The Journal of Open Source Education

PyQInt: A Teaching-Oriented Hartree–Fock Implementation in Python

I. A. W. Filot [1¶]

1 Inorganic Materials and Catalysis, Department of Chemical Engineering and Chemistry, Eindhoven University of Technology ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 24 April 2025
Published: unpublished

Summary

`PyQInt` is a modular Python package for learning and prototyping quantum chemistry methods, with a particular focus on the Hartree-Fock formalism(Roothaan, 1951) using Gaussian-type orbitals.(Pople et al., 1995) Designed to prioritize educational transparency, `PyQInt` exposes all computational building blocks—integrals, matrices, Hamiltonians, SCF procedures, and gradients—through a clean, inspectable API.

Users can evaluate molecular integrals(Taketa et al., 1966), perform self-consistent field calculations with direct inversion of iterative subspace (DIIS)(Pulay, 1980), construct and localize orbitals, compute crystal orbital Hamilton population (COHP) coefficients(Dronskowski & Bloechl, 1993), and optimize molecular geometries. `PyQInt` is especially well suited for students and researchers who want to interact with and understand the underlying steps of electronic structure theory, offering full access to intermediate data structures and algorithmic pathways.

While numerical efficiency is not the primary goal, `PyQInt` connects to a C++ backend for integral evaluation, enabling practical computations on small molecules. The package is fully documented and tested, and is ideal for use in courses, tutorials, or prototyping new electronic structure ideas.

Statement of need

Electronic structure theory plays a foundational role in modern computational chemistry, with widespread applications in materials discovery, catalyst design, drug development, and the prediction of molecular properties. As simulation tools become increasingly powerful and accessible, they are now integral to both academic research and industrial workflows.(Gordon & Windus, 2020)

However, many students and early-career researchers engage with these tools as users—relying on established software packages—without gaining a clear understanding of the underlying theoretical models, numerical procedures, or methodological limitations. This lack of transparency can lead to misinterpretation of results, inappropriate method selection, and an underappreciation of the approximations involved in electronic structure calculations.(Hulyadi et al., 2023; Stefani & Tsaparlis, 2009)

Although the Hartree-Fock (HF) method is rarely used in isolation for practical applications, it remains a critical pedagogical foundation for understanding more advanced approaches such as Density Functional Theory (DFT) and post-Hartree-Fock correlation methods.(Szabo & Ostlund, 1996) In particular, the explicit evaluation of the exchange energy in Hartree-Fock forms the conceptual and mathematical basis for hybrid functionals

JOSE
The Journal of Open Source Education

like B3LYP(Becke, 1993; Lee et al., 1988), which are among the most widely used methods in applied quantum chemistry.(Sousa et al., 2007)

PyQInt is designed to support education and exploration in electronic structure theory through a modular and transparent implementation of Hartree–Fock methodology using Gaussian-type orbitals. In contrast to software packages that abstract away computational details, PyQInt provides access to individual steps such as integral evaluation, matrix construction, SCF procedures, and orbital manipulation. This structure makes the program suitable for instructional use as well as for prototyping and method development.

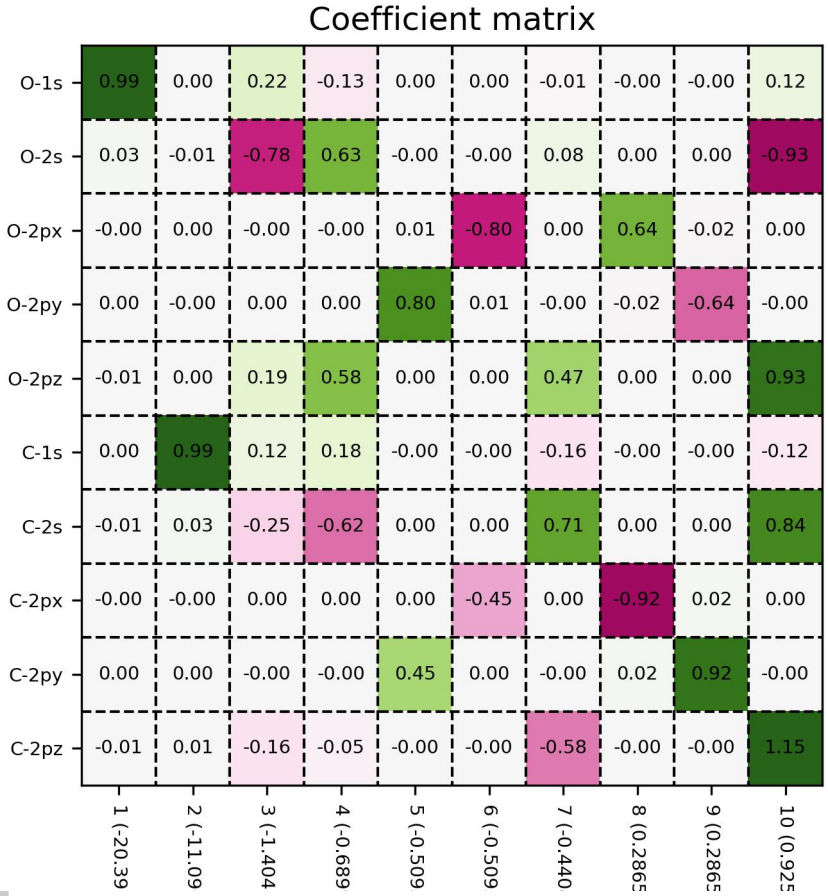

Figure 1: Visualization of the coefficient matrix from a Hartree–Fock calculation of the CO molecule, obtained using PyQInt.

Features

PyQInt is a Python-based software package developed to support instruction and exploration in electronic structure theory. It implements the Hartree–Fock method using Gaussian-type orbitals, with a particular emphasis on clarity, inspectability, and modularity. Designed as both a pedagogical tool and a platform for method prototyping, PyQInt exposes the underlying components of electronic structure calculations in a form that facilitates learning and experimentation. In addition to its modular structure, the source code is richly commented throughout, with the explicit intention that students and early-stage researchers can read and understand the implementation details. This transparency allows

users not only to use the code but also to study it as an educational resource, reinforcing theoretical concepts through hands-on engagement with working algorithms.

The package provides functionality for constructing and manipulating Gaussian basis functions, along with the evaluation of the corresponding one- and two-electron integrals. These integrals are computed using a C++ backend with OpenMP parallelization, which ensures efficient performance suitable for small to medium-sized systems. This low-level access supports detailed exploration of integral evaluation and basis set structure. In addition, PyQInt includes higher-level capabilities such as self-consistent field (SCF) Hartree–Fock calculations with DIIS acceleration, orbital localization using the Foster–Boys method(Boys, 1960), Crystal Orbital Hamilton Population (COHP) analysis (Dronskowski & Bloechl, 1993), and geometry optimization based on analytic energy gradients.

A key design feature is that all calculations return structured Python dictionary objects, which expose the internal matrices and multidimensional arrays used during computation. These include, for example, the overlap, kinetic energy, Coulomb, coefficient (see Figure 1), and Fock matrices, as well as the four-dimensional array representing the two-electron repulsion integrals. By providing this level of access, the program allows users to inspect, manipulate, and recompute quantities such as electronic energy and orbital-specific contributions using standard tools such as NumPy. This design supports a more detailed understanding of the theoretical framework and computational procedures that underpin quantum chemical models.

`PyQInt` also supports molecular orbital visualization through both two-dimensional contour plots (Figure 2) via Matplotlib(Hunter, 2007) and three-dimensional isosurface rendering (Figure 3). These features aid in connecting computational results to chemical concepts and spatial representations.

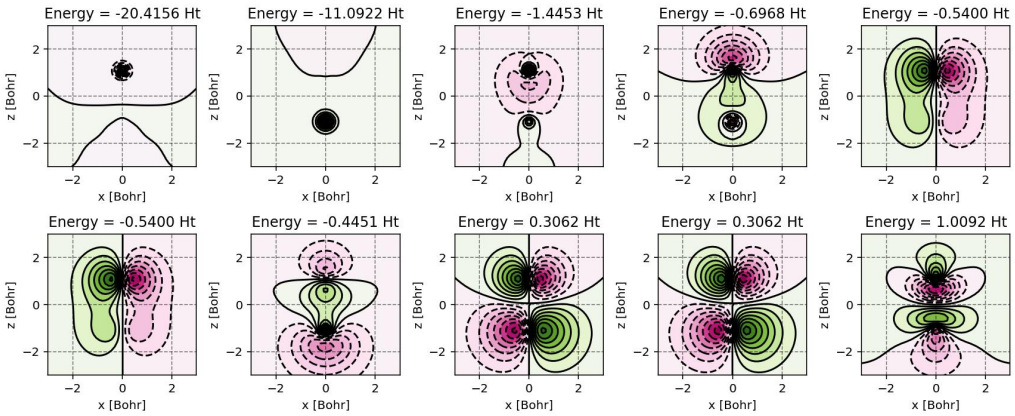

Figure 2: Two-dimensional contour plots of selected molecular orbitals of the CO molecule, visualized using `PyQInt`.

JOSE
The Journal of Open Source Education

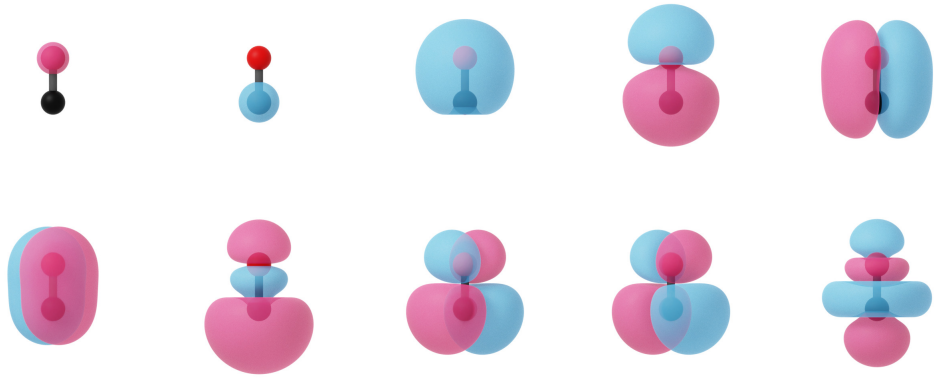

Figure 3: Three-dimensional isosurface representations of selected molecular orbitals of the CO molecule, generated with PyQInt and rendered using Blender.

All features of PyQInt are accompanied by comprehensive documentation, which includes numerous examples and explanatory materials. The documentation is designed to guide users through both basic and advanced functionality, with an emphasis on clarity and reproducibility. Many of the examples are presented as richly commented Python code snippets, illustrating typical use cases and highlighting key computational steps. This approach allows users to connect theoretical concepts with practical implementation and lowers the barrier to entry for students and early-stage researchers engaging with electronic structure theory through programming.

Use in Teaching and Curriculum

`PyQInt` is one of two computational tools used throughout the open-access textbook *Elements of Electronic Structure Theory* (Filot, 2025), which is freely available online. The textbook combines theoretical instruction with practical Python-based exercises, aiming to provide students with both a conceptual foundation and a working familiarity with electronic structure methods. `PyQInt` is introduced as a lightweight and readable implementation of Hartree–Fock theory, enabling learners to explore the mathematical and computational framework of quantum chemistry from first principles.

The software supports exercises focused on basis function construction, integral evaluation, self-consistent field procedures, and orbital analysis. These activities are intended to promote active engagement with the subject matter by encouraging students to investigate and manipulate the internal components of electronic structure calculations, rather than relying exclusively on preconfigured, black-box software. The modular architecture of `PyQInt` is consistent with the pedagogical progression adopted in the accompanying textbook, facilitating incremental learning and conceptual reinforcement through hands-on, code-based experimentation.

`PyQInt` has been used in four iterations of the course *Theoretical and Computational Chemistry* at *Eindhoven University of Technology*, where it was integrated into lectures and assignments. Student feedback collected through course evaluations indicates a high level of appreciation for the tool, with many students identifying it as instrumental in developing their understanding of electronic structure calculations. The transparency of the code and the accessibility of key computational elements have been noted as particularly valuable for clarifying how abstract theoretical concepts are translated into numerical procedures.

This integration highlights the effectiveness of `PyQInt` as an educational resource, par-

JOSE
The Journal of Open Source Education

ticularly in settings where algorithmic transparency and practical skill development are prioritized. By facilitating direct interaction with core matrices, energy terms, and orbital visualizations, the software supports deeper insight into both the theory and implementation of electronic structure methods. It is well suited for use in introductory courses, flipped classroom environments, and independent study contexts.

Availability and Deployment

In addition to its educational utility, `PyQInt` is distributed through widely used package managers, including PyPI and Anaconda, which simplifies installation and integration across a range of computing environments. This accessibility ensures that students and instructors can easily incorporate the software into classroom exercises, Jupyter Notebooks, or larger Python-based projects without the need for complex setup procedures. The ability to install `PyQInt` with a single command facilitates its use in teaching environments where consistency and ease of deployment are critical, while also making it suitable for use in virtual labs, remote instruction, and open science workflows.

References

Becke, A. D. (1993). Density-functional thermochemistry. III. The role of exact exchange. *The Journal of Chemical Physics*, *98*(7), 5648–5652. https://doi.org/10.1063/1.464913

Boys, S. F. (1960). Construction of some molecular orbitals to be approximately invariant for changes from one molecule to another. *Rev. Mod. Phys.*, *32*, 296–299. https://doi.org/10.1103/RevModPhys.32.296

Dronskowski, R., & Bloechl, P. E. (1993). Crystal orbital hamilton populations (COHP): Energy-resolved visualization of chemical bonding in solids based on density-functional calculations. *The Journal of Physical Chemistry*, *97*(33), 8617–8624. https://doi.org/10.1021/j100135a014

Filot, I. A. W. (2025). *Elements of electronic structure theory*. https://ifilot.pages.tue.nl/elements-of-electronic-structure-theory/index.html

Gordon, M. S., & Windus, T. L. (2020). Editorial: Modern architectures and their impact on electronic structure theory. *Chemical Reviews*, *120*(17), 9015–9020. https://doi.org/10.1021/acs.chemrev.0c00700

Hulyadi, H., Muhali, M., & Gargazi, G. (2023). Reducing student misconceptions through problem-based learning with a computational chemistry-assisted question map approach. *Jurnal Penelitian Pendidikan IPA*, *09*, 11207–11217. https://doi.org/10.29303/jppipa.v9i12.5936

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Lee, C., Yang, W., & Parr, R. G. (1988). Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, *37*, 785–789. https://doi.org/10.1103/PhysRevB.37.785

Pople, J. A., Gill, P. M. W., & Handy, N. C. (1995). Spin-unrestricted character of kohn-sham orbitals for open-shell systems. *International Journal of Quantum Chemistry*, *56*(4), 303–305. https://doi.org/10.1002/qua.560560414

Pulay, P. (1980). Convergence acceleration of iterative sequences. The case of scf iteration. *Chemical Physics Letters*, *73*(2), 393–398. https://doi.org/10.1016/0009-2614(80)80396-4

Roothaan, C. C. J. (1951). New developments in molecular orbital theory. *Rev. Mod. Phys.*, *23*, 69–89. https://doi.org/10.1103/RevModPhys.23.69

Sousa, S. F., Fernandes, P. A., & Ramos, M. J. (2007). General performance of density functionals. *The Journal of Physical Chemistry A*, *111*(42), 10439–10452. https://doi.org/10.1021/jp0734474

Stefani, C., & Tsaparlis, G. (2009). Students' levels of explanations, models, and misconceptions in basic quantum chemistry: A phenomenographic study. *Journal of Research in Science Teaching*, *46*, 520–536. https://doi.org/10.1002/tea.20279

Szabo, A., & Ostlund, N. S. (1996). *Modern quantum chemistry: Introduction to advanced electronic structure theory*. Dover.

Taketa, H., Huzinaga, S., & O-ohata, K. (1966). Gaussian-expansion methods for molecular integrals. *Journal of the Physical Society of Japan*, *21*(11), 2313–2324. https://doi.org/10.1143/JPSJ.21.2313