The Journal of Open Source Software

Den2Obj: A command line tool for producing isosurfaces from electron density data files

I. A. W. Filot [1¶]

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

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 11 May 2025
Published: unpublished

Summary

Electron density plays a foundational role in electronic structure calculations. It offers a detailed spatial representation of the distribution of electrons in molecules and materials. In computational methodologies, most notably Density Functional Theory (DFT), electron density serves as a central quantity from which key properties such as total energy, electrostatic potential, and atomic forces are derived. The analysis of electron density provides significant insights into chemical bonding, reactivity, and electron redistribution, especially during bond formation and dissociation. This capability enables the prediction and interpretation of the properties of materials and molecular systems.

Electron density is mathematically represented as a scalar field. It assigns a scalar value, corresponding to the probability density of finding an electron, to each point in three-dimensional space. Due to the complexity and high dimensionality of such data, specialized visualization techniques are necessary for meaningful interpretation. Common methods include contour plotting, isosurface generation, and volumetric rendering. These techniques facilitate the exploration of electronic behavior and structural characteristics.

This utility, `Den2Obj`, is a C++ based command line program developed to generate isosurfaces from electron density scalar fields. It supports input formats that are commonly used in electronic structure calculations, including CHGCAR and PARCHG files produced by VASP, as well as Gaussian Cube files. `Den2Obj` is capable of converting these formats into a native `.d2o` file format, which achieves substantial reductions in file size through efficient compression. The resulting isosurfaces can be exported in several widely supported geometry file formats, including STL (Stereolithography), PLY (Polygon File Format), and OBJ (Wavefront). These output formats enable convenient integration with a broad range of visualization and post-processing software tools.

Statement of need

Isosurface Visualization with `Den2Obj`

Isosurfaces play a central role in scientific research and engineering applications by providing an effective means of visualizing complex scalar fields and understanding intricate physical phenomena. An isosurface represents a set of points in a field where a scalar quantity, known as the *isovalue*, remains constant. Due to their importance, many software packages support isosurface generation, including `Open Data Explorer` (Thompson et al., 2004), `MATLAB` (The MathWorks Inc., 2022), `ParaView` (Ahrens et al., 2005), and `VESTA` (Momma & Izumi, 2011). These tools are primarily designed for interactive use and typically rely on graphical user interfaces.

In contrast, `Den2Obj` is a C++-based command line tool developed for automated isosurface extraction from electron density and wavefunction data. It supports input formats commonly produced by VASP (Hafner, 2008), such as CHGCAR and PARCHG, as well as Gaussian (Frisch et al., 2016) Cube files. The resulting isosurfaces can be exported in widely used 3D geometry formats, including Stereolithography (.stl), Polygon File Format (.ply), and Wavefront (.obj).

Isosurface generation in Den2Obj is carried out using either the marching cubes algorithm (Lorensen & Cline, 1987) or the marching tetrahedra algorithm (Burke, 1994), both of which can be selected via command line arguments. These algorithms are implemented with OpenMP parallelization to leverage the performance of modern multi-core CPUs. When visualizing wavefunctions rather than electron densities, it is often useful to generate separate isosurfaces for positive and negative lobes. Den2Obj accommodates this by enabling dual isosurface generation through a single command line argument.

An illustrative example is presented in Figure 1, which displays the canonical molecular orbitals of the benzene molecule. These orbitals were computed using the PyQInt program.(Filot, 2020) Isosurfaces were generated via the marching cubes algorithm as implemented in Den2Obj, producing .ply files that were subsequently imported into Blender(*Blender - a 3D Modelling and Rendering Package*, 2018), along with the atomic coordinates of benzene, for rendering.

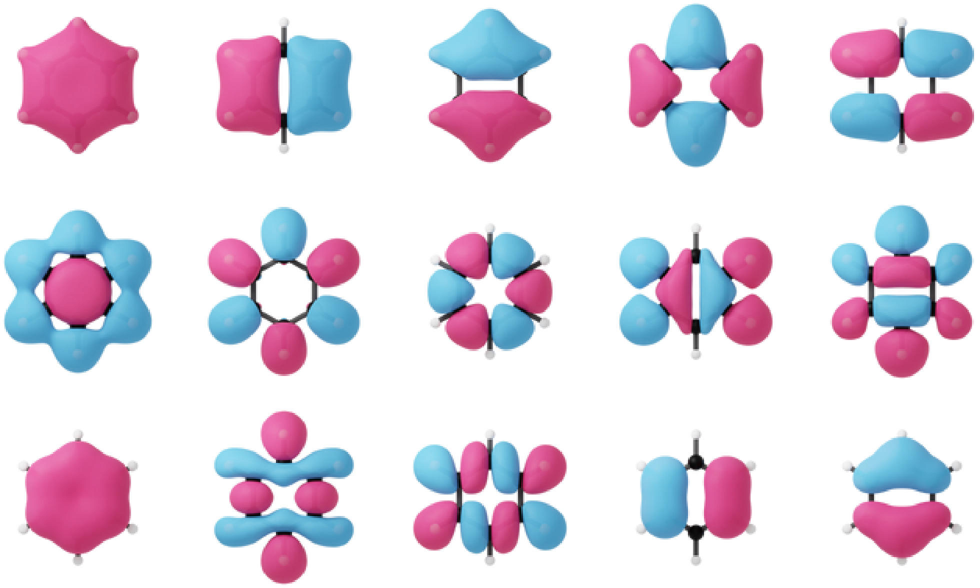

Figure 1: Isosurfaces of the first 15 canonical valence molecular orbitals of benzene.

For efficient research data management purposes, Den2Obj is also able to convert CHGCAR and PARCHG files to its own custom d2o format, which is a lossless format that stores the scalar field as a collection of floats utilizing compression. Upon conversion of input files to the native d2o file type, the program explores various compression algorithms, i.e. lzma (Pavlov, 1996), bzip2 (Wielaard et al., 1996) and gzip (Gailly & Adler, 1992), and uses the one that yields optimal results. In comparison to the original CHGCAR or PARCHG files, d2o files are able to achieve a compression ratio around 10%.

Besides building isosurfaces, Den2Obj can also produce OpenVDB (Museth, 2013) files allowing for volumetric rendering in programs such as Blender. In contrast to the rendering of isosurfaces, the main advantage of volumetric rendering is that internal details and density variations are more prominently shown, providing a comprehensive and nuanced understanding of the scalar field. In a way, volumetric rendering lies in between isosurfaces and contour plots in terms of visualizing a scalar field. An example for the molecular orbitals of benzene is provided in

Figure 2. For demonstration and testing purposes, also a scalar field generator functionality is included that can create a number of relevant scalar fields to test the algorithms on.

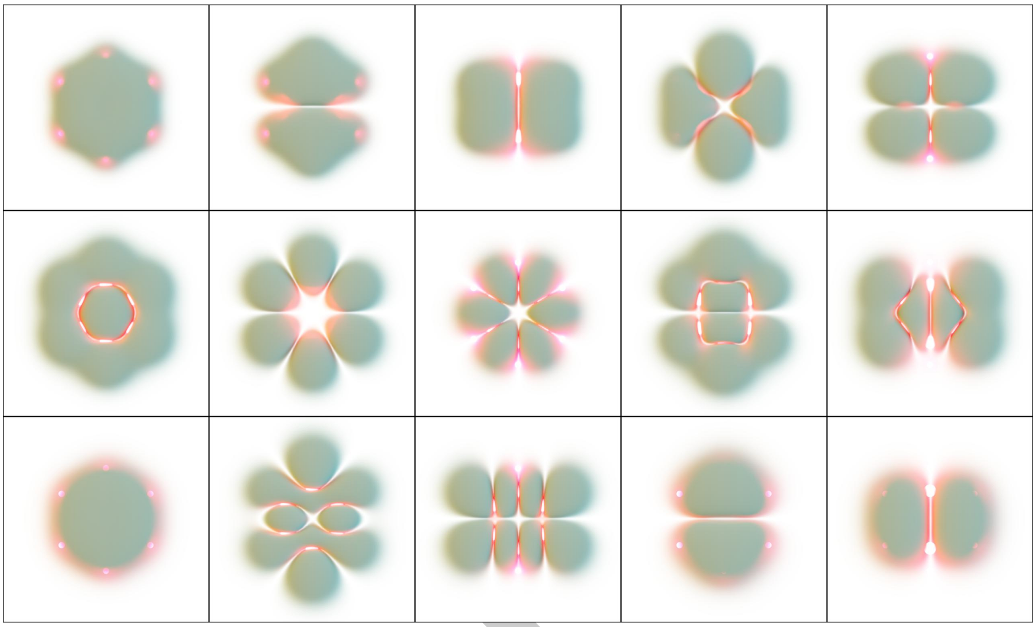

Figure 2: Volumetric rendering of the electron density associated with the first 15 canonical valence molecular orbitals of benzene using the OpenVDB format.

`Den2Obj` requires a relatively small set of dependencies, being Eigen3 (Guennebaud et al., 2010), Boost (Koranne, 2011), `TCLAP` (Smoot et al., 2009), `lzma` (Pavlov, 1996), `bzip2` (Wielaard et al., 1996) and `gzip` (Gailly & Adler, 1992). Creation of VDB files requires the presence of the OpenVDB library (Museth, 2013). The user can select during compilation whether they want to include this functionality or not. `Den2Obj` is designed to be used by researchers and students working in computational materials modelling using the quantum chemical software. It has already been used in a number of scientific publications. (Filot et al., 2016; Su et al., 2016, 2018)

An extensive user guide including examples, compilation instructions, tutorials (including a rendering tutorial in Blender) and documentation of the command-line arguments, is available at https://den2obj.imc-tue.nl/.

Acknowledgements

This work was supported by the Netherlands Center for Multiscale Catalytic Energy Conversion, and NWO Gravitation program funded by the Ministry of Education, Culture and Science of the government of the Netherlands. The Netherlands Organization for Scientific Research is acknowledged for providing access to computational resources.

References

Ahrens, J., Geveci, B., & Law, C. (2005). *ParaView: An end-user tool for large-data visualization* (C. D. Hansen & C. R. Johnson, Eds.; pp. 717–731). Butterworth-Heinemann. https://doi.org/10.1016/B978-012387582-2/50038-1

Blender - a 3D modelling and rendering package. (2018). Blender Foundation. http://www.blender.org

Burke, P. (1994). *Polygonising a scalar field*. https://paulbourke.net/geometry/polygonise/

Filot, I. A. W. (2020). *PyQInt: An educational hartree-fock code in python*. https://github.com/ifilot/pyqint

Filot, I. A. W., Fariduddin, F., Broos, R. J. P., Zijlstra, B., & Hensen, E. J. M. (2016). A quantum-chemical DFT study of CO dissociation on fe-promoted stepped rh surfaces. *Catalysis Today*, *275*, 111–118. https://doi.org/https://doi.org/10.1016/j.cattod.2015.10.009

Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Scalmani, G., Barone, V., Petersson, G. A., Nakatsuji, H., Li, X., Caricato, M., Marenich, A. V., Bloino, J., Janesko, B. G., Gomperts, R., Mennucci, B., Hratchian, H. P., Ortiz, J. V., … Fox, D. J. (2016). *Gaussian16 Revision C.01*.

Gailly, J., & Adler, M. (1992). *Gzip*. https://www.gnu.org/software/gzip/.

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

Hafner, J. (2008). Ab-initio simulations of materials using VASP: Density-functional theory and beyond. *Journal of Computational Chemistry*, *29*(13), 2044–2078. https://doi.org/10.1002/jcc.21057

Koranne, S. (2011). Boost c++ libraries. In *Handbook of open source tools* (pp. 127–143). Springer US. https://doi.org/10.1007/978-1-4419-7719-9_6

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, *21*(4), 163–169. https://doi.org/10.1145/37402.37422

Momma, K., & Izumi, F. (2011). *VESTA3* for three-dimensional visualization of crystal, volumetric and morphology data. *Journal of Applied Crystallography*, *44*(6), 1272–1276. https://doi.org/10.1107/S0021889811038970

Museth, K. (2013). VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.*, *32*(3). https://doi.org/10.1145/2487228.2487235

Pavlov, I. (1996). *Lzma*. https://www.7-zip.org/.

Smoot, M. E., Aarno, D., & others. (2009). *Templatized c++ command line parser library*. https://tclap.sourceforge.net/.

Su, Y.-Q., Filot, I. A. W., Liu, J.-X., & Hensen, E. J. M. (2018). Stable pd-doped ceria structures for CH4 activation and CO oxidation. *ACS Catalysis*, *8*(1), 75–80. https://doi.org/10.1021/acscatal.7b03295

Su, Y.-Q., Filot, I. A. W., Liu, J.-X., Tranca, I., & Hensen, E. J. M. (2016). Charge transport over the defective CeO2(111) surface. *Chemistry of Materials*, *28*(16), 5652–5658. https://doi.org/10.1021/acs.chemmater.6b01548

The MathWorks Inc. (2022). *MATLAB version: 9.13.0 (R2022b)*. The MathWorks Inc. https://www.mathworks.com

Thompson, D. L., Braun, J. A., & Ford, R. (2004). *OpenDX paths to visualization*. Visualization; Imagery Solutions Inc.

Wielaard, M., Mena, F., & Snyder, M. (1996). *bzip2*. https://sourceware.org/bzip2/.