The Journal of Open Source Software

Ocellaris: a DG FEM solver for free-surface flows

Tormod Landet[1]

1 Department of Mathematics, University of Oslo

DOI: 10.21105/joss.01239

Software

- Review
- Repository
- Archive

Submitted: 06 February 2019
Published: 27 March 2019

Summary

Free-surface flows are found wherever two immiscible fluids come into contact, such as at the interface between water and air in the ocean. Simulations of free-surface flows at high Reynolds numbers 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 large difference in density between water and air poses problems for numerical approximation across the interface, and the highly non-linear behaviour of the free surface, which can break and overturn, makes separating computations into two different fluid domains difficult. As a model for free-surface flows, Ocellaris solves the variable-density incompressible Navier–Stokes equations with discontinuous density fields,

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) = \nabla\cdot\mu\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) - \nabla p + \rho\boldsymbol{g},$$
$$\nabla\cdot\boldsymbol{u} = 0,$$
$$\frac{\partial \rho}{\partial t} + \boldsymbol{u}\cdot\nabla\rho = 0.$$

Low-order finite-volume methods (FVM) are currently the most popular methods for solving the above equations when simulating free-surface flows at high Reynolds numbers[1]. Open source FVM codes include OpenFOAM (Weller, Tabor, Jasak, & Fureby, 1998) and Gerris (Popinet, 2003). Potential-flow methods are also used for simulation of non-breaking ocean waves (Tong et al., 2019), and ship-wave interaction (Faltinsen, 2005; Kring, Huang, Sclavounos, Vada, & Braathen, 1997), but these methods require that viscous effects and vorticity can be disregarded. Finite-volume methods are able to include these effects, and can produce exactly incompressible velocity fields. By this we mean that the velocity is pointwise divergence-free, the velocity facet fluxes sum to zero for each cell, and the velocity facet flux is continuous between neighbouring cells. In the mentioned FVM programs, the free-surface advection is implemented with the volume-of-fluid (VOF) method (Hirt & Nichols, 1981), which requires that the advected fluid-indicator function c is bounded, $c \in [0, 1]$. Numerical VOF advection methods rely upon divergence-free velocity fields to ensure mass conservation and bounded transport.

Due to the piecewise-constant discretisation, low-order FVM methods do not suffer from Gibbs oscillations near the large jump in density and momentum at the interface between the immiscible fluids, as long as an appropriate flux limiter is applied (Leonard, 1979). But there are 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. Implementing higher-order FVM methods is complicated on

[1] FINE/Marine, FLOW-3D, Fluent, Orca3D, SHIPFLOW XCHAP, and StarCCM+ are examples of proprietary FVM free surface flow solvers used in the industry.

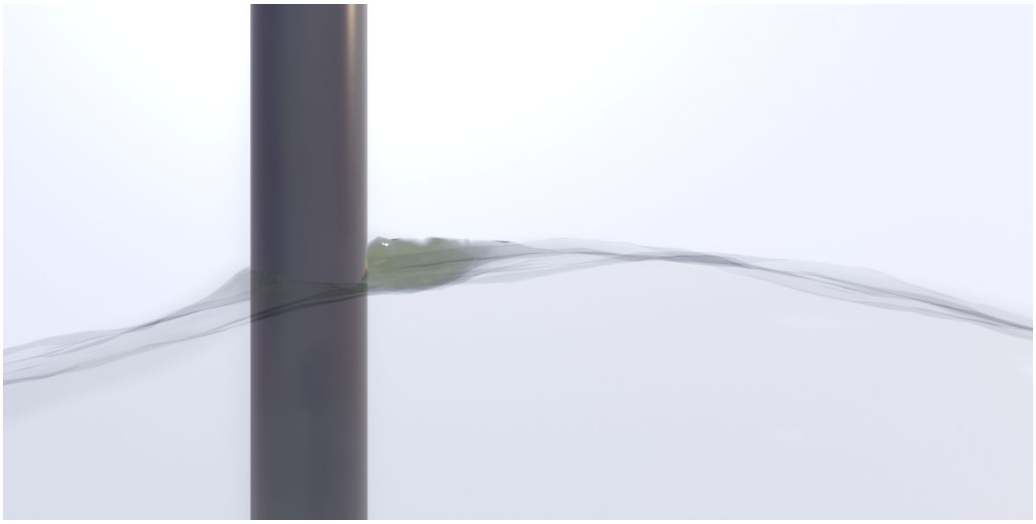

Figure 1: Cylinder in waves. Rendered in Blender after using Paraview to extract the free surface from an Ocellaris simulation.

unstructured meshes, due to the need for large reconstruction stencils in order to obtain higher-order approximations. Higher-order finite-element methods (FEM), such as the discontinuous Galerkin method (DG FEM), uses higher-order basis functions in each cell to overcome this problem. The discontinuous nature of DG FEM methods additionally allows more computation to be performed locally with less coupling of cells, which can be beneficial for the overall computational efficiency (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., 2018; Balay, Gropp, McInnes, & Smith, 1997; L. D. Dalcin, Paz, Kler, & Cosimo, 2011; Davis, 2004; "*hypre*," n.d.).

Ocellaris uses a YAML based input file format documented in the user guide available at 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).

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

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., et al. (2018). *PETSc users manual* (No. ANL-95/11 - Revision 3.9). Argonne National

Laboratory. Retrieved from http://www.mcs.anl.gov/petsc

Balay, S., Gropp, W. D., McInnes, L. C., & Smith, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, & H. P. Langtangen (Eds.), *Modern software tools in scientific computing* (pp. 163–202). Birkhäuser Press. doi:10.1007/978-1-4612-1986-6_8

Dalcin, L. D., Paz, R. R., Kler, P. A., & Cosimo, A. (2011). Parallel distributed computing using Python. *Advances in Water Resources*, *34*(9), 1124–1139. doi:10.1016/j.advwatres.2011.04.013

Davis, T. A. (2004). Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, *30*(2), 196–199. doi:10.1145/992200.992206

Faltinsen, O. M. (2005). *Hydrodynamics of High-Speed Marine Vehicles*. Cambridge University Press. doi:10.1017/CBO9780511546068

Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, *39*(1), 201–225. doi:10.1016/0021-9991(81)90145-5

hypre: High performance preconditioners. (n.d.). Lawrence Livermore National Laboratory. Retrieved from http://www.llnl.gov/CASC/hypre/

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

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

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

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

Leonard, B. P. (1979). Adjusted quadratic upstream algorithms for transient incompressible convection. In *4th Computational Fluid Dynamics Conference* (pp. 226–233). Williamsburg, VA, U.S.A: American Institute of Aeronautics; Astronautics. doi:10.2514/6.1979-1469

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

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. doi:10.1016/S0021-9991(03)00298-5

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/

Tong, C., Shao, Y., Hanssen, F.-C. W., Li, Y., Xie, B., & Lin, Z. (2019). Numerical analysis on the generation, propagation and interaction of solitary waves by a Harmonic Polynomial Cell Method. *Wave Motion*, *88*, 34–56. doi:10.1016/j.wavemoti.2019.01.007

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. doi:10.1063/1.168744