces the energy
- Selecting all determinants leads rigorously to the FCI energy
- Refer to QMCPACK manual For details about how to generate the trial wavefunction

1. Define a reference wave function:

$$|\Psi\rangle = \sum_{i \in D} c_i |i\rangle$$

$$E_{var} = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

2. Generate external Determinants

All single and double excitations

3. Second order perturbative contribution of each determinant  $|\alpha\rangle$

$$\Delta E = \frac{\langle \Psi | \hat{H} | \alpha \rangle \langle \alpha | \hat{H} | \Psi \rangle}{E_{var} - \langle \alpha | \hat{H} | \alpha \rangle}$$

4. Select  $|\alpha\rangle$ 's with largest  $\Delta E_\alpha$  and add them to Determinant space (D)

5. Diagonalize  $\hat{H}$  in D then update  $|\Psi\rangle$  and  $E_{var}$

6. Iterate until reaching convergence.



Quantum Package 2

# Running a DMC Calculation

File located in:

week4\_molecules/03\_qmc\_MD/nx\_Be2\_ae\_qmc\_workflow.py

Important:

- QP2 is not installed in the VM! It is recommended to follow the tutorial on QP2 from the 2019 QMCPACK workshop  
[https://github.com/QMCPACK/qmcpack\\_workshop\\_2019](https://github.com/QMCPACK/qmcpack_workshop_2019)
- The multideterminant trial wavefunction generated with QP2 is provided in *week4\_molecules/02\_qmc\_MD/Refs/Be2/QP/QP.h5*
- Calculations are started from the existing H5 File
- We follow same steps for time step extrapolation as SD runs

Change of file



```
# convert orbitals to QMCPACK format
c4q = generate_convert4qmc(
    identifier = 'c4q',
    path       = 'Be2/QP',
    job        = job(cores=1),
    add_cusp   = True,
    multidet   = 'QP.h5',
    orbitals   = 'QP.h5',
)
```

## sCI workflow

We generate initial orbitals from HF in a cc-pvtz basis-set

We generate a multideterminant wavefunction using CIPSI and converge the energy to the FCI limit (0.5M determinants)

We truncate the wavefunction to keep only coefficients larger than  $10^{-5}$  (395 determinants)

# Running a DMC Calculation

File located in:  
week4\_molecules/03\_qmc\_MD/nx\_Be2\_ae\_qmc\_workflow.py

CIPSI Variational Energy **-29.2331 Ha**

Single Determinant DMC

```
qmca -q ev runs/Be2/cc-pvqz/SCAN/dmc/*.s001.scalar.dat
                                         LocalEnergy
runs/Be2/cc-pvqz/SCAN/dmc/dmc  series 1  -29.319851 +/- 0.000505
                                         Variance   0.215959 +/- 0.000217
                                         ratio      0.0074
```

Multideterminant DMC

```
qmca -q ev runs/Be2/dmc_msd/*.scalar.dat
                                         LocalEnergy
VMC  series 0  -29.333057 +/- 0.000908  Variance   0.104126 +/- 0.001649  ratio      0.0035
DMC  series 3  -29.337331 +/- 0.000403  Variance   0.107536 +/- 0.001219  ratio      0.0037
```

Experimental<sup>1</sup> total energy:-29.33897 Ha

- Small MSD expansion corrects the total energy to 1.5 +/- 0.5 mHa from experimental energy
- VMC energy with MSD wavefunction significantly more accurate than SD-DMC (Smaller variance)

[1] V. V. Meshokov, A. V. Stolyarov, M. C. Heaven, C. Haugen, and R. J. Leroy, J. Chem. Phys. 140, 064315 (2014).

# Noncovalent Interactions



Journal of Chemical Theory and Computation

Letter

pubs.acs.org/JCTC

## Quantum Monte Carlo Methods Describe Noncovalent Interactions with Subchemical Accuracy

Matúš Dubecký,<sup>\*,†</sup> Petr Jurečka,<sup>†</sup> René Derian,<sup>‡</sup> Pavel Hobza,<sup>§,†</sup> Michal Otyepka,<sup>†</sup> and Lubos Mitas<sup>||</sup>

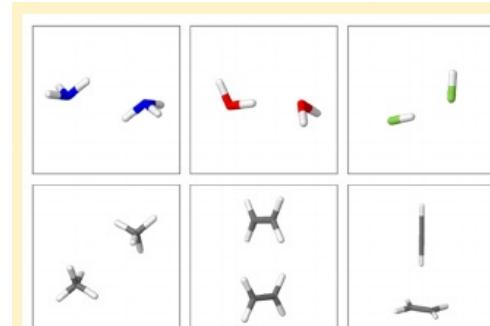
Journal of Chemical Theory and Computation

Letter

Table 1. Calculated FN-DMC Interaction Energies (kcal/mol), Based on B3LYP/aug-TZV Guiding Functions, Compared to the Previous FN-DMC<sup>26</sup> and Reference CCSD(T)/CBS Data<sup>8,40</sup> Together with the Corresponding Differences  $\Delta^a$

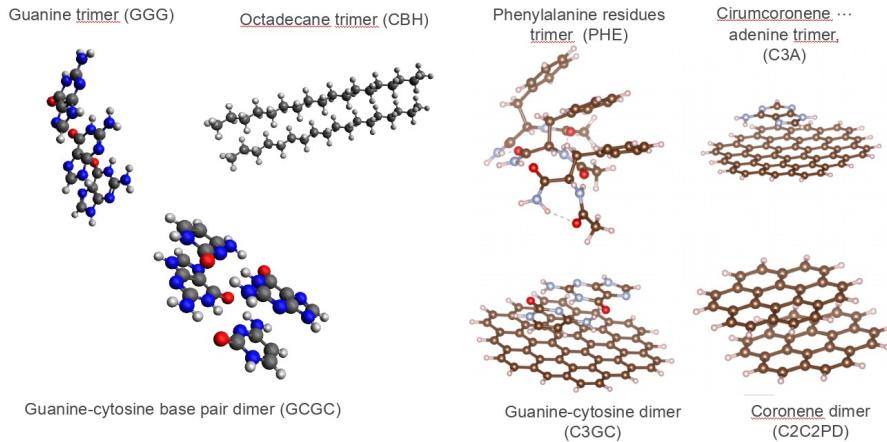
complex	reference	FN-DMC <sup>b</sup>	$\Delta^b$	FN-DMC <sup>c</sup>	$\Delta^c$
ammonia dimer	-3.15	-3.19 ± 0.09	0.04	-3.10 ± 0.06	-0.05
water dimer	-5.07	-5.34 ± 0.09	0.27	-5.15 ± 0.08	0.08
hydrogen fluoride dimer	-4.58			-4.68 ± 0.10	0.10
methane dimer	-0.53	-0.48 ± 0.08	-0.05	-0.44 ± 0.05	-0.09
ethene dimer	-1.48	-1.38 ± 0.13	-0.10	-1.47 ± 0.09	-0.01
ethene/ethyne	-1.50	-1.22 ± 0.12	-0.28	-1.56 ± 0.08	0.06
benzene/water	-3.29	-3.69 ± 0.24	0.40	-3.53 ± 0.13	0.24
benzene/methane	-1.45	-0.63 ± 0.21	-0.87	-1.30 ± 0.13	-0.15
benzene dimer T	-2.71	-3.77 ± 0.39	1.06	-2.88 ± 0.16	0.17

<sup>a</sup>The reference values for the test set (the top panel) are obtained by extrapolation of CCSD(T) (aug-TZV, aug-QZV) to the CBS limit, while for larger complexes (the bottom panel), the estimates combine contributions from MP2/CBS (aug-TZV, aug-QZV) and CCSD(T)/CBS (aug-DZV, aug-TZV). <sup>b</sup>Körth et al.<sup>26</sup> <sup>c</sup>This work.



QMC  
vs.  
CCSD(T)  
agree to  
0.1 kcal/mol

# L7 Benchmark (van der Waals dominated molecules)



- DMC in all electrons (up to 5000 MO) at CBS limit.
- CCSD(T) with ecp and approximated CBS.
- DMC in very good agreement with CCSD(T)
- DMC all electrons and ECP from different basissets, functionals and codes reach exact same binding energies

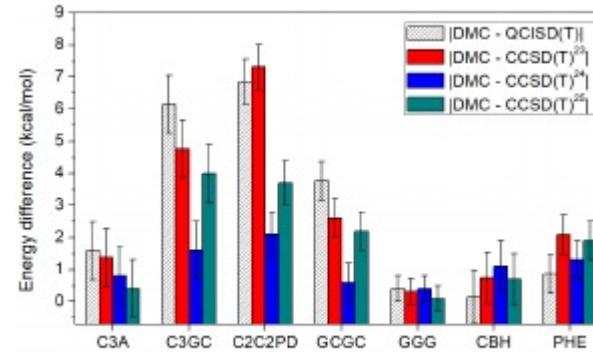


FIG. 1. Banding energy differences in kcal/mol relative to QCISD(T) and CCSD(T).

	<b>CASINO<sup>1</sup></b>	<b>QMCPACK<sup>2</sup></b>	<b>Δ</b>
<b>GGG</b>	<b>-1.5 +/- 0.3</b>	<b>-2.0 +/- 0.4</b>	<b>0.5 +/- 0.5</b>
<b>CBH</b>	<b>-11.4 +/- 0.4</b>	<b>-10.9 +/- 0.8</b>	<b>0.5 +/- 0.9</b>
<b>PHE</b>	<b>-26.5 +/- 0.7</b>	<b>-24.9 +/- 0.6</b>	<b>1.6 +/- 0.9</b>
<b>C3A</b>	<b>-15.0 +/- 0.5</b>	<b>-16.6 +/- 0.9</b>	<b>1.6 +/- 1.0</b>
<b>GCGC</b>	<b>-12.3 +/- 0.3</b>	<b>-10.6 +/- 0.6</b>	<b>1.7 +/- 0.7</b>
<b>C3GC</b>	<b>-24.2 +/- 0.7</b>	<b>-25.1 +/- 0.9</b>	<b>0.9 +/- 1.1</b>
<b>C2C2PD</b>	<b>-18.1 +/- 0.4</b>	<b>-17.5 +/- 0.7</b>	<b>0.6 +/- 0.8</b>

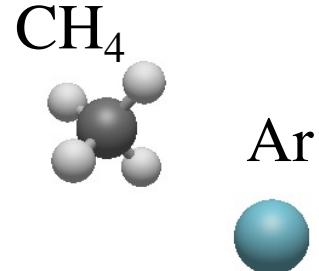
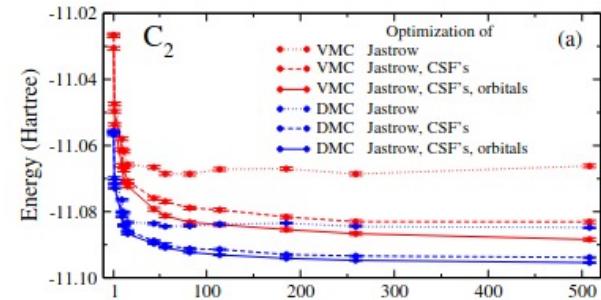
[1] "Interactions between Large Molecules: Puzzle for Reference Quantum-Mechanical Methods" Y. S. Al-Hamdani, P. R. Nagy, D. Barton, M. Kallay, J.G. Brandenburg, A. Tkatchenko, arXiv:2009.08927 (2020)

[2] "Quantum Monte Carlo benchmarking of large noncovalent complexes in the L7 benchmark set" A. Benali, H. Shin, O. Heinonen, J. Chem. Phys.

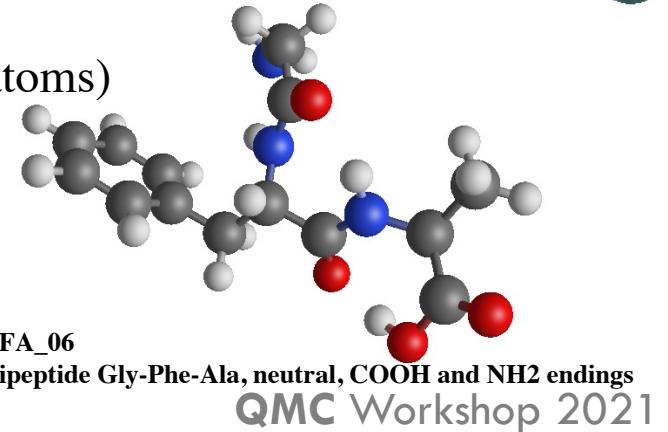
153, 194113 (2020)

# Suggestions of applications

- On Workstation (~10 to 20 electrons):  
Dissociation curves of O<sub>2</sub>, [1]C<sub>2</sub>, F<sub>2</sub>, LiF LiH etc..
- On larger cluster (~350 electrons )  
van der Waals systems (A24 Benchmark) (binding of ~7kcal/mol)



Small peptides containing aromatic side chain (~40 atoms)



[1] C. J. Umrigar, J. Toulouse, C. Filippi, S. Sorella, and R. G. Hennig Phys. Rev. Lett. 98, 110201 (2007)

[2] <http://www.begdb.org/>

# Summary

This tutorial covered how to run VMC and DMC with wavefunctions defined in gaussian basis sets

Basis set convergence, time-step, and other QMC parameters should be checked

DMC can be a very powerful tool to study consistently molecular systems

DMC dependence to the basis set is minimal when compared to quantum chemistry methods but requires careful checks and adequate extrapolations of local approximations.

It is possible to systematically improve the quality of the trial wavefunction and extrapolate out the fixed-node error with multideterminant wavefunctions, but at a higher computational cost.

Nexus can significantly reduce the complexity of a QMC calculation and can be used to manage large scale campaigns

Next week (#week 5) will be focused on QMC for solid state by Dr. Joshua Townsend (SNL)

Questions ?