QMC Summer School 2025

Demystifying QMC Terminology

10 July 2025

Paul Kent, kentpr@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.

CPSFM Center for Predictive Simulation of Functional Materials

Sandia National Laboratories

Argonne NATIONAL LABORATORY

NC STATE UNIVERSITY

BROWN

OAK RIDGE National Laboratory

steps

walkers

timestep

Demystifying QMC Terminology

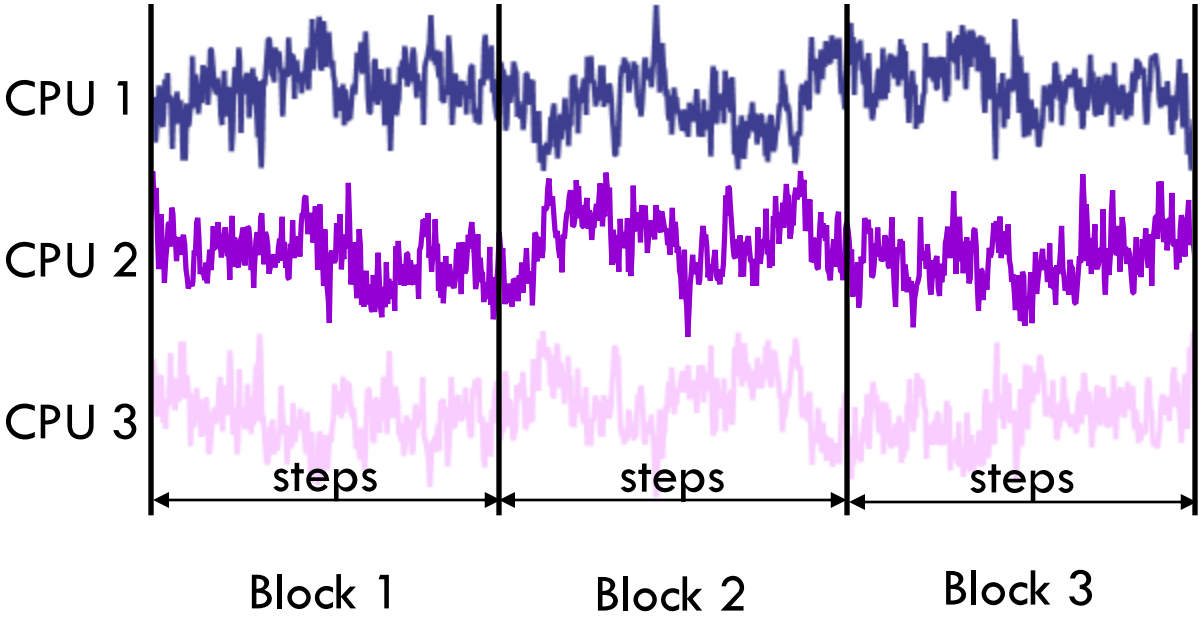
blocks

samples

VMC Inputs – blocks, steps

Run is subdivided into equal length "**blocks**" and summary statistical results written every block *.scalar.dat, *.stat.h5. Restart files/checkpoints written every N blocks.

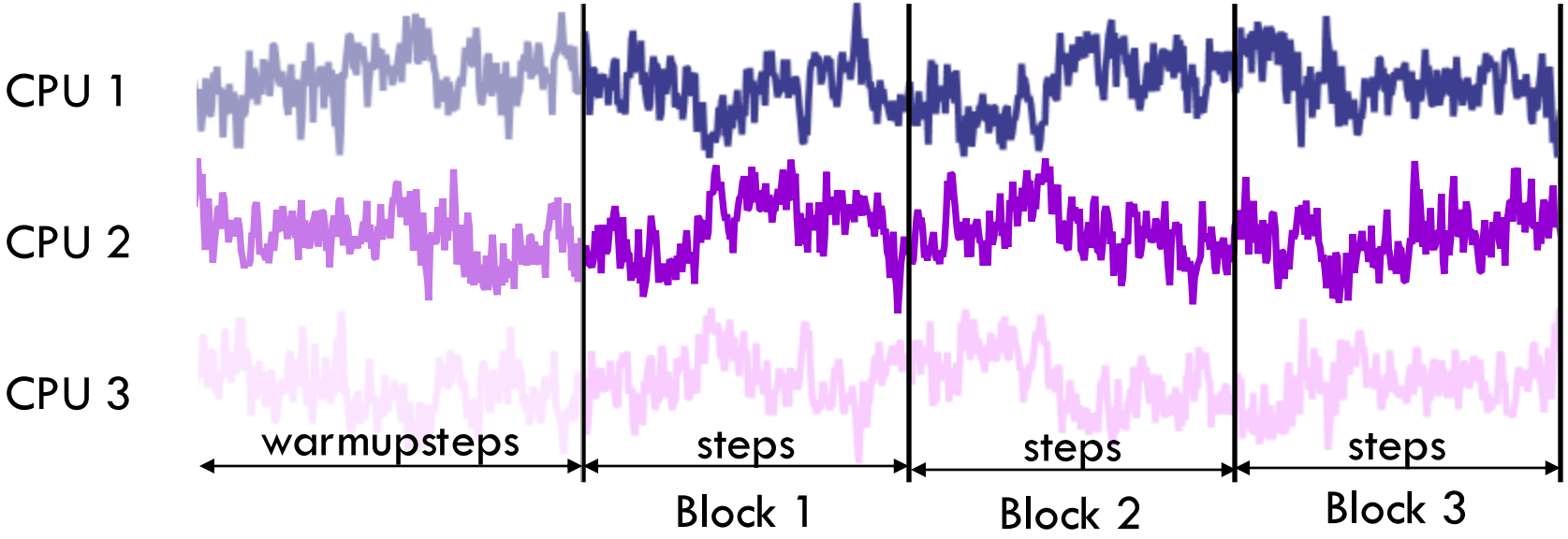
Each block consists of 1 or more "**steps**" where we make 1 attempt to move each electron in each walker by Monte Carlo and then measure properties.



VMC Inputs – timestep, warmupsteps

timestep – Controls size of displacement for proposed moves. Resulting acceptance ratio should not be too large (>>90%) or too small (<<10%) for efficiency.

warmupsteps - Steps without property measurement before first block (cheap). Not included in output statistics files. *Doesn't check equilibration completed.*



How many walkers in VMC?

VMC will default to 1 walker per CPU thread per MPI task (best use of resources)

OMP_NUM_THREADS=1 mpirun -n 4 qmcpack input.xml => 4 walkers

OMP_NUM_THREADS=8 mpirun -n 64 qmcpack input.xml => 512 walkers

Check output!

```
=============================================================
--- Memory usage report : VMCBatched before initialization ---
=============================================================
Available memory on node 0, free + buffers :    5324 MiB
Memory footprint by rank 0 on node 0       :      39 MiB
=============================================================
VMCBatched Driver running with
            total_walkers     = 4
            walkers_per_rank  = [1(x4)]
            num_crowds        = 1
  on rank 0, walkers_per_crowd = [1]

                       steps = 10
                      blocks = 400
```

Can also specify: total_walkers, walkers_per_rank. (old “walkers” is deprecated)

Note: VMC implementation can run any number of walkers. Critical for GPU runs.

How many property measurements?

total walkers x blocks x steps

Applies to all properties (observables)*.

You can get the same statistics by doubling the number of walkers and halving the step count.

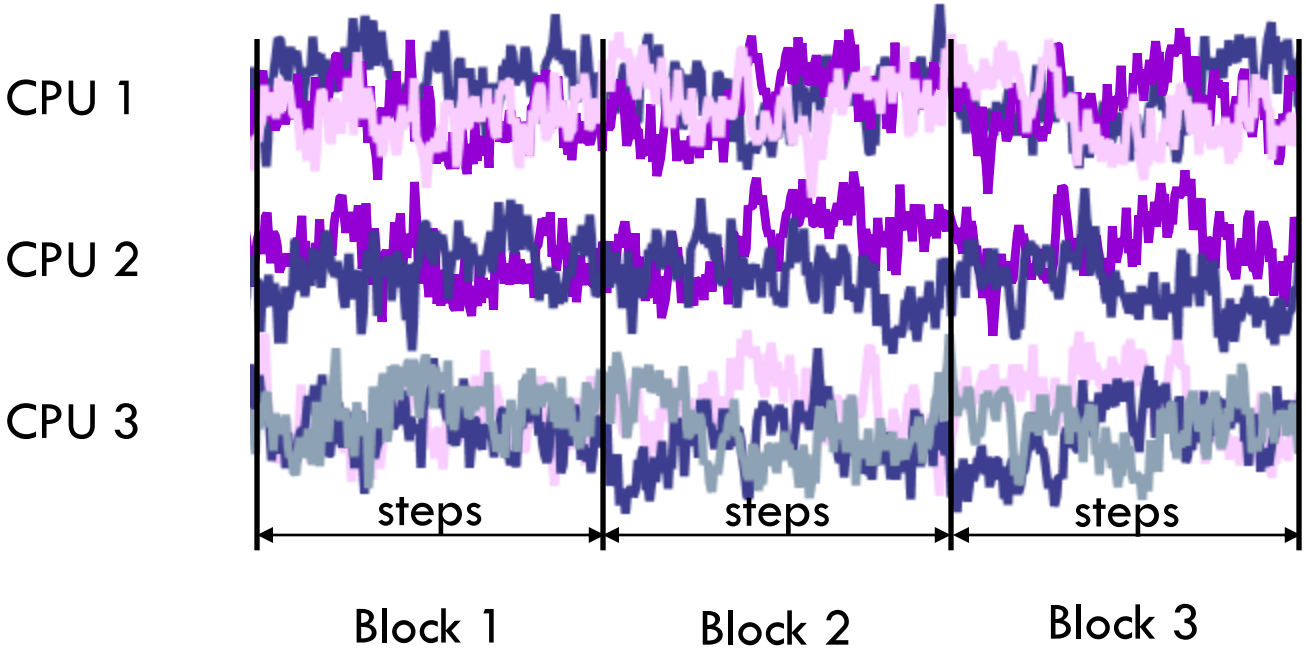
Reduce the error bar 2x by increasing the block count 4x.

* Ask if you have expensive observables, long correlation period

DMC Inputs – blocks, steps

Run is subdivided into equal length "**blocks**" and summary statistical results written every block *.scalar.dat, *.stat.h5. *.dmc.dat also updated every step.

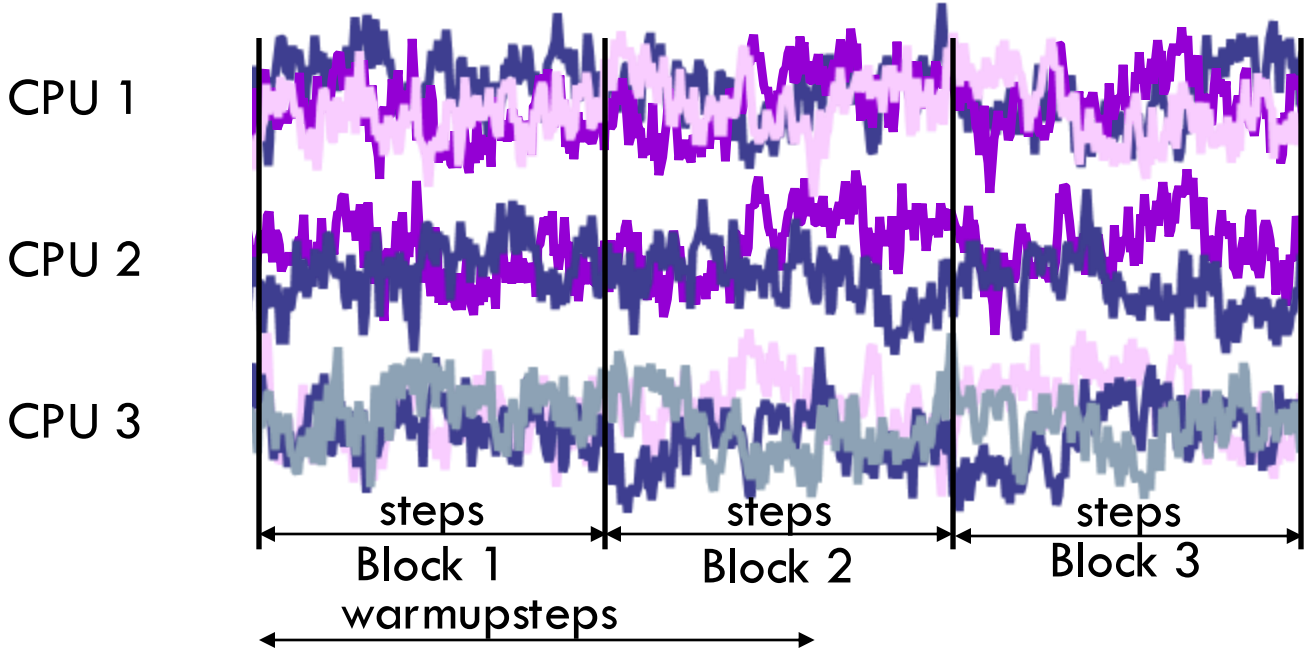
Each block consists of >=1 "**steps**" in the Monte Carlo where we make 1 attempt to move each electron in each walker, control population & measure properties.



DMC Inputs – timestep, warmupsteps

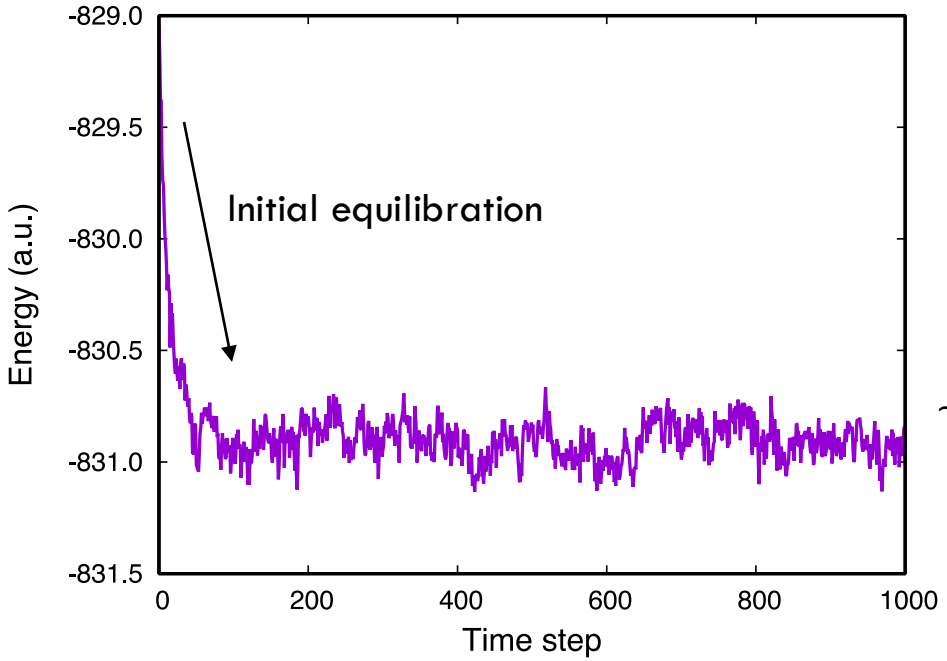
timestep – Controls size of displacement for proposed moves. Resulting acceptance ratio should very high(>>99%) for accuracy (timestep error).

warmupsteps - Steps for strong population control to aid equilibration. *Included in output files*. Doesn't check equilibration completed. Different behavior to VMC.



Real-world DMC: 48 atoms, bulk VO_2

Production run from Kylanpaa et al. PRM **1** 065408 (2017). 200 electrons, 0.01 a.u. timestep, wavefunction from DFT+U (U varied to minimize DMC energy).



Could run with
total_walkers=3000
steps=100
blocks=80
warmupsteps=200
Ignore first 4 blocks in final statistics

~8000 steps of 3000 walkers for statistics →

Average energy of walkers at each time step from *.dmc.dat

How many walkers in DMC?

Specify: total_walkers, walkers_per_rank. (old “walkers” is deprecated)

target_walkers : Goal for average population. Defaults to initial number of walkers. No need to set if total_walkers/walkers_per_rank set.

QMCPACK will load balance across MPI ranks/tasks.

Check output!

```
=============================================================
--- Memory usage report : DMCBatched before initialization ---
=============================================================
Available memory on node 0, free + buffers :    5324 MiB
Memory footprint by rank 0 on node 0       :      41 MiB
=============================================================
DMCBatched Driver running with
             total_walkers     = 128
             walkers_per_rank  = [32(x4)]
             num_crowds        = 1
  on rank 0, walkers_per_crowd = [32]

                         steps = 10
                        blocks = 200
```

Note: DMC implementation can necessarily run any number of walkers.

How many property measurements?

Very approximately total walkers x blocks x steps

Only true if population is very well controlled. Need to compute averages, totals etc. from actual run.

You can get the same statistics by doubling the number of walkers and halving the step count.

Reduce the error bar 2x by increasing the block count 4x.

* Ask if you have expensive observables, long correlation period

Additional Settings - samples

samples – Sets total number of measurements (=walkers*steps*blocks) fed into optimizer. Used only for wavefunction optimization in QMCPACK and NEXUS. Allows specifying only blocks & samples; QMCPACK computes correct step count.

QMCPACK may overshoot if walkers & blocks are not factors of samples.

Samples could be large! 10^3-10^6

Future versions of NEXUS and QMCPACK will support this more generally to simplify specifying runs. Currently only fully implemented in wavefunction optimization.

Summary

Control the number of measurements in VMC and DMC via blocks, steps, and total walker count

Commentary

QMCPACK will respect your inputs where possible (will warn or abort if not). Good defaults are used for any unspecified parameter. Output includes enough detail to confirm choices.

Aim to keep inputs to QMCPACK and NEXUS as simple as possible.

All inputs are documented – search on qmcpack.readthedocs.io, or browse https://qmcpack.readthedocs.io/en/develop/methods.html . There are tables of input parameters for each QMC method.