

# Ocellaris: a DG FEM solver for free surface flows

Tormod Landet<sup>1</sup>

<sup>1</sup> Department of Mathematics, University of Oslo

DOI: [10.21105/joss.01239](https://doi.org/10.21105/joss.01239)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 06 February 2019

Published: 09 March 2019

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

## Summary

state the problem you're solving (PDE)

Free surface flows are found wherever two immiscible fluids come into contact, such as at the interface between water and air in the ocean. Simulation of high Reynolds number free surface flows are important for the design of coastal, bottom fixed, and floating structures exposed to ocean waves, as well as partially filled pipes and tanks. The high density difference between water and air poses problems for numerical approximation across the interface, and the highly non-linear behaviour of free surface, which can break and overturn, makes separating computations into two different fluid domains difficult.

Low-order finite volume methods are currently the most popular methods for solving the Navier-Stokes equations when simulating high Reynolds number free surface flows, though boundary element potential flow methods can also give reasonable results for non-breaking ocean waves and for simulation of ship-wave interaction (Kring, Huang, Sclavounos, Vada, & Braathen, 1997; Popinet, 2003; Weller, Tabor, Jasak, & Fureby, 1998). Finite volume methods can produce exactly incompressible velocity fields, which are important for the advection of the free surface, and, due to the piecewise constant discretisation, they do not suffer from Gibbs oscillations near the large jump in density and momentum at the interface between the immiscible fluids. But, there are also some downsides to using low-order methods. Increasing the approximation order is more computationally efficient than increasing the mesh density in areas where the solution is expected to be smooth, which for wave simulations is most of the domain away from the free surface. Higher order methods such as discontinuous Galerkin finite element methods (DG FEM) additionally allows more computation to be performed locally, which is beneficial for large distributed calculations (Kirby, Sherwin, & Cockburn, 2012; see e.g. Kubatko, Bunya, Dawson, Westerink, & Mirabito, 2009).

Ocellaris is an exactly incompressible Navier-Stokes solver for free surface flows with a DG FEM based numerical method that supports higher order finite elements and contains specially designed slope limiting stabilisation filters to be able to handle large density transitions (Landet, 2019; Landet, Mardal, & Mortensen, 2018). Ocellaris is implemented in Python and C++ with FEniCS (Alnæs et al., 2015; Logg, Mardal, Wells, & others, 2012) as the backend for the mesh and finite element assembly. PETSc is used for solving the resulting linear systems (Balay et al., 2019).

Ocellaris uses a YAML based input file format documented in the user guide available at [ocellaris.org](https://ocellaris.org). The mesh geometry can be defined directly in the input file for simple geometries, or it can be loaded from any file format supported by meshio (Schlömer, 2019), as long as the unstructured meshes are simplicial; triangles in 2D and tetrahedra in 3D. Due to the flexible nature of the implementation, custom numerical models can be added to a simulation by referencing external Python modules from the input file. Ocellaris uses the XDMF file format ("The XDMF format," 2019) for visualisations and a custom HDF5 format for restart files ("The HDF5 format," 2019).

⊗ citation needed

define

why?

incompressible - depends on reconstruction order

why? Python works fine as a config file format.

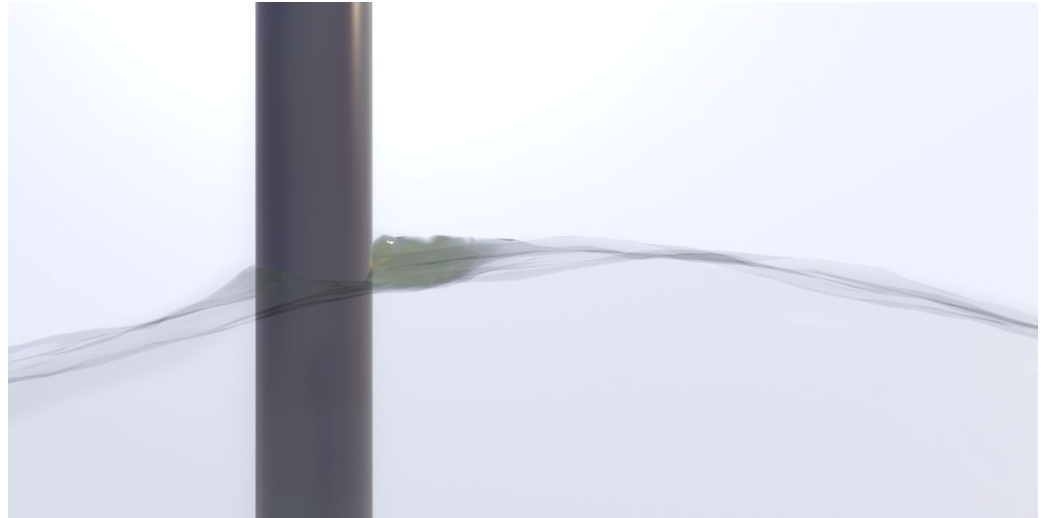
(high-order) curved geometry supported?

what does that mean? comp. intensity?

did you use the PETSc bibTeX generator?

- Impact? / Practical use?  
- Pert. claims?

- Relationship to state of the art?  
- Competitor codes?



**Figure 1:** Cylinder in waves. Rendered in Blender after using Paraview to extract the free surface from an Ocellaris simulation.

## References

- Alnæs, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., et al. (2015). The fenics project version 1.5. *Archive of Numerical Software*, 3(100). doi:[10.11588/ans.2015.100.20553](https://doi.org/10.11588/ans.2015.100.20553)
- Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., et al. (2019). PETSc: Portable, extensible toolkit for scientific computation. Retrieved from <https://www.mcs.anl.gov/petsc/>
- Kirby, R. M., Sherwin, S. J., & Cockburn, B. (2012). To CG or to HDG: A comparative study. *Journal of Scientific Computing*, 51(1), 183–212. doi:[10.1007/s10915-011-9501-7](https://doi.org/10.1007/s10915-011-9501-7)
- Kring, D., Huang, Y.-F., Sclavounos, P., Vada, T., & Braathen, A. (1997). Nonlinear ship motions and wave-induced loads by a rankine method. In *Twenty-first symposium on naval hydrodynamics* (pp. 45–63). Trondheim, Norway: The National Academies Press, Washington, DC. doi:[10.17226/5870](https://doi.org/10.17226/5870)
- Kubatko, E. J., Bunya, S., Dawson, C., Westerink, J. J., & Mirabito, C. (2009). A performance comparison of continuous and discontinuous finite element shallow water models. *Journal of Scientific Computing*, 40(1), 315–339. doi:[10.1007/s10915-009-9268-2](https://doi.org/10.1007/s10915-009-9268-2)
- Landet, T. (2019). The ocellaris project web page, user guide and blog. Retrieved from <https://www.ocellaris.org/>
- Landet, T., Mardal, K.-A., & Mortensen, M. (2018). Slope limiting the velocity field in a discontinuous galerkin divergence free two-phase flow solver. *arXiv:1803.06976 [physics]*. Retrieved from <http://arxiv.org/abs/1803.06976>
- Logg, A., Mardal, K.-A., Wells, G. N., & others. (2012). *Automated solution of differential equations by the finite element method*. Springer. doi:[10.1007/978-3-642-23099-8](https://doi.org/10.1007/978-3-642-23099-8)
- Popinet, S. (2003). Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics*, 190(2), 572–600.
- Schlömer, N. (2019). Meshio. Retrieved from <https://github.com/nschloe/meshio/>
- The HDF5 format. (2019). Retrieved from <https://www.hdfgroup.org/>
- The XDMF format. (2019). Retrieved from <http://xdmf.org/>

Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6), 620–631.