

QMCPACK

Engaging with the Project

Paul Kent, kentpr@ornl.gov

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.

Goals

Outline how to get started, the different QMCPACK websites, and means of interacting with the project

Highlight some new developments

Get suggestions for improvements

QMCPACK.org

Main website for the code. Includes links to releases, news, workshop info etc.

We would like to highlight your papers & science results!

The screenshot shows the QMCPACK.org website. At the top, there's a navigation bar with links for "About QMCPACK", "Downloads", "Documentation", "Nexus", "Pseudopotential Library", a search bar, and a "Home | QMCPACK" link. The main content area features a large molecular simulation image with a red-to-yellow gradient ribbon. A dark overlay box contains the text: "Correlated Defects via QMC: Mn doped phosphors" and "Quantum Monte Carlo is applied to phosphors for the first time." Below this, there are sections for "Downloads" (with a "Latest release" link) and "QMCPACK Users Workshop 2019" (with details about the event date, location, and registration). The bottom right corner of the slide has the text "Workshop 2019".

Workshop 2019

docs.qmcpack.org

Contains PDF manuals from current and previous versions.

Manual includes a list of all QMCPACK papers
– keep us updated or update
qmcpack_papers.bib directly

We make improvements, additions, and fill gaps in part based on requests. E.g. Currently being edited.

QMCPACK

User's Guide and Developer's Manual
Development Version
May 13, 2019

QMCPACK website: <http://www.qmcpack.org>

Releases and source code: <https://github.com/QMCPACK>

Google Group: <https://groups.google.com/forum/#!forum/qmcpack>

Latest manual: https://docs.qmcpack.org/qmcpack_manual.pdf

GitHub.com/qmcpack/qmcpack

Live development repository with complete version history

Has a searchable list of issues including all reported bugs, discussion questions, enhancement requests...

The screenshot shows the GitHub repository page for `QMCPACK / qmcpack`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The Issues tab is selected, showing 227 open issues. A banner at the top indicates there are 8 pull requests, 0 projects, 1 wiki, and 57 forks. Below the tabs, a section titled "Pinned issues" displays two pinned issues:

- QMCPACK workshop 14-15 May 2019** (#1456) - Opened on Mar 15 by prkent. One comment.
- Full list of known gaps and tasks for AoS to SoA transition** (#861) - Opened on May 11, 2018 by ye-luo. Nine comments.

Below the pinned issues, there are filters for "Filters", "Labels 18", "Milestones 1", and a "New issue" button. The main list of issues shows the following entries:

- pw2qmcpack.x error with large cell/cutoff** (#1593) - Opened 3 days ago by kayahans. Three comments.
- Issues with Cartesian Spherical Harmonics between QMCPACK and pyscf** (#1592) - Opened 3 days ago by rcclay. One comment.
- Change of DriftModifier type not allowed when reusing a driver** (#1589) - Opened 5 days ago by ye-luo. Three comments.

Workshop 2019

Pseudopotentiallibrary.org

ccECPs and other recent pseudopotentials/ECPs in formats suitable for QMCPACK, quantum chemistry and condensed-matter codes. Feedback and contributions welcome.

A screenshot of a web browser displaying the 'Pseudopotential Library' page from pseudopotentiallibrary.org. The page title is 'Pseudopotential Library'. Below the title, a subtitle reads: 'A community website for pseudopotentials/effective core potentials developed for high accuracy correlated many-body methods such as quantum Monte Carlo and quantum chemistry.' The main feature is a large grid of colored squares, each containing an element symbol. The grid is organized into several rows and columns. The first row contains H (yellow), He (light blue), and empty squares. The second row contains Li (red), Be (orange), and empty squares. The third row contains Na (red), Mg (orange), and empty squares. Subsequent rows contain various elements like K, Ca, Sc, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Ga, Ge, As, Se, Br, Kr, Rb, Sr, Y, Zr, Nb, Mo, Te, Ru, Rh, Pd, Ag, Cd, In, Sn, Sb, Te, I, Xe, Cs, Ba, La, HF, Ta, W, Re, Os, Ir, Pt, Au, Hg, Tl, Pb, Bi, Po, At, Rn, Fr, Ra, Ac, Rf, Db, Sg, Bh, Hs, Mt, Ds, Rg, Cn, Nh, Fl, Mc, Lv, Ts, Og, and Cf, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Th, Pa, U, Np, Pu, Am, Cm, Bk, Cf, Es, Fm, Md, No, Lr. Below the grid, a message says: 'Click an element above to view available recipes.'

Help, Discussion

QMCPACK Google Groups

GitHub

Contact a developer:

Comment at this workshop

Email

QMCPACK Slack

Groups & GitHub preferred: public, searchable, anyone can see & comment...

Be sure to try the latest release, state where you are running, how built, and try to make a small reproducer

Releases, Versions

We recommend to use the latest release version (v3.7.0) for research. A new release will be made after workshop dust has settled.

Develop version has latest features but fewer guarantees; only use if essential or developing on QMCPACK

Change Log

Notable changes to QMCPACK are documented in this file.

[3.7.0] - 2019-03-29

Notes

This release includes GPU support for the AFQMC implementation, Quantum Espresso v6.4 support, and in the real-space code makes the structure-of-arrays (SoA) code path the default. A large number of feature refinements, bugfixes, testing improvements and source code cleanup have been performed.

- The improved structures of arrays (SoA) build is now the default. This is generally significantly faster and uses less memory than the AoS build due to better algorithms, but does not yet have the full range of functionality. The older AoS build can be selected with `-DENABLE_SOA=0`.
- AFQMC code fully supports GPU acceleration via NVIDIA CUDA. Use `-DENABLE_CUDA=1`.
- Quantum Espresso v6.4 is supported. [#1457](#)

Installing

The manual has instructions and extensive examples for different systems, including HPC. Use as a starting point for similar systems.

Try to use an up-to-date system with modern software.

e.g. With Intel compilers, MPI & MKL (free for students)

```
mkdir build; cd build
```

```
cmake -DCMAKE_C_COMPILER=mpiicc -DCMAKE_CXX_COMPILER=mpiicpc ..
```

```
make -j 8
```

Testing

As described in the manual we recommend running tests to verify QMCPACK installation

```
ctest -R unit # Run all the unit tests. Takes O(1min).
```

```
ctest -E long # Run all the tests except the long ones. Takes O(1h).
```

NEW Deterministic tests are reliable consistency checks that run in a few minutes total. Avoids inherent unreliability of statistical tests. These tests should always pass.

```
ctest -R deterministic
```

“Missing” tests

We have high confidence in the core QMC algorithms and for features where there are test. E.g. DMC with splines, Gaussians, common jastrows... (many hundreds of tests)

Coverage is sparse for many observables and older wavefunction types

We recommend checking status and performing your own checks where there are no tests, or even adding a test. This will help keep a feature healthy.

Let us know what you need!

Testing Dashboard cdash.qmcpack.org

The screenshot displays the CDash testing dashboard for the QMCPACK project. The top navigation bar includes links for Login, All Dashboards, and the current project. The main content area is divided into two sections: 'Deterministic' and 'Converter'. Each section contains a table with columns for Site, Build Name, Test (Not Run, Fail, Pass), and Start Time.

Deterministic

Site	Build Name	Test			Start Time
		Not Run	Fail	Pass	
oxygen.ornl.gov	CLANG6-CUDA-Release	0	54	37	May 12, 2019 - 21:02 UTC
bora.alcf.anl.gov	Intel2018-Complex-Mixed-SoA-CUDA2-Release	0	0	91	May 12, 2019 - 13:11 UTC
bora.alcf.anl.gov	Intel2018-Real-Mixed-SoA-CUDA2-Release	0	0	121	May 12, 2019 - 12:53 UTC
oxygen.ornl.gov	Intel2018-Complex-SoA-Release	0	0	91	May 12, 2019 - 12:30 UTC
cetus.alcf.anl.gov	Clang-Complex-SoA-Release	0	0	18	May 12, 2019 - 12:19 UTC
bora.alcf.anl.gov	Intel2018-Complex-Mixed-Release	0	0	91	May 12, 2019 - 11:55 UTC
cetus.alcf.anl.gov	Clang-Real-SoA-Release	0	0	18	May 12, 2019 - 11:21 UTC
bora.alcf.anl.gov	Intel2018-Complex-Release	0	0	91	May 12, 2019 - 11:07 UTC
cetus.alcf.anl.gov	Clang-Complex-Mixed-SoA-Release	0	0	18	May 12, 2019 - 10:19 UTC
bora.alcf.anl.gov	Intel2018-Real-Mixed-Release	0	0	121	May 12, 2019 - 10:07 UTC

Items per page: 10

Converter

Site	Build Name	Test			Start Time
		Not Run	Fail	Pass	
oxygen.ornl.gov	CLANG6-CUDA-Release	0	0	14	May 12, 2019 - 21:03 UTC
bora.alcf.anl.gov	Intel2018-Complex-Mixed-SoA-CUDA2-Release	0	0	14	May 12, 2019 - 13:11 UTC
bora.alcf.anl.gov	Intel2018-Real-Mixed-SoA-CUDA2-Release	0	0	14	May 12, 2019 - 12:53 UTC
cetus.alcf.anl.gov	Clang-Complex-SoA-Release	0	0	0	May 12, 2019 - 12:31 UTC
oxygen.ornl.gov	Intel2018-Complex-SoA-Release	0	0	14	May 12, 2019 - 12:30 UTC
bora.alcf.anl.gov	Intel2018-Complex-Mixed-Release	0	0	62	May 12, 2019 - 11:55 UTC

Test results for the latest development version using many different compilers, library versions, processors etc.

SPACK <https://spack.io>

Spack is a relatively new “version aware” package manager that can install in userspace. i.e. No privileges required.

Spack will install compilers and all needed libraries in addition to QMCPACK, QE, and many other applications.

Can be fragile but well worth a try and potentially a big time saving.

```
git clone https://github.com/spack/spack.git .
spack/share/spack/setup-env.sh
spack install qmcpack    # See spack docs for
                           # configuration help
```

Sources of Trial Wavefunctions

Quantum Espresso

GAMESS

PySCF

Quantum Package (QP)

Qbox prototype (<https://github.com/QMCPACK/qmcpack/pull/1425>)

Other sources considered after weighing distinctive features & science potential, userbase, difficulty...

Amazon Web Services Instance

Based on feedback from this workshop we will consider publishing and maintaining an image (AMI) with QMCPACK.

This would provide a fast route to trying out the code and the surrounding ecosystem. Needs some knowledge of AWS to use.

Note: We can pay for occasional usage, but not act as an ongoing computational resource.

Suggestions?

Note: Feedback on all aspects of this workshop is appreciated