

QMC & QMCPACK

Summer School 2025

Tuesdays and Thursdays, 1 through 24 July

Session 3 Statistical Analysis and Nexus Workflows: Jaron T. Krogel, krogeljt@ornl.gov

https://github.com/QMCPACK/qmc_summer_school_2025

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.

Outline

Statistical analysis of QMC time series data

Overview of Nexus QMC workflow automation system

Overview of session 3 lab material

VMC, Jastrow Optimization, and DMC with QMCPACK

Next week: Molecular and Solid-State QMC calculations

Statistical Analysis of QMC Time Series

Gaussian Distributions: Mean and Variance

Understanding Gaussian/Normal distributions essential to analysis of all QMC data.

Gaussian Distribution Form

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

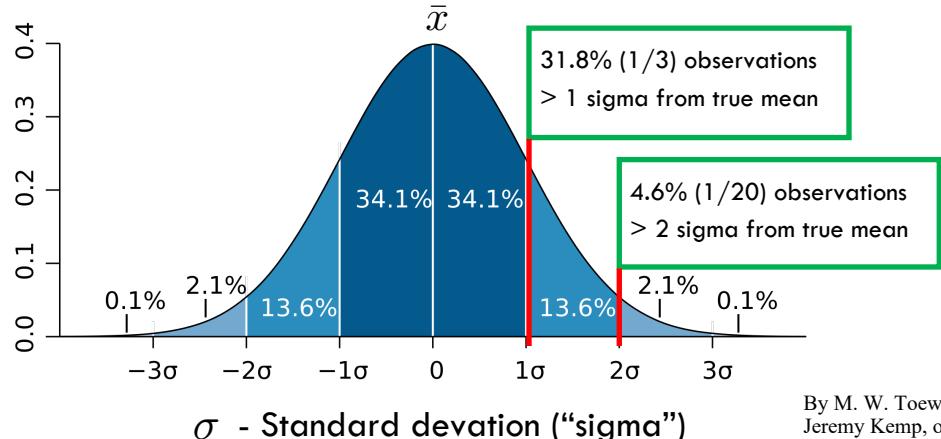
Mean/Expected Value

$$\begin{aligned}\langle x \rangle &= \int dx P(x)x \\ &= \bar{x}\end{aligned}$$

Variance (Distribution Width)

$$\begin{aligned}\sigma^2(x) &= \langle x^2 \rangle - \langle x \rangle^2 \\ &= \int dx P(x)x^2 - \left(\int dx P(x)x \right)^2 \\ &= \sigma^2\end{aligned}$$

Percentiles: Statistical Significance



Estimated Mean/Variance

$$\langle x \rangle \approx \frac{1}{M} \sum_{i=1}^M x_i$$

$$\sigma^2(x) \approx \frac{1}{M-1} \left[\sum_{i=1}^M x_i^2 - \frac{1}{M} \left(\sum_{i=1}^M x_i \right)^2 \right]$$

M - Number of (independent)
statistical samples

By M. W. Toews - Own work, based (in concept) on figure by Jeremy Kemp, on 2005-02-09, CC BY 2.5,
<https://commons.wikimedia.org/w/index.php?curid=1903871>

The Central Limit Theorem and the “Error Bar”

Central limit theorem (CLT): distribution of mean value estimates is Gaussian

Gaussian Random Variable (GRV)

$$\eta \sim \mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

$$\langle \eta \rangle = 0 \quad \sigma^2(\eta) = 1$$

Sums of Gaussian Random Variables

$$\sum_{i=1}^M \eta_i = \sqrt{M} \eta$$

“Arbitrary” Random Variables

$$x_i \sim P(x), \quad \langle x_i \rangle = \bar{x}, \quad \sigma^2(x_i) = \sigma_x^2$$

CLT: Sums of Arbitrary Random Variables

$$\sum_{i=1}^M x_i = M\bar{x} + \sqrt{M}\sigma_x\eta$$

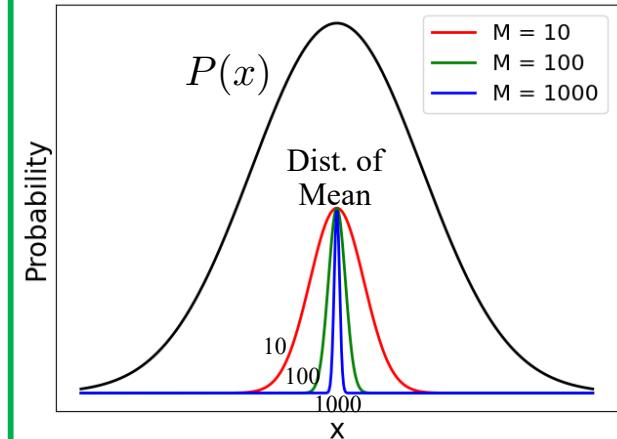
CLT Applied to Mean

$$\begin{aligned} \langle x \rangle &= \int dx P(x)x \\ &\approx \frac{1}{M} \sum_{i=1}^M x_i \\ &= \frac{1}{M} \left(M\bar{x} + \sqrt{M}\sigma_x\eta \right) \\ &= \bar{x} + \frac{\sigma_x}{\sqrt{M}}\eta \end{aligned}$$

Error bar: standard dev. of distribution of means

$$\sigma_e = \frac{\sigma_x}{\sqrt{M}}$$

(Independent samples!)



Variational and Zero Variance Principles

QMC energy distributions obey variational and zero variance principles.

Application: Judge quality of different candidate solutions.

QMC Energy Distributions

$$\langle E \rangle = \int dE P(E) E = \int dR P(R) E(R)$$

Electron Positions (3N)

VMC Energy

$$E_{VMC} = \frac{\langle \Psi_T | \hat{H} | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle}$$
$$= \frac{\int dR |\Psi_T|^2 \Psi_T^{-1} \hat{H} \Psi_T}{\int dR |\Psi_T|^2}$$
$$= \int dR \frac{f_T(R)}{\int dR f_T(R)} E_L(R)$$

$$f_T(R) = |\Psi_T|^2$$

$$E_L(R) = \Psi_T^{-1} \hat{H} \Psi_T$$

DMC Energy

$$E_{DMC} = \lim_{t \rightarrow \infty} \frac{\langle \Psi_T | e^{-t\hat{H}} \hat{H} | \Psi_T \rangle}{\langle \Psi_T | e^{-t\hat{H}} | \Psi_T \rangle}$$
$$= \frac{\langle \Psi_0 | \hat{H} | \Psi_T \rangle}{\langle \Psi_0 | \Psi_T \rangle}$$
$$= \int dR \frac{f_0(R)}{\int dR f_0(R)} E_L(R)$$

$$f_0(R) = \Psi_0 \Psi_T$$

Variational Principle

$$\hat{H}\Psi = E\Psi$$

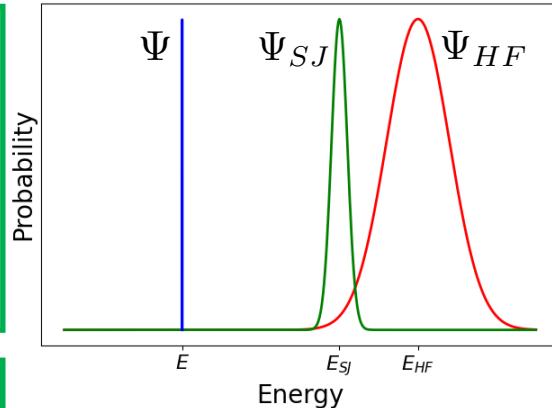
$$\langle E_{QMC} \rangle \geq E$$

$$\langle E_{QMC} \rangle \xrightarrow{\Psi_T \rightarrow \Psi} E$$

Zero Variance Principle

$$\sigma^2(E_{QMC}) \geq 0$$

$$\sigma^2(E_{QMC}) \xrightarrow{\Psi_T \rightarrow \Psi} 0$$

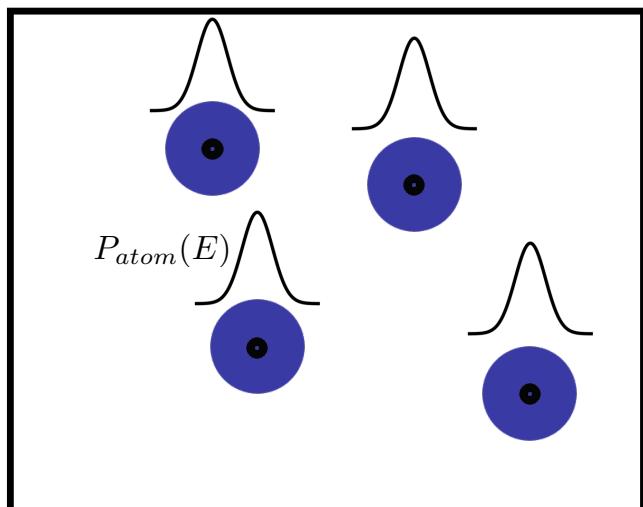


Better trial wavefunctions simultaneously result in better (lower) energies and smaller error bars!

Scaling of QMC Energy and Energy Variance

Total energy is extensive (scales linearly with system size). What is the system size scaling of the energy variance?

- Applications:** 1) Transferable wavefunction quality metric: variance/energy ratio
2) Predict sampling needs for larger systems from smaller ones



Single Atom Energy

$$E_i \sim P_{atom}(E)$$

$$\langle E_i \rangle = \bar{E}_{atom}$$

$$\sigma^2(E_i) = \sigma_{E_{atom}}^2$$

CLT: Total Energy

$$\begin{aligned} E_{tot} &= \sum_{i=1}^N E_i \\ &= N\bar{E}_{atom} + \sqrt{N}\sigma_{E_{atom}}\eta \end{aligned}$$

Energy & Variance Extensivity

$$\langle E_{tot} \rangle = N\bar{E}_{atom}$$

$$\sigma^2(E_{tot}) = N\sigma_{E_{atom}}^2$$

Variance/Energy Ratio

$$\frac{\sigma^2(E_{tot})}{\langle E_{tot} \rangle} = \frac{\sigma_{E_{atom}}^2}{\bar{E}_{atom}}$$

$$\leq 0.04 \text{ Ha}$$

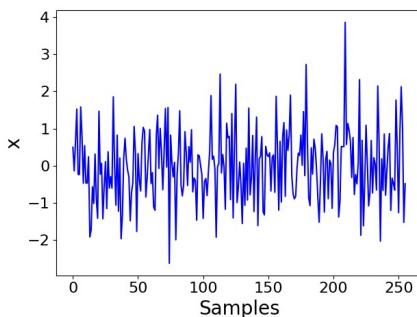
(Decent wavefunction)

Accounting for Temporal Autocorrelation

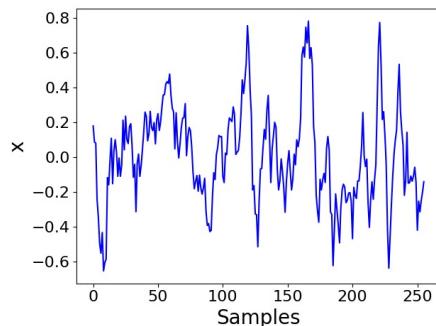
Autocorrelation: Adjacent MC data samples are not independent.

Error-bar is underestimated with fewer effective independent samples.

Uncorrelated Samples



Autocorrelated Samples

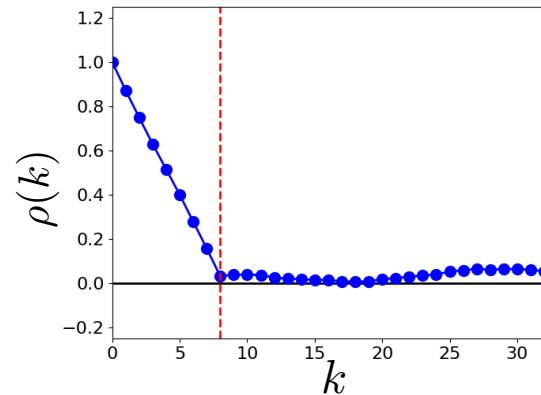


Corrected Error-bar

$$\sigma_e(x) = \frac{\sigma_x}{\sqrt{M_{eff}}} = \frac{\sigma_x}{\sqrt{M/\kappa}}$$

(Linear) Correlation

$$\rho(x, y) \approx \frac{\sum_{i=1}^M (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^M (x_i - \bar{x})^2 \sum_{i=1}^M (y_i - \bar{y})^2}}$$



Autocorrelation Function

$$\rho(k) \approx \frac{\frac{1}{M-k} \sum_{i=1}^{M-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\frac{1}{M} \sum_{i=1}^M (x_i - \bar{x})^2}$$

Autocorrelation Time

$$\kappa = \sum_{i=-\infty}^{\infty} \rho(k) \approx 1 + 2 \sum_{i=1}^K \rho(k)$$

Source of Temporal Autocorrelation in QMC Data

Drift-diffusion Green's function creates new electron positions from old ones.

Positions are correlated via narrow Gaussian distribution in imaginary time.

Distribution of new electron positions

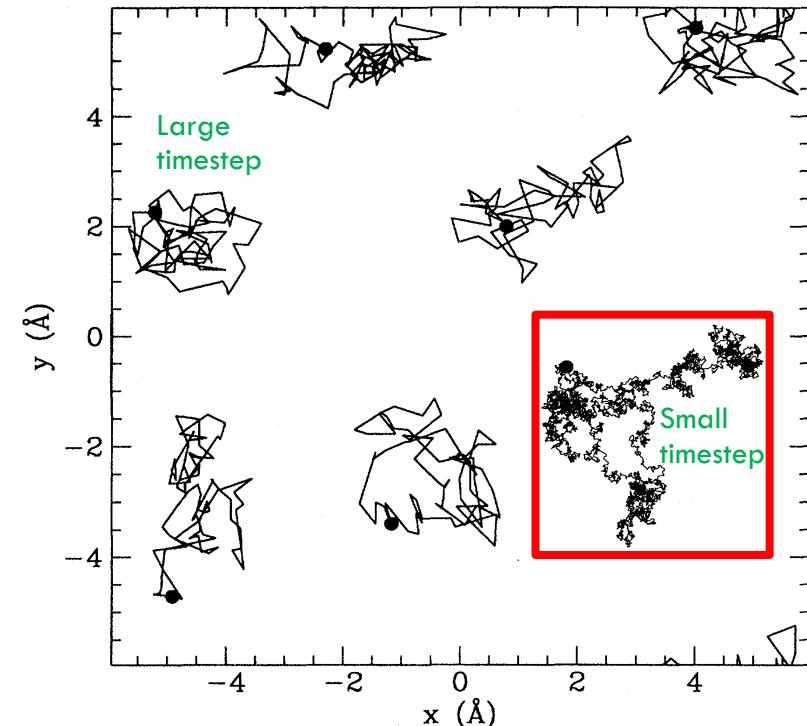
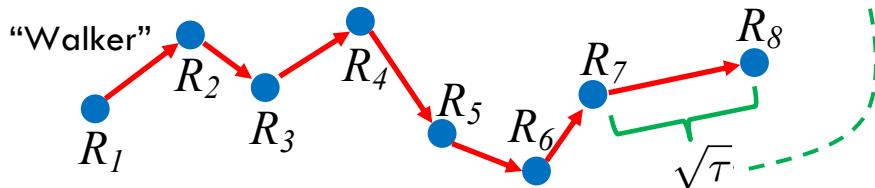
Distribution of old electron positions

$$\begin{aligned} f_T(R) &= \int dR' G_d(R, R', \tau) f_T(R') \\ &= \int dR' \exp\left(-\frac{(R - R' - \tau v_d(R'))^2}{2\tau}\right) f_T(R') \end{aligned}$$

Drift-diffusion Green's Function

Random Walk

$$\begin{array}{cccc} \text{New} & \text{Old} & \text{Drift} & \text{Diffusion} \\ R_{i+1} = R_i + \tau v_d(R_i) + \sqrt{\tau} \eta \end{array}$$

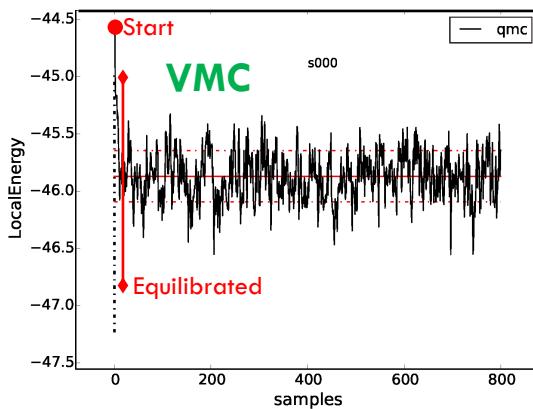


Statistical Equilibration (A.K.A. “Burn-in”)

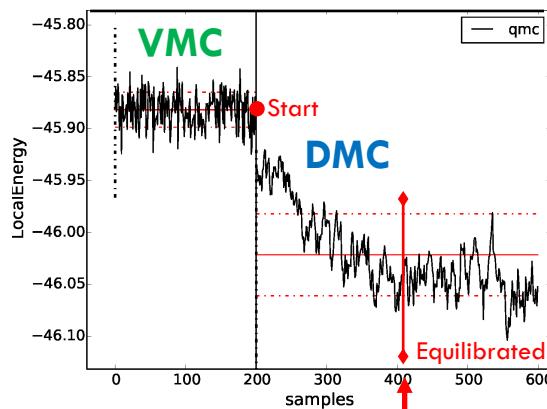
Random walk starts with arbitrary electron positions (low probability configuration).

Equilibration period: MC steps needed to begin sampling equilibrium distribution.

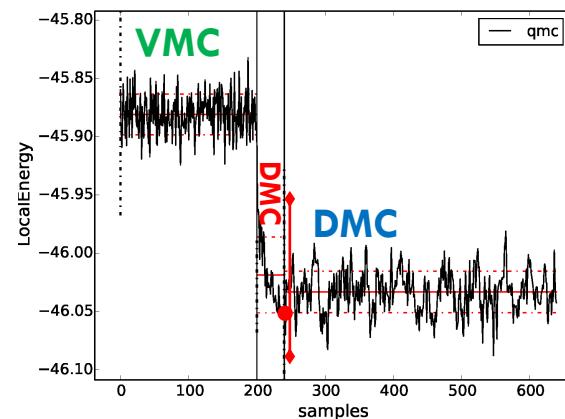
Variational Monte Carlo



DMC: Small Timestep



DMC: Large/Small Timestep



Estimation of mean,
autocorrelation, and
error bar starts here

Small Timestep: 0.002/Ha

Large Timestep: 0.02/Ha

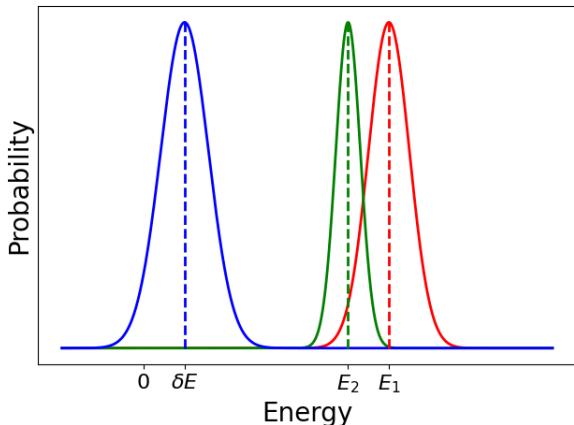
Estimating Uncertainty in Energy Differences

- Applications:** 1) Correctly assign statistical error bar of energy differences
2) Assess whether two values differ with statistical significance

Gaussian Distributed Energy Means

$$E_1 = \bar{E}_1 + \sigma_{1e}\eta_1$$

$$E_2 = \bar{E}_2 + \sigma_{2e}\eta_2$$



Gaussian Distributed Energy Difference

$$\begin{aligned} \delta E &= E_1 - E_2 \\ &= \bar{E}_1 - \bar{E}_2 + \sqrt{\sigma_{1e}^2 + \sigma_{2e}^2} \end{aligned}$$

(Distribution Broadens!)

Error Bar of Energy Difference

$$\sigma_e = \sqrt{\sigma_{1e}^2 + \sigma_{2e}^2}$$

Statistical Significance

$$|\bar{E}_1 - \bar{E}_2| > n \sqrt{\sigma_{1e}^2 + \sigma_{2e}^2}$$

Energies differ?

$n < 1$? Don't know/little evidence

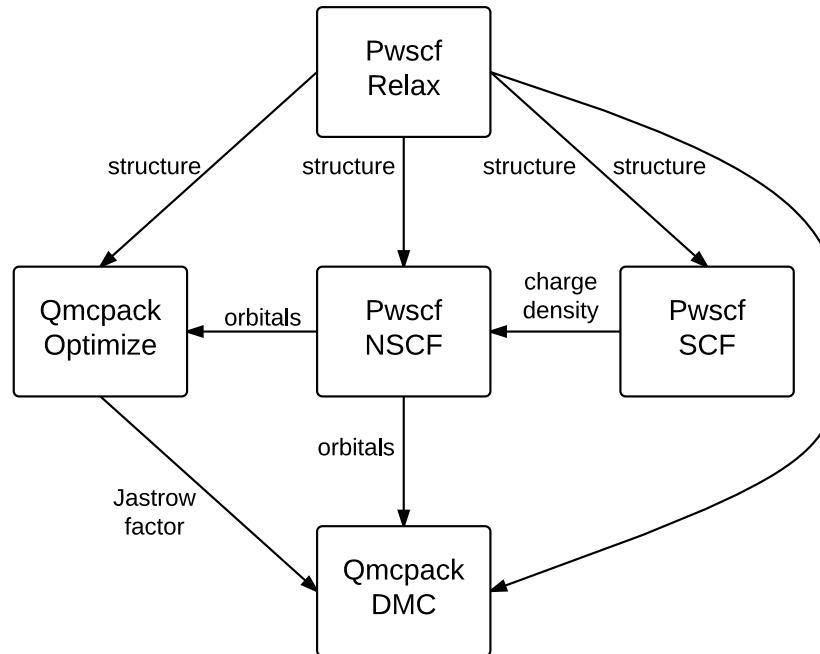
$n = 1$? Maybe (1/3 chance same)

$n = 2$? Probably so (1/20 chance same)

$n > 3$? Highly likely (<1/100 chance same)

Questions?

Nexus: QMC Workflow Automation



Nexus Workflow System Overview

Goals

- Make QMC calculations more accessible (less steep learning curve for new users)
- Automate time-consuming steps in QMC simulation process
- Hands-off: no need to compose/edit simulation input files by hand
- Enable large scale job submission on a variety of supercomputing platforms
- Make simulation workflows scriptable: e.g. scans over DFT functionals
- Support every type of calculation possible with QMCPACK and provide documented examples (still aspirational, but support is broad)

Nexus Core Features

Python interface for workflow composition

Meta input file for execution of all simulation codes

Brings all simulation programs together in single, coherent framework

Input file generators

Automatic input file creation from minimal user inputs

Monitoring

Automatic execution of simulation workflows and job monitoring

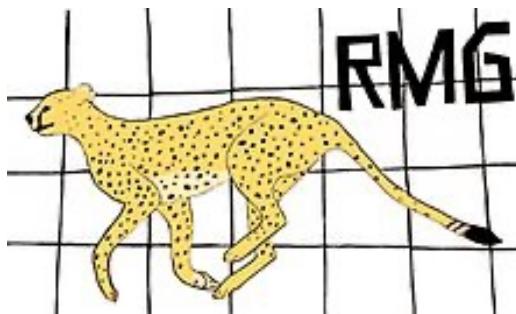
Analysis

Automatic extraction of simulation results, accessible within user script

Supported Electronic Structure Codes

QMCPACK

 **pyscf**

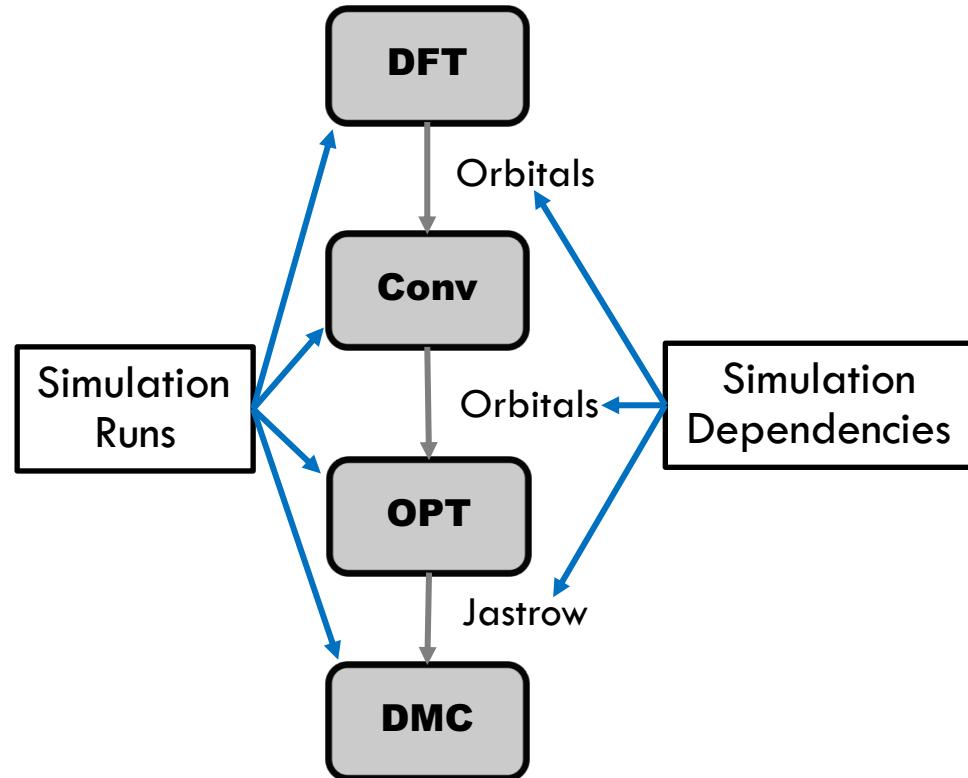

RMG



Nexus Workflow Execution/Monitoring

Example Molecular/ Solid State Workflow (DAG)

- DMC calculation of coronene/diamond
- DFT with PySCF/Quantum Espresso
- Orbital conversion with
convert4qmc/pw2qmcpack
- Jastrow optimization with QMCPACK
- Diffusion Monte Carlo with QMCPACK



Working with Nexus: Installation

Download

```
>mkdir /your/path/to  
>cd /your/path/to  
>git clone https://github.com/QMCPACK/qmcpack  
>cd /your/path/to/qmcpack/nexus
```

Installation

```
export PYTHONPATH=/your/path/to/qmcpack/nexus/lib:$PYTHONPATH  
export PATH=/your/path/to/qmcpack/nexus/bin:$PATH
```

Testing

```
>nxs-test
```

1/16 generic_logging.....	Passed	0.00 sec
2/16 generic_intrinsics.....	Passed	0.00 sec
3/16 generic_extensions.....	Passed	0.00 sec
4/16 nexus_imports.....	Passed	0.32 sec
5/16 settings.....	Passed	0.00 sec
6/16 machines.....	Passed	2.30 sec
7/16 structure.....	Passed	0.54 sec
8/16 pwscf_input.....	Passed	0.02 sec
9/16 example_gamess_H2O.....	Passed	1.60 sec
10/16 example_qmcpack_oxygen_dimer.....	Passed	0.73 sec
11/16 example_qmcpack_H2O.....	Passed	0.74 sec
12/16 example_qmcpack_LiH.....	Passed	0.84 sec
13/16 example_qmcpack_diamond.....	Passed	1.22 sec
14/16 example_pwscf_relax_Ge_T.....	Passed	0.71 sec
15/16 example_qmcpack_c20.....	Passed	0.74 sec
16/16 example_qmcpack_graphene.....	Passed	0.84 sec

100% tests passed, 0 tests failed out of 16

Total test time = 10.61 sec

Working with Nexus: User Script Composition

Invoke interpreter

Nexus imports

User settings

System definition

Workflow composition

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

from nexus import settings, job, run_project
from nexus import generate_physical_system
from nexus import generate_pwscf
from nexus import generate_pw2qmcpack
from nexus import generate_qmcpack, vmc

settings(
    pseudo_dir      = './pseudopotentials',
    status_only     = 0,
    generate_only   = 0,
    sleep           = 3,
    machine         = 'ws4'
)

dia16 = generate_physical_system(
    structure = './diamond.poscar',
    tiling    = (2,2,2),
    kgrid     = (1,1,1),
    kshift    = (0,0,0),
    C         = 4
)

scf = generate_pwscf(
    identifier   = 'scf',
    path         = 'diamond/scf',
    job          = job(cores=4, app='pw.x'),
    input_type   = 'generic',
    calculation  = 'scf',
    input_dft    = 'lda',
    ecutwfc     = 200,
    conv_thr     = 1e-8,
    nosym        = True,
    wf_collect   = True,
    system       = dia16,
    pseudos      = ['C.BFD.upf'],
)
```

```
conv = generate_pw2qmcpack(
    identifier   = 'conv',
    path         = 'diamond/scf',
    job          = job(cores=1,
                      app='pw2qmcpack.x'),
    write_psir   = False,
    dependencies = (scf, 'orbitals'),
)

qmc = generate_qmcpack(
    identifier   = 'vmc',
    path         = 'diamond/vmc',
    job          = job(cores=4,
                      threads=2,
                      app='qmcpack'),
    input_type   = 'basic',
    system       = dia16,
    pseudos      = ['C.BFD.xml'],
    jastrows     = [],
    calculations = [
        vmc(
            walkers      = 1,
            warmupsteps = 20,
            blocks      = 200,
            steps       = 10,
            substeps    = 2,
            timestep    = .4
        )
    ],
    dependencies = (conv, 'orbitals'),
)

run_project()
```

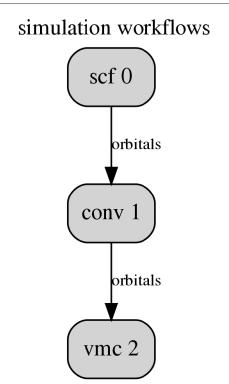
Workflow composition

Workflow execution

Working with Nexus: User Script Execution

Graph simulation workflows

```
>./diamond.py --graph_sims
```



Check initial run status

```
>./diamond.py --status_only
```

```
...
cascade status
  setup, sent_files, submitted, finished, ...
  000000 0 ----- scf      ./runs/diamond/scf
  000000 0 ----- conv     ./runs/diamond/scf
  000000 0 ----- vmc     ./runs/diamond/vmc
  setup, sent_files, submitted, finished, ...
```

Execute workflows (monitored via polling)

```
>./diamond.py
```

```
...
starting runs:
=====
elapsed time 0.0 s   memory 102.16 MB
...
Entering ./runs/diamond/scf 0
Executing:
  export OMP_NUM_THREADS=1
  mpirun -np 4 pw.x -input scf.in

elapsed time 3.0 s   memory 102.29 MB
elapsed time 6.1 s   memory 102.32 MB
...
Entering ./runs/diamond/scf 1
Executing:
  export OMP_NUM_THREADS=1
  mpirun -np 1 pw2qmcpack.x<conv.in

elapsed time 9.1 s   memory 102.32 MB
elapsed time 12.2 s   memory 102.32 MB
...
Entering ./runs/diamond/vmc 2
Executing:
  export OMP_NUM_THREADS=4
  mpirun -np 4 qmcpack vmc.in.xml

elapsed time 15.2 s   memory 418.06 MB
...
elapsed time 45.6 s   memory 102.41 MB

Project finished
```

Check final run status

```
>./diamond.py --status_only
```

```
...
cascade status
  setup, sent_files, submitted, finished, ...
  111111 0 10641    scf      ./runs/diamond/scf
  111111 0 10723    conv     ./runs/diamond/scf
  111111 0 10742    vmc     ./runs/diamond/vmc
  setup, sent_files, submitted, finished, ...
```

Process results

```
grep '!  ' runs/diamond/scf/scf.out
!  total energy          =      -22.52257768 Ry
>qmca -e 20 -q e runs/diamond/vmc/*scalar*
runs/diamond/vmc/vmc series 0 LocalEnergy = -87.789516
+/- 0.032402
```

Working with Nexus: Changing Machine Environments

Workstation

```
settings(  
    ...  
    machine = 'ws4'  
)  
  
qmc = generate_qmcpack(  
    ...  
    job = job(cores = 4,  
              threads = 2),  
)
```

```
export OMP_NUM_THREADS=2  
mpirun -np 2 qmcpack qmc.in.xml
```

Institutional Cluster (CADES at ORNL)

```
settings(  
    ...  
    machine = 'cades',  
    account = 'qmc',  
)  
  
qmc = generate_qmcpack(  
    ...  
    job = job(nodes = 4,  
              threads = 4,  
              hours = 8),  
)
```

```
#!/bin/bash  
#SBATCH -A qmc  
#SBATCH -p skylake  
#SBATCH -J vmc  
#SBATCH -t 08:00:00  
#SBATCH -N 4  
#SBATCH --ntasks-per-node=9  
#SBATCH --cpus-per-task=4  
#SBATCH --mem=0  
#SBATCH -o vmc.out  
#SBATCH -e vmc.err  
#SBATCH --exclusive  
#SBATCH --export=ALL  
  
export OMP_NUM_THREADS=4  
mpirun -np 36 qmcpack vmc.in.xml
```

Leadership Computing (Perlmutter at NERSC)

```
settings(  
    ...  
    machine = 'perlmutter',  
    account = 'QMCACC',  
)  
  
qmc = generate_qmcpack(  
    ...  
    job = job(nodes = 128,  
              threads = 4,  
              hours = 4),  
)
```

```
#!/bin/bash  
#SBATCH -A QMCACC  
#SBATCH -C cpu  
#SBATCH -q regular  
#SBATCH -t 04:00:00  
#SBATCH -N 128  
#SBATCH --ntasks-per-node=32  
#SBATCH -c 8  
#SBATCH -J vmc  
#SBATCH -o vmc.out  
#SBATCH -e vmc.err  
#SBATCH --export=ALL  
  
export OMP_NUM_THREADS=4  
srun qmcpack vmc.in.xml
```

Supported Machine Environments

NERSC: Perlmutter

OLCF: Frontier, Andes

ALCF: Aurora, Improv

SNL: Sky Bridge, Serrano

NREL: Kestrel

Many Institutional Clusters

Adding a new Slurm Cluster

```
class Kestrel(Supercomputer):
    name = 'kestrel'
    requires_account = True
    batch_capable = True

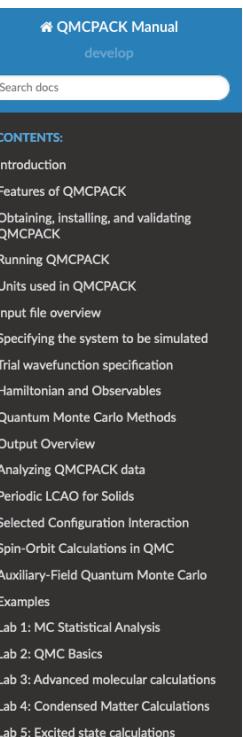
    def write_job_header(self, job):
        if job.queue is None:
            job.queue = 'regular'
        c = '#!/bin/bash\n'
        c += '#SBATCH -A ' + job.account + '\n'
        c += '#SBATCH -p ' + job.queue + '\n'
        c += '#SBATCH -J ' + str(job.name) + '\n'
        c += '#SBATCH -t ' + job.sbatch_walltime() + '\n'
        c += '#SBATCH -N ' + str(job.nodes) + '\n'
        c += '#SBATCH --ntasks-per-node=' + str(job.processes_per_node) + '\n'
        c += '#SBATCH --cpus-per-task=' + str(job.threads) + '\n'
        c += '#SBATCH -o ' + job.outfile + '\n'
        c += '#SBATCH -e ' + job.errfile + '\n'
        if job.user_env:
            c += '#SBATCH --export=ALL\n'
        else:
            c += '#SBATCH --export=NONE\n'
        c += ''
        echo $SLURM_SUBMIT_DIR
        cd $SLURM_SUBMIT_DIR
        ...
        return c
```

Adding new machines is easy: 30 lines of Python (contact developers)!

Nexus & QMCPACK Documentation

QMCPACK

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



QMCPACK Manual
develop

Search docs

CONTENTS:

- Introduction
- Features of QMCPACK
- Obtaining, installing, and validating QMCPACK
- Running QMCPACK
- Units used in QMCPACK
- Input file overview
- Specifying the system to be simulated
- Trial wavefunction specification
- Hamiltonian and Observables
- Quantum Monte Carlo Methods
- Output Overview
- Analyzing QMCPACK data
- Periodic LCAO for Solids
- Selected Configuration Interaction
- Spin-Orbit Calculations in QMC
- Auxiliary-Field Quantum Monte Carlo
- Examples
- Lab 1: MC Statistical Analysis
- Lab 2: QMC Basics
- Lab 3: Advanced molecular calculations
- Lab 4: Condensed Matter Calculations
- Lab 5: Excited state calculations

» User's Guide and Developer's Manual

Edit on GitHub

User's Guide and Developer's Manual

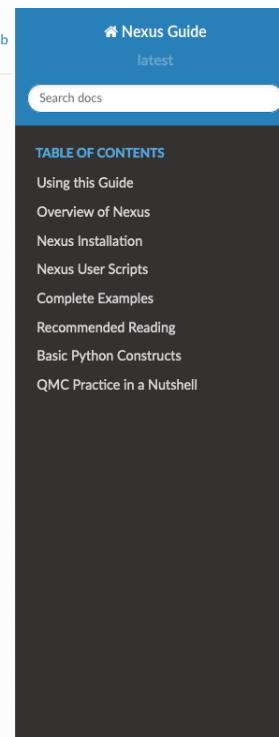
QMCPACK

Contents:

- Introduction
 - Quickstart and a first QMCPACK calculation
 - Authors and History
 - Support and Contacting the Developers
 - Performance
 - Open Source License
 - Contributing to QMCPACK
 - QMCPACK Roadmap
- Features of QMCPACK
 - Real-space Monte Carlo
 - Auxiliary-Field Quantum Monte Carlo
 - Supported GPU features for real space QMC
 - Sharing of spline data across multiple GPUs
- Obtaining, installing, and validating QMCPACK
 - Installation steps
 - Obtaining the latest release version
 - Obtaining the latest development version
 - Prerequisites

Nexus

<https://nexus-workflows.readthedocs.io/en/latest/>



Nexus Guide
latest

Search docs

TABLE OF CONTENTS

- Using this Guide
- Overview of Nexus
- Nexus Installation
- Nexus User Scripts
- Complete Examples
- Recommended Reading
- Basic Python Constructs
- QMC Practice in a Nutshell

» The Nexus User Guide

Edit on GitHub

The Nexus User Guide

Table of Contents

- Using this Guide
- Overview of Nexus
 - What Nexus is
 - What Nexus can do
 - How Nexus is used
- Nexus Installation
 - Setting environment variables
 - Installing Python dependencies
 - Testing your Nexus installation
- Nexus User Scripts
 - Nexus imports
 - Nexus settings: global state and user-specific information
 - Physical system specification
 - Workflow specification
 - Workflow execution
 - Data analysis
- Complete Examples
 - Example 1: Bulk Diamond VMC
 - Example 2: Graphene Sheet DMC

Summary

Proper statistical analysis (statistical significance, accounting for equilibration/autocorrelation effects, applying CLT) is essential for QMC practitioners.

QMC calculations are actually workflows of interconnected simulations; these can be managed efficiently with the Nexus workflow tool.

Lab examples: practice statistics skill in context of real VMC, wavefunction optimization, and DMC calculations with QMCPACK.

Next week: Molecular and Solid-State QMC calculations

Questions?

Hands-On: VMC, Jastrow Optimization, and DMC on the virtual machine

Accessing the Lab Material

Read and follow Lab README's via browser:

[https://github.com/QMCPACK/qmc_summer_school_2025
/tree/main/session3_statistics_and_workflows](https://github.com/QMCPACK/qmc_summer_school_2025/tree/main/session3_statistics_and_workflows)

← → 🔍 https://github.com/QMCPACK/qmc_workshop_2021/tree/master/week3_stats_and_nexus 120% ⭐ 3

QMCPACK / qmc_workshop_2021 Public Notifications Star 14 Fork 5

Code Pull requests Actions Security Insights

master

qmc_workshop_2021 / week3_stats_and_nexus /

Go to file

jtkrogel update readme

c63d459 3 days ago History

..

01_diamond_eos_workflow

update LiH

19 days ago

02_lih_hf_vmc

update readme

3 days ago

03_lih_opt_vmc

fix typos in readme

4 days ago

04_lih_dmc

update readme

3 days ago

pseudopotentials

add potentials

17 days ago

README.rst

3 days ago

README.rst

Statistical Analysis + Running QMC Workflows with QMCPACK and Nexus

In this lab, you will familiarize yourself with running VMC, wavefunction optimization, and DMC calculations with QMCPACK and analyzing the statistical results at each step. Quantum Monte Carlo (QMC) calculations join several constituent calculations (most related to obtaining the trial wavefunction) into a computational workflows.

This lab uses the Nexus workflow automation system to compactly represent simulation inputs and perform the DFT and QMC calculations in sequence. While the examples here are designed to run on a laptop, Nexus can be used on computational resources ranging from desktop machines to clusters and supercomputers, enabling production QMC calculations to be performed from within a uniform environment.

Follow README in each Examples' sub-directory

Starting the Lab: File Update & Navigation

Update workshop example files:

```
cd $HOME/qmc_summer_school_2025
```

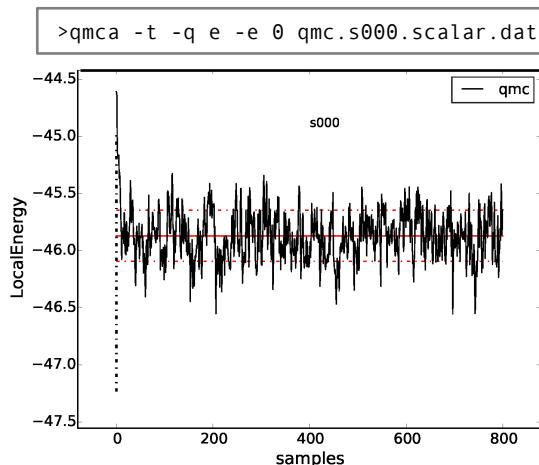
```
git pull
```

Enter session 3 lab directory:

```
cd $HOME/qmc_summer_school2025/session3_statistics_and_workflows
```

Statistical Analysis with the QMCA Tool

Make a Trace Plot



Remove Equilibration

```
>qmca -q e -e 100 qmc.s000.scalar.dat
```

qmc series 0 LocalEnergy = -45.877363 +/- 0.017432

Mean

Error-bar

Show Autocorrelation

```
>qmca -q e -e 100 qmc.s000.scalar.dat --sac
```

qmc series 0 LocalEnergy = -45.877363 +/- 0.017432 [4.8](#)

Energy, Variance, V/E ratio

```
>qmca -q ev opt*scalar.dat
          LocalEnergy      Variance      ratio
opt series 0 -44.823616 +/- 0.007430 7.054219 +/- 0.041998 0.1574
opt series 1 -45.877643 +/- 0.003329 1.095362 +/- 0.041154 0.0239
...
```

Full Documentation with More Examples

<https://qmcpack.readthedocs.io/en/develop/analyzing.html#qmca>

Example 1: Diamond Equation of State with Quantum Espresso

Directory

01_diamond_eos_workflow

Goals

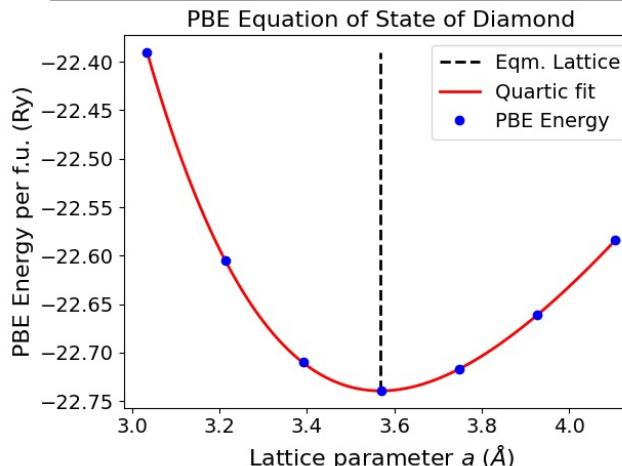
- Compute diamond energy vs. volume at 7 lattice constants and fit the EOS
- Gain familiarity with Nexus
- Learn major functions in user script

Running the Example

> ./diamond_eos.py

Example Results

```
>grep '! ' runs/a_*/*.out
runs/a_3.0345/scf.out:! total energy = -22.39012981 Ry
runs/a_3.2130/scf.out:! total energy = -22.60492242 Ry
runs/a_3.3915/scf.out:! total energy = -22.71014282 Ry
runs/a_3.5700/scf.out:! total energy = -22.73944516 Ry
runs/a_3.7485/scf.out:! total energy = -22.71717281 Ry
runs/a_3.9270/scf.out:! total energy = -22.66103531 Ry
runs/a_4.1055/scf.out:! total energy = -22.58424388 Ry
```



```
>fit_eos.py Edata.txt
Eqm lattice const: 3.5687 Å
```

Example 2: LiH Molecule Hartree Fock with PySCF and QMCPACK

Directory

02_lih_hf_vmc

Statistics/QMC Topics

- Estimating VMC equilibration time
- Estimating VMC energy mean, variance and error-bar
- Estimating VMC autocorrelation time
- Obtaining more precise estimates (Central limit theorem)

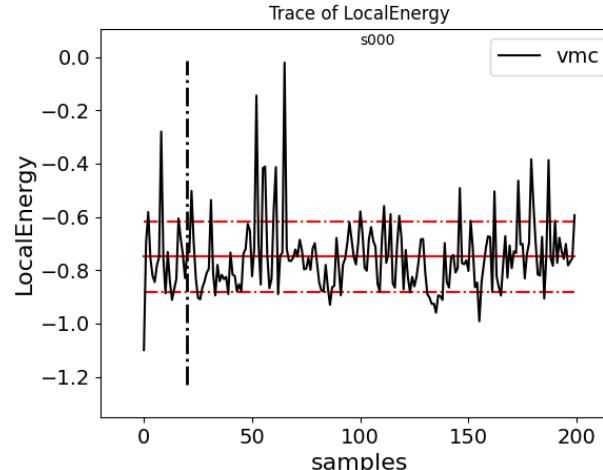
Running the Example

`>./lih_workflow.py`

Example Results

```
>cat runs/LiH/hf/scf.out
converged SCF energy = -0.749663314779369

>qmca -q ev -e 30 runs/LiH/vmc_hf/*scalar*
          LocalEnergy      Variance      ratio
vmc series 0 -0.747790 +/- 0.012450 0.130273 +/- 0.036354 0.1742
```



Example 3: LiH Molecule Jastrow Optimization and VMC with QMCPACK

Directory

03_lih_opt_vmc

Statistics/QMC Topics

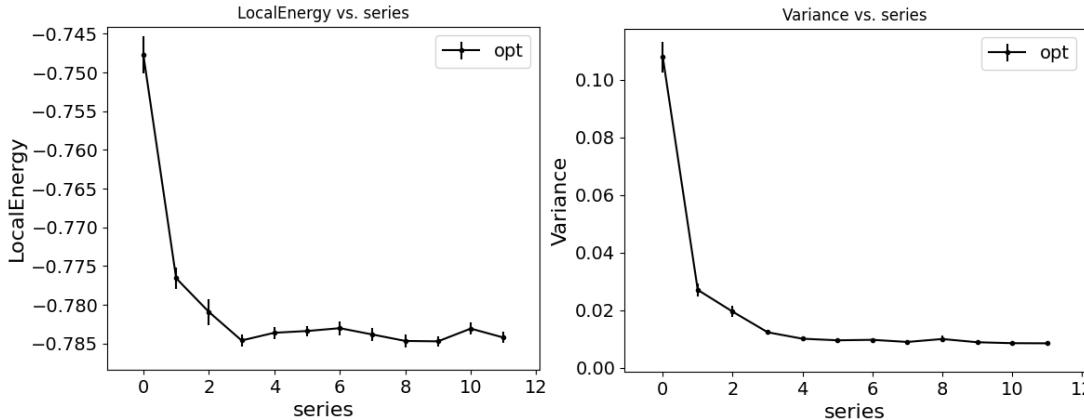
- Two body Jastrow form
- Performing Jastrow optimization with the “linear” method
- Judging wavefunction optimization
- Utility of variance/energy ratio
- Energy differences and convergence

Running the Example

`>./lih_workflow.py`

Example Results

```
>qmca -e 20 -q ev runs/LiH/opt/*scalar*
      LocalEnergy          Variance        ratio
opt series 0 -0.746956 +/- 0.002803 0.109144 +/- 0.005954 0.1461
opt series 1 -0.776778 +/- 0.001127 0.027871 +/- 0.002182 0.0359
opt series 2 -0.780653 +/- 0.001714 0.019933 +/- 0.001993 0.0255
opt series 3 -0.784852 +/- 0.000834 0.012195 +/- 0.001039 0.0155
...
opt series 8 -0.784658 +/- 0.000805 0.010037 +/- 0.001171 0.0128
opt series 9 -0.784699 +/- 0.000731 0.009089 +/- 0.000480 0.0116
opt series 10 -0.782754 +/- 0.000776 0.008703 +/- 0.000474 0.0111
opt series 11 -0.784038 +/- 0.000795 0.008558 +/- 0.000610 0.0109
```



Example 4: LiH Molecule Diffusion Monte Carlo with QMCPACK

Directory

04_lih_dmc

Statistics/QMC Topics

- Exponential DMC equilibration
- Longer autocorrelation time in DMC due to (necessary) small timestep
- DMC timestep extrapolation
- DMC population control bias
- Statistics of resampled curve fits

Running the Example

`>./lih_workflow.py`

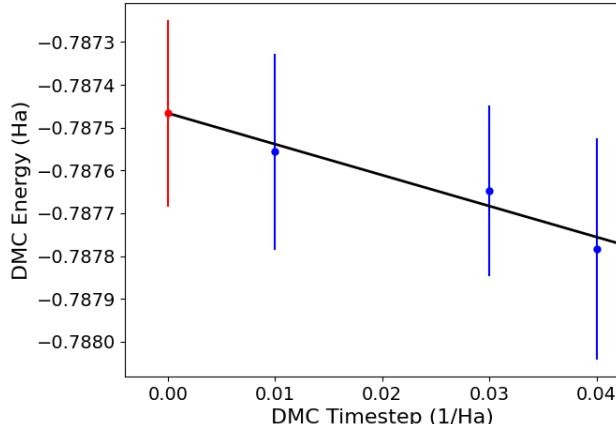
Example Results

```
>qmca -e 30 -q e runs/LiH/dmc_textrap/*s00{1,2,3}*scalar*
```

```
dmc series 1 LocalEnergy = -0.787784 +/- 0.000258
dmc series 2 LocalEnergy = -0.787648 +/- 0.000200
dmc series 3 LocalEnergy = -0.787557 +/- 0.000230
```

```
qmcfit ts -e 30 -b 8 -t '0.04 0.03 0.01' runs/LiH/dmc_textrap/*s00{1,2,3}*scalar*
```

```
fit function : linear
fitted formula: (-0.78747 +/- 0.00022) + (-0.0072 +/- 0.0076)*t
intercept : -0.78747 +/- 0.00022 Ha
```



DMC is expensive:
these runs will take
a few minutes!

Questions?