The Journal of Open Source Software

Open Computational Chemistry (OCC) - A portable software library and program for quantum chemistry and crystallography

Peter R. Spackman [1¶]

1 School of Molecular and Life Sciences, Curtin University, Perth, WA 6845, Australia ¶ Corresponding author

DOI: 10.21105/joss.09609

Software

- Review
- Repository
- Archive

Editor: Rachel Kurchin

Reviewers:

- @muralidhar-nalabothula
- @sef43

Submitted: 21 October 2025
Published: 27 January 2026

Summary

Open Computational Chemistry (OCC) is a modern software library designed for calculating the properties and electronic structure and interactions of molecules and molecular crystals. The software serves a dual purpose: it provides an open, accessible platform for researchers developing new computational methods, while also being production-ready and fast enough for relatively large-scale calculations on personal computers and high-performance computing systems alike.

Many traditional computational chemistry programs require complex installation procedures or rely on large dependencies can only run on specific operating systems. OCC is designed as a priority to be portable: it runs on Windows, macOS, and Linux, provides interfaces for C++, Python, and JavaScript, and with the capability to even run entirely within web browsers through WebAssembly (WASM), making computational chemistry accessible without specialized hardware or software installation. OCC is already widely used as the primary computational backend of CrystalExplorer (Spackman et al., 2021), a widely-used graphical program for crystal structure analysis, providing capabilities such as automatic calculation of interaction energies between molecules in crystals, accurate determination of total crystal energies with automatic convergence checking, and evaluation of electron density distributions.

Statement of need

The landscape of computational chemistry software is dominated by established packages that, while powerful, often suffer from legacy codebases, limited portability, and barriers to modification. Most existing quantum chemistry software requires complex installation procedures and system-level dependencies, limiting accessibility for educational purposes and rapid prototyping. Furthermore, the integration between quantum chemistry and crystallography remains fragmented across different specialized tools, requiring researchers to chain together multiple incompatible programs for multi-scale workflows.

OCC addresses these challenges as an open, portable, and modifiable platform for computational chemistry and crystallography. The software serves dual purposes: as a computational backend for production applications like CrystalExplorer, a widely-adopted graphical tool for crystal structure analysis, and as a research platform for developing new computational methods. This distinguishes OCC from purely academic codes (often limited in scope, modifiability or performance) and purely commercial alternatives (which lack transparency and modifiability). In a similar spirit to Psi4 (Smith et al., 2020) and PySCF (Sun et al., 2020), OCC prioritises extensibility and openness, with a unique focus on the properties and interactions of molecular crystals.

Key features enabling this versatility include:

Portability and Accessibility: OCC is, to our knowledge, the first full-featured quantum chemistry package that can run entirely in web browsers via WASM, enabling interactive calculations without installation or specialized hardware, on computers, tablets, phones or any device with a browser supporting WASM. Native builds support Windows, macOS, and Linux with comprehensive continuous integration testing, while language bindings for C++, Python, and JavaScript facilitate integration into diverse workflows.

Unified Framework: By integrating electronic structure methods with crystallographic analysis in a single modern C++ codebase, OCC eliminates the friction of multi-scale workflows. Unique capabilities include crystal growth free energy predictions bridging molecular-level calculations and macroscopic properties, automatic lattice energy convergence, and pairwise interaction energy calculations used in production by CrystalExplorer.

Open and Modifiable: The codebase follows contemporary software development best practices including comprehensive continuous integration testing across all platforms, automated releases for Python packages via PyPI, and Javascript via NPM and well-documented APIs with dependency management via CPM.cmake. This enables researchers to rapidly develop and test new methods. Unlike monolithic legacy codes, OCC's modular architecture facilitates extension and modification for specific research needs.

Implementation and Features

OCC implements Hartree-Fock and Density Functional Theory (DFT) with support for LDA, GGA, and meta-GGA functionals via libxc (Lehtola et al., 2018), density fitting (RI-JK) methods for Coulomb and exchange interactions, implicit solvation via COSMO (Klamt & Schüürmann, 1993) and SMD (Marenich et al., 2009), and dispersion corrections (XDM (Becke & Johnson, 2007; Johnson & Becke, 2006; Otero-de-la-Roza & Johnson, 2012), DFTD4 (Caldeweyher et al., 2017, 2019, 2020)). For crystallography, OCC provides CIF file processing via gemmi (Wojdyr, 2022), fast periodic bond detection, symmetry-unique molecule generation, CrystalExplorer model energies (Mackenzie et al., 2017; Spackman, Spackman, et al., 2023), and automatic pair-based lattice energy summation for neutral molecular crystals with symmetry.

Unique capabilities include crystal growth free energy predictions (Spackman, Walisinghe, et al., 2023) combining lattice energies, interaction energy decomposition, and vibrational/configurational entropy contributions. The distributed multipole analysis (DMA) (Stone, 2005) implementation provides multipole expansions up to hexadecapole level with GDMA-compatible output. WebAssembly compilation via Emscripten (Zakai, 2011) enables full-featured quantum chemistry calculations directly in web browsers, supporting interactive educational tools and client-side computational workflows without server infrastructure.

OCC employs a modular C++ architecture using Eigen3 (Guennebaud et al., 2010) for linear algebra, libcint (Sun, 2015) for integral evaluation, and libecpint (R. A. Shaw & Hill, 2017; R. Shaw & Hill, 2021) for effective core potentials.

Performance

Native builds provide performance comparable to established quantum chemistry packages through optimized integral evaluation, SIMD instructions, and parallel execution, while WebAssembly builds offer unprecedented accessibility with acceptable performance overhead for educational and prototyping use cases.

Availability and Documentation

OCC is available as open-source software under the GNU General Public License v3 at https://github.com/peterspackman/occ. Pre-built Python packages are available via PyPI (occpy). WebAssembly builds can be integrated via npm or used directly in browsers. Comprehensive continuous integration testing ensures reliability across Windows, macOS, and Linux platforms for all language bindings (C++, Python, JavaScript/WASM). Documentation at https://getocc.xyz includes interactive tutorials, where the code runs in the browser, complete C++ API references, and examples.

Conclusions

OCC provides a modern, accessible platform for quantum chemistry and crystallography that serves both as a research tool for method development and as a production-ready library powering applications like CrystalExplorer. Its unique combination of traditional electronic structure methods, crystallographic analysis, and WebAssembly support addresses key accessibility and integration challenges in computational chemistry. Future development will focus on expanding functional support, implementing excited state methods, and adding GPU acceleration.

Acknowledgements

OCC would not exist if it weren't for Tonto (Jayatilaka & Grimwood, 2003) its predecessor library and program at the backend of CrystalExplorer.

OCC builds upon a foundation of high-quality open-source libraries, and we gratefully acknowledge the developers and maintainers of these projects:

Core computational libraries: Eigen3 (Guennebaud et al., 2010) for linear algebra operations, libcint (Sun, 2015) for Gaussian integral evaluation, libxc (Lehtola et al., 2018) for exchange-correlation functionals, libecpint (R. Shaw & Hill, 2021) for effective core potential integrals, and gau2grid (Smith et al., 2025) for efficient grid evaluation of Gaussians.

Crystallographic and optimization tools: gemmi (Wojdyr, 2022) for CIF file handling and crystallographic operations, and LBFGS++ (Qiu, 2025) for optimization algorithms. The molecular geometry optimization implementation was significantly informed by pyberny (Herrmann, 2019).

Infrastructure and utilities: fmt (Churaev, 2025) for string formatting, spdlog (Melman, 2025) for logging, oneTBB (*oneAPI Threading Building Blocks (oneTBB)*, 2025) for parallelization, CLI11 (Schreiner, 2025) for command-line parsing, nlohmann/json (Lohmann, 2025) for JSON handling, scnlib (Kosunen, 2025) for string parsing, and unordered_dense (Ankerl, 2025) for efficient hash containers.

Development tools: nanobind (Jakob, 2022) for Python bindings, Emscripten (Zakai, 2011) for WebAssembly compilation, and CPM.cmake (Melzer, 2024) for dependency management.

References

Ankerl, M. L.-. (2025). *Unordered_dense: A fast & densely stored hashmap and hashset based on robin-hood backward shift deletion*. https://github.com/martinus/unordered_dense

Becke, A. D., & Johnson, E. R. (2007). Exchange-hole dipole moment and the dispersion interaction revisited. *The Journal of Chemical Physics*, *127*(15). https://doi.org/10.1063/1.2795701

Caldeweyher, E., Bannwarth, C., & Grimme, S. (2017). Extension of the D3 dispersion coefficient model. *The Journal of Chemical Physics*, *147*(3). https://doi.org/10.1063/1.4993215

Caldeweyher, E., Ehlert, S., Hansen, A., Neugebauer, H., Spicher, S., Bannwarth, C., & Grimme, S. (2019). A generally applicable atomic-charge dependent London dispersion correction. *The Journal of Chemical Physics*, *150*(15). https://doi.org/10.1063/1.5090222

Caldeweyher, E., Mewes, J.-M., Ehlert, S., & Grimme, S. (2020). Extension and evaluation of the D4 London-dispersion model for periodic systems. *Physical Chemistry Chemical Physics*, *22*(16), 8499–8512. https://doi.org/10.1039/d0cp00502a

Churaev, V. (2025). *Fmt: A modern formatting library*. https://github.com/fmtlib/fmt

Guennebaud, G., Jacob, B., & others. (2010). *Eigen v3*. http://eigen.tuxfamily.org

Herrmann, J. (2019). *Pyberny: Molecular structure optimizer*. https://github.com/jhrmnn/pyberny

Jakob, W. (2022). *Nanobind: Tiny and efficient C++/Python bindings*.

Jayatilaka, D., & Grimwood, D. J. (2003). Tonto: A Fortran based object-oriented system for quantum chemistry and crystallography. *Proceedings of the 2003 International Conference on Computational Science*, 142–151. https://doi.org/10.1007/3-540-44864-0_15

Johnson, E. R., & Becke, A. D. (2006). A post-Hartree-Fock model of intermolecular interactions: Inclusion of higher-order corrections. *The Journal of Chemical Physics*, *124*(17). https://doi.org/10.1063/1.2190220

Klamt, A., & Schüürmann, G. (1993). COSMO: A new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2*, *5*, 799–805. https://doi.org/10.1039/p29930000799

Kosunen, E. (2025). *Scnlib: Scanf for modern C++*. https://github.com/eliaskosunen/scnlib

Lehtola, S., Steigemann, C., Oliveira, M. J., & Marques, M. A. (2018). Recent developments in libxc: A comprehensive library of functionals for density functional theory. *SoftwareX*, *7*, 1–5. https://doi.org/10.1016/j.softx.2017.11.002

Lohmann, N. (2025). *JSON for modern C++*. https://github.com/nlohmann/json

Mackenzie, C. F., Spackman, P. R., Jayatilaka, D., & Spackman, M. A. (2017). CrystalExplorer model energies and energy frameworks: Extension to metal coordination compounds, organic salts, solvates and open-shell systems. *IUCrJ*, *4*(5), 575–587.

Marenich, A. V., Cramer, C. J., & Truhlar, D. G. (2009). Universal solvation model based on solute electron density and on a continuum model of the solvent defined by the bulk dielectric constant and atomic surface tensions. *The Journal of Physical Chemistry B*, *113*(18), 6378–6396. https://doi.org/10.1021/jp810292n

Melman, G. (2025). *Spdlog: Fast C++ logging library*. https://github.com/gabime/spdlog

Melzer, L. (2024). *CPM.cmake: A setup-free CMake dependency management*. https://github.com/cpm-cmake/CPM.cmake

oneAPI Threading Building Blocks (oneTBB). (2025). https://github.com/uxlfoundation/oneTBB

Otero-de-la-Roza, A., & Johnson, E. R. (2012). Van der Waals interactions in solids using the exchange-hole dipole moment model. *The Journal of Chemical Physics*, *136*(17). https://doi.org/10.1063/1.4705760

Qiu, Y. (2025). *LBFGS++: A header-only C++ library for L-BFGS and L-BFGS-B algorithms*. https://github.com/yixuan/LBFGSpp

Schreiner, H. (2025). *CLI11: Command line parser for C++11*. https://github.com/CLIUtils/CLI11

Shaw, R. A., & Hill, J. G. (2017). Prescreening and efficiency in the evaluation of integrals over ab initio effective core potentials. *The Journal of Chemical Physics*, *147*(7). https://doi.org/10.1063/1.4986887

Shaw, R., & Hill, J. (2021). libecpint: A C++ library for the efficient evaluation of integrals over effective core potentials. *Journal of Open Source Software*, *6*(60), 3039. https://doi.org/10.21105/joss.03039

Smith, D. G. A., Burns, L. A., Simmonett, A. C., Parrish, R. M., Schieber, M. C., Galvelis, R., Kraus, P., Kruse, H., Di Remigio, R., Alenaizan, A., James, A. M., Lehtola, S., Misiewicz, J. P., Scheurer, M., Shaw, R. A., Schriber, J. B., Xie, Y., Glick, Z. L., Sirianni, D. A., … Sherrill, C. D. (2020). Psi4 1.4: Open-source software for high-throughput quantum chemistry. *The Journal of Chemical Physics*, *152*(18). https://doi.org/10.1063/5.0006002

Smith, D. G. A., Parrish, R. M., Burns, L. A., Simmonett, A. C., & Pritchard, B. (2025). *gau2grid: Fast computation of a Gaussian and its derivative on a grid*. https://github.com/psi4/gau2grid

Spackman, P. R., Spackman, M. A., & Gale, J. D. (2023). A transferable quantum mechanical energy model for intermolecular interactions using a single empirical parameter. *IUCrJ*, *10*(6), 754–765. https://doi.org/10.1107/s2052252523008941

Spackman, P. R., Turner, M. J., McKinnon, J. J., Wolff, S. K., Grimwood, D. J., Jayatilaka, D., & Spackman, M. A. (2021). *CrystalExplorer*: a program for Hirshfeld surface analysis, visualization and quantitative analysis of molecular crystals. *Journal of Applied Crystallography*, *54*(3). https://doi.org/10.1107/S1600576721002910

Spackman, P. R., Walisinghe, A. J., Anderson, M. W., & Gale, J. D. (2023). CrystalClear: An open, modular protocol for predicting molecular crystal growth from solution. *Chemical Science*, *14*(26), 7192–7207. https://doi.org/10.1039/d2sc06761g

Stone, A. J. (2005). Distributed multipole analysis: stability for large basis sets. *Journal of Chemical Theory and Computation*, *1*(6), 1128–1132. https://doi.org/10.1021/ct050190+

Sun, Q. (2015). Libcint: An efficient general integral library for Gaussian basis functions. *Journal of Computational Chemistry*, *36*, 1664–1671. https://doi.org/10.1002/jcc.23981

Sun, Q., Zhang, X., Banerjee, S., Bao, P., Barbry, M., Blunt, N. S., Bogdanov, N. A., Booth, G. H., Chen, J., Cui, Z.-H., Eriksen, J. J., Gao, Y., Guo, S., Hermann, J., Hermes, M. R., Koh, K., Koval, P., Lehtola, S., Li, Z., … Chan, G. K.-L. (2020). Recent developments in the PySCF program package. *The Journal of Chemical Physics*, *153*(2). https://doi.org/10.1063/5.0006074

Wojdyr, M. (2022). GEMMI: A library for structural biology. *Journal of Open Source Software*, *7*(73), 4200. https://doi.org/10.21105/joss.04200

Zakai, A. (2011). Emscripten: An LLVM-to-JavaScript compiler. *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, 301–312. https://doi.org/10.1145/2048147.2048224