Latest news of Basilisk

Stéphane Popinet

Institut ∂'Alembert, CNRS / Sorbonne Université, Paris

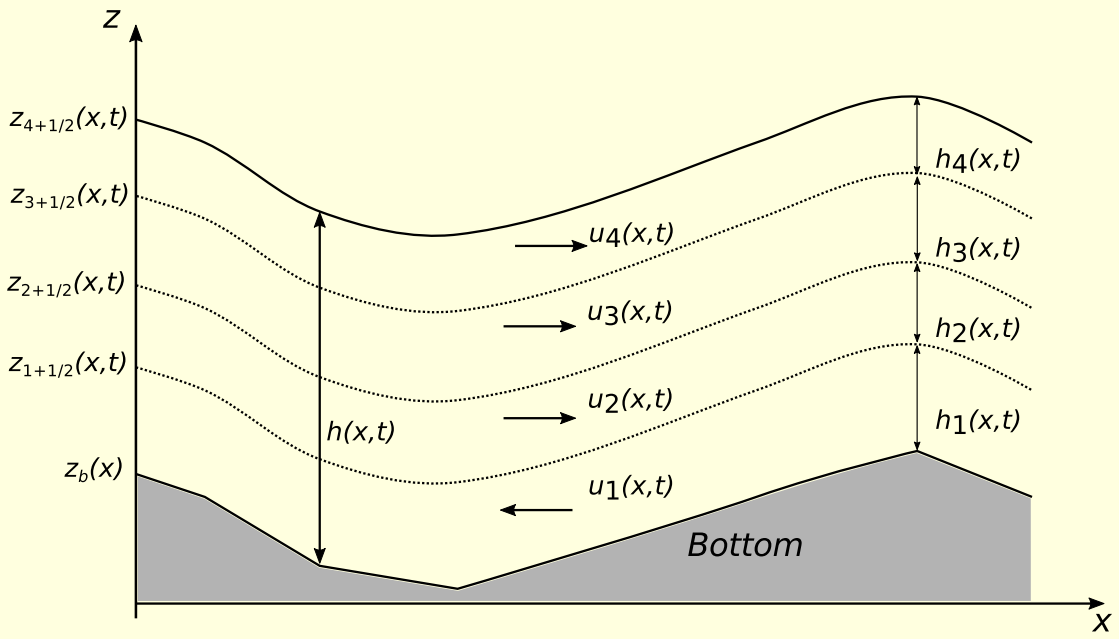
BGUM 2017

Outline

- Newish features (already there)
- To appear soon (already in sandboxes)
- Work in progress
- Plans
- Main issues / priorities
- Statistics

Newish features (2016-2017)

- Multilayer Saint-Venant solver (Francesco De Vita et al.)



Implemented as a simple extension of the single layer Saint-Venant model

`basilisk.fr/src/multilayer.h`

- Semi-implicit Saint-Venant solver (Kirstetter & Popinet)

  Relaxes the gravity wave speed CFL restriction $\Rightarrow$ low Froude number regimes can be captured much more efficiently

  Naturally well-balanced

  Extension of the all-Mach solver

  `basilisk.fr/src/saint-venant-implicit.h`

- VOF-scheme can transport associated tracers

$$\begin{aligned} \partial_t f_i + \mathbf{u}_f \cdot \nabla f_i &= 0 \\ \partial_t t_{i,j} + \nabla \cdot (\mathbf{u}_f t_{i,j}) &= 0 \\ t_{i,j} &= c_j f_i \end{aligned}$$

  `basilisk.fr/src/vof.h`

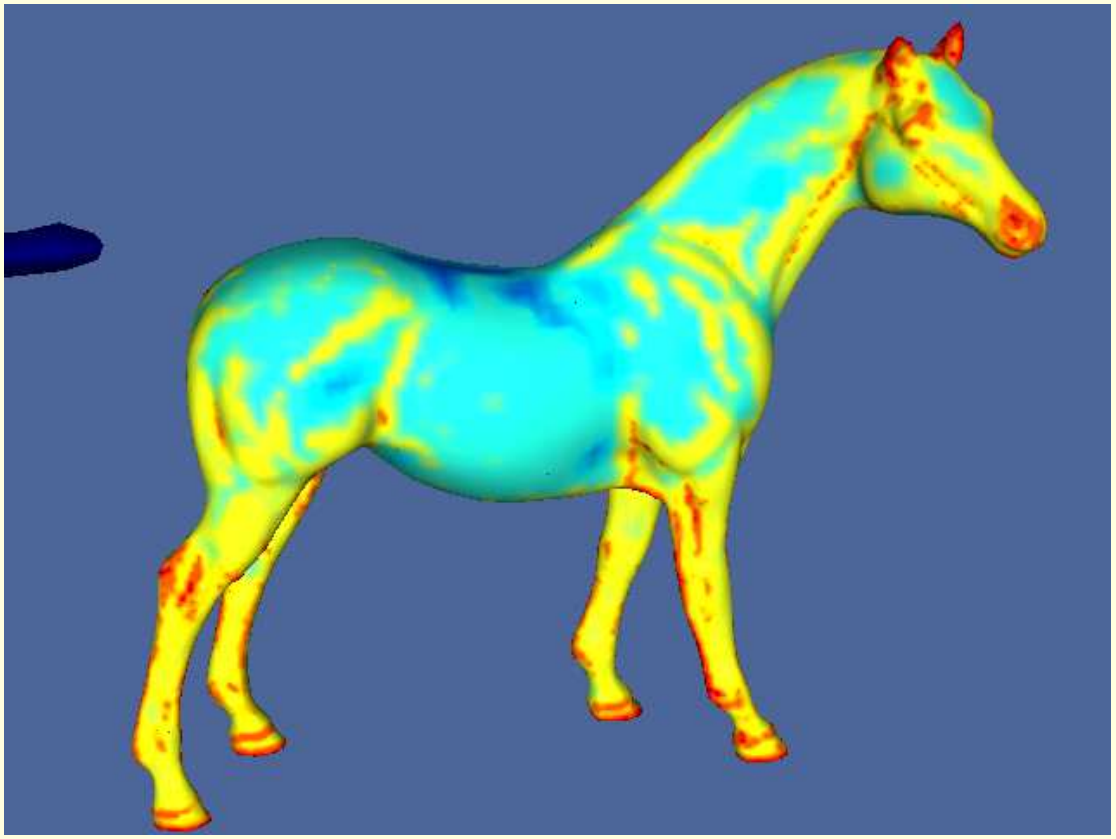
- Momentum-conserving two-phase flow scheme

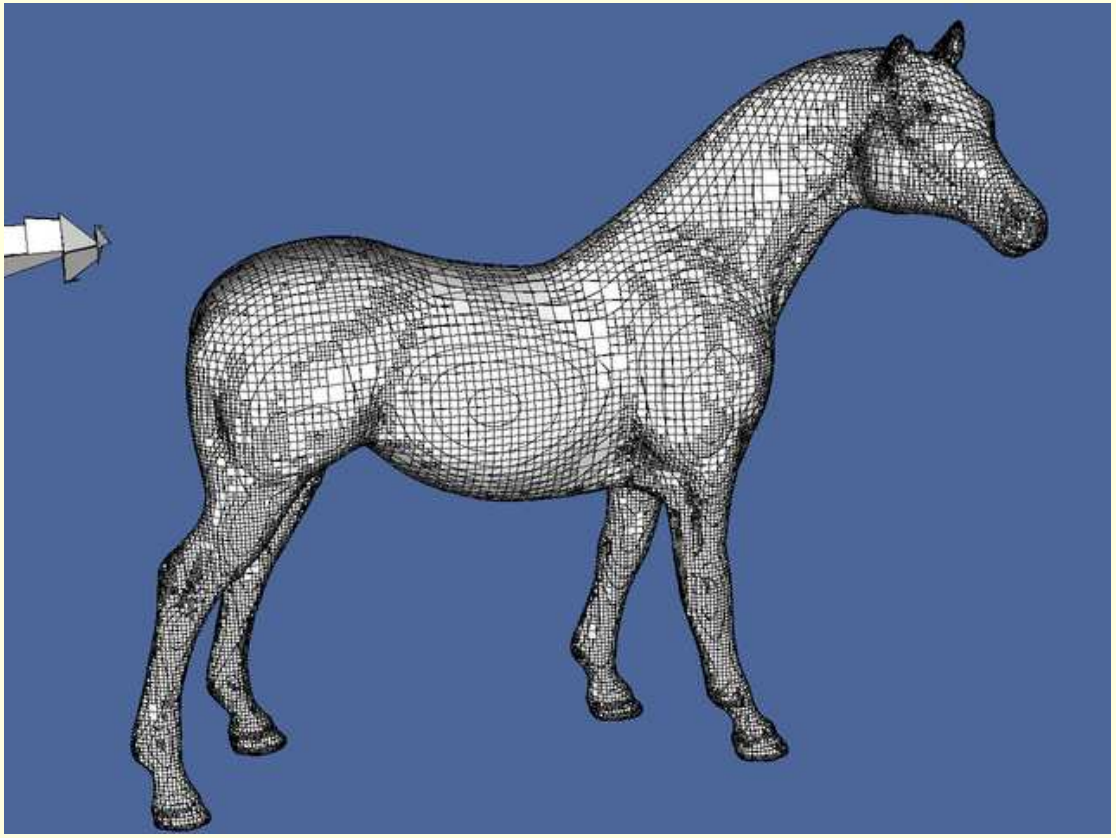
  Not conserving (yet) in the adaptive case

  `basilisk.fr/src/momentum.h`

- Construction of distance functions (i.e. levelsets) from STL files

  STL is a standard for (surface) CAD models (i.e. in industry)





Much more robust than GfsSolid in Gerris (i.e. tolerates self-intersecting surfaces, inconsistent orientations etc.)

Adaptive refinement

`basilisk.fr/src/examples/distance.c`

- Tagging of connected neighborhoods

  A simple parallel algorithm

  Used for drop distribution statistics etc.

  `basilisk.fr/src/tag.h`

- Dump/restore/restart works in parallel for multigrid or quadtree MPI

  Can also restart using different number of processors

  Uses the Z-curve linear encoding on disk to automatically load-balance at read time, on a variable number of MPI processes

- Good MPI performances on a range of large parallel systems: Infiniband clusters, Silicon Graphics shared-memory machines, IBM BlueGene

- The Basilisk web site has been split in two: src/ and the rest. This allows access using the command-line to everybody for the non src/ part.

To appear soon (already in sandboxes)

- Reduced gravity: body force $\rightarrow$ surface force

$$\begin{aligned} -\boldsymbol{\nabla} p + \rho \boldsymbol{g} &= -\boldsymbol{\nabla} p' - [\rho] \boldsymbol{g} \cdot \boldsymbol{x} \boldsymbol{n} \delta_s \\ p' &= p - \rho \boldsymbol{g} \cdot \boldsymbol{x} \end{aligned}$$

  see Annual Review of Fluid Mechanics, 2018.

  well-balanced with adaptivity

  simplifies open boundary conditions for gravity waves

- A new momentum-conserving two-phase flow scheme

  A simple extension of `navier-stokes/centered.h` rather than a derivation from the `all-mach.h` solver

- Basilisk view (`bview`) i.e. a (better!) GfsView for Basilisk
  - an (OpenGL) library to display geometric objects derived from Basilisk fields: VOF surfaces, isosurfaces, isolines, vector fields etc.

```
#include "view.h"
...
clear();
draw_vof (f);
cells();
squares (p, linear = true);
output_view (stdout);
```

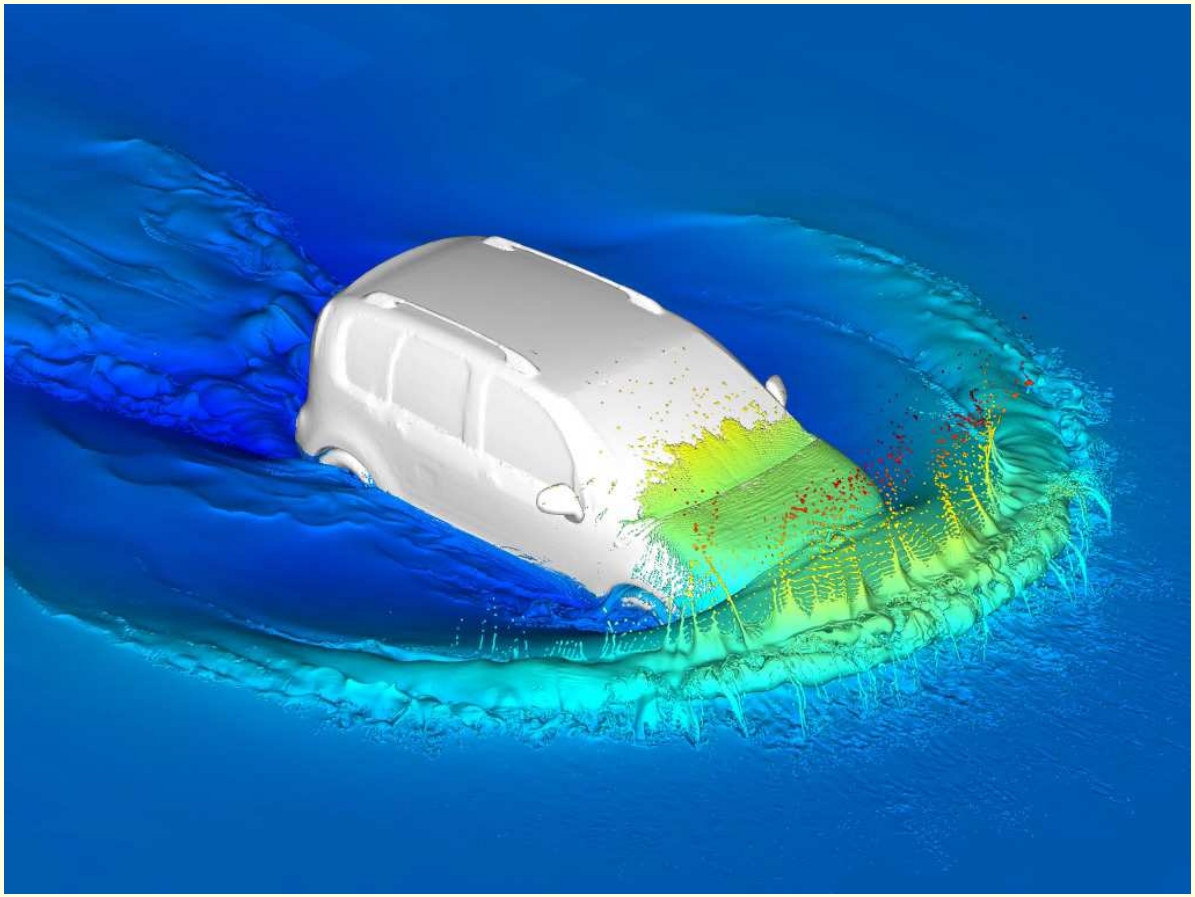
- MPI-parallel, only depends on OpenGL: either the hardware-accelerated version or OSMesa
- an interactive client/server program (similar to paraview):

  the MPI-parallel server can run on large clusters

  the client(s) send commands through unix pipes

  the commands are just library function calls (as above)
- A simple (245 lines), portable (python + tkinter) client: allows rotation, translation, zoom etc. with the mouse but no buttons, menus etc.
- bview reads "dump" file format: there will be changes in this format

- Example: new momentum-conserving scheme + STL files + simple embedded boundaries + reduced gravity + MPI + dump/restart + adaptivity + bview



12 levels, 16 million grid points (vs $2^{36} = 68 \times 10^9$), 24 hours $\sim$ 600 cores

A new version of the preprocessor (qcc)

- Almost ready since December 2016!
- Automatic boundary conditions

  Current:

```
foreach() {
  a[] = ...
  b[] = ...
}
// apply BCs to make sure stencils are consistent
boundary ({a,b});
...
foreach()
  // note that here we access the stencil for b
  // but not for a
  a[] += (b[1] - b[-1])/(2.*Delta);
```

New: BCs are automatically enforced only when necessary

```
foreach() {
  a[] = ...
  b[] = ...
} // a and b are marked as "dirty" (by qcc)
...
// qcc detects that the stencil of b is dirty
// and applies BCs
// nothing is required for a[]
foreach()
  a[] += (b[1] - b[-1])/(2.*Delta);
```

- Simplifies significantly the user interface
- The preprocessor will do a better job of tracking read/write dependencies than the user $\longrightarrow$ consistency and optimisation (particularly important for MPI)
- The preprocessor will also check illegal stencil access patterns: indices out of stencil range, illegal writes etc.

- Allows for other more complex optimisations, for example automatic overlapping communications/computations with MPI

  User-code:

```
foreach()
  a[] += (b[1] - b[-1])/(2.*Delta);
```

  Behind-the-scenes:

```
// apply non-MPI BCs (domain/refinement boundaries)
boundary_non_mpi ({b});
// start asynchronous MPI communication
boundary_mpi_start ({b});
// loop over process-local cells whose stencils do
not contain MPI ghost cells
foreach_local_stencil()
  a[] += (b[1] - b[-1])/(2.*Delta);
// wait for completion of MPI communication
```

```
boundary_mpi_wait ({b});
// update process-local cells whose stencils contain
MPI ghost cells
foreach_remote_stencil()
  a[] += (b[1] - b[-1])/(2.*Delta);
```

- Looks simple but this is not so simple to implement in practice!

  This is also limited by the "rustic" nature of qcc (which should be re-implemented using a yacc grammar).

To appear soon (following presentations)

- Higher-order schemes for hyperbolic and elliptic systems (Rajarshi)
- Multiphase compressible solvers (Daniel)
- Phase-change (Quentin)
- Viscoelastic fluids, hyperelastic solids (Jose-Maria)
- Rheologies for granular materials, Bingham fluids (Pierre-Yves)

Work in progress

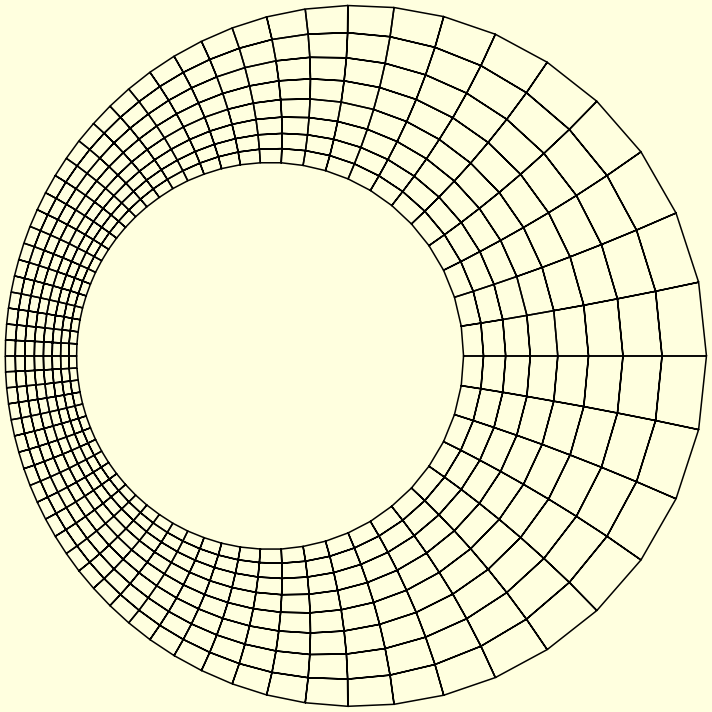
- Embedded solid boundaries: several approaches
  - Cut cells "à la Gerris"
  - `mask()` rework/redesign: solve MPI/dump incompatibilities, possibility of coupling different solvers on the same domain
  - Solid particules: 2nd-order schemes of Anthony Wachs (Can Celsuk, UBC post-doc)
- Augmented Lagrangian formulation for yield-stress fluids (A. Wachs)
- Conservative integral formulation for variable surface tension

  Levelset paper submitted to JCP (Abu-Al-Saoud, Popinet, Tchelepi)

  Marangoni flows etc.

  3D? VOF + Height Functions?

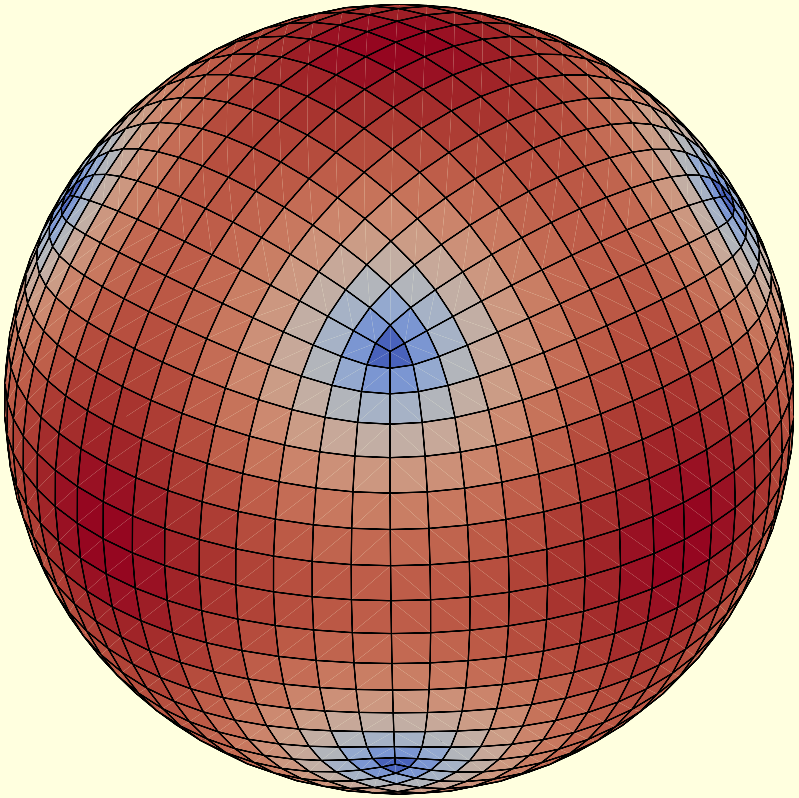
- Metric: goal: reach similar point as in Gerris i.e. Navier–Stokes (single phase) using e.g. bipolar coordinates



`http://gerris.dalembert.upmc.fr/gerris/tests/tests/wannier.html#bipolar`

Future plans

- Periodic boundary conditions and more general topologies e.g. cubed sphere (for geophysical fluid dynamics)



"Multi-boxes" but more flexible than Gerris (2:1 box connections)

- Multi-layer non-hydrostatic "generalised Saint-Venant" model:

  Saint-Venant $\longrightarrow$ multilayer Saint-Venant $\longrightarrow$ free-surface Navier–Stokes
- Conservative Serre–Green–Naghdi model (Clamond et al. 2016)
- Contact angles (using the integral formulation)
- Generalised fluid/solid solver using e.g. the "reference map" method of Kamrin & Nave (2009)

Main issues / priorities

1. Periodic boundaries and other topologies
2. "Masking" / Cut cell solids and other embedded boundary representations
3. Metric
4. Documentation / workshops / summer schools
    a) update existing docs
    b) new "developer" tutorial
5. Further development of bview (other visualisation options?)
6. Various improvements to the wiki/servers

Some statistics

197 members in basilisk-fr google group

Published papers or PhD manuscripts: `basilisk.fr/Bibliography`

2017

J. Eggers et al., “Spatial structure of shock formation", *Journal of Fluid Mechanics*

Palas Kumar Farsoiya et al., “Axisymmetric viscous interfacial oscillations - theory and simulations", *Journal of Fluid Mechanics*

Emily Lane et al., “Effects of Inundation by the 14th November, 2016 Kaikoura Tsunami on Banks Peninsula, Canterbury, New Zealand", *Pure and Applied Geophysics*

G. Legros et al., “Direct numerical simulation of an atomizing biodiesel jet: Impact of fuel properties on atomization characteristics"

Roberto Marivela-Colmenarejo, “Numerical Perspective on Tsunami Hazards and Their Mitigation by Coastal Vegetation"

An Yi et al., “Propagation of landslide induced impulse wave in channel type reservoirs”

Amir Zainali et al. “Numerical simulation of nonlinear long waves in the presence of discontinuous coastal vegetation", *Marine Geology*

Y. Zhou et al. “Experiments on, and discrete and continuum simulations of, the discharge of granular media from silos with a lateral orifice", *Journal of Fluid Mechanics*

E. Beetham et al., “Future reef growth can mitigate physical impacts of sea-level rise on atoll islands", *Earth's Future*

2016

Andres Castillo-Castellanos et al., “Reversal cycle in square Rayleigh–Bénard cells in turbulent regime", *Journal of Fluid Mechanics*

Christopher J Howland et al., “It's Harder to Splash on Soft Solids", *Physical review letters*

Geoffroy Kirstetter et al., “Modeling rain-driven overland flow: Empirical versus analytical friction terms in the shallow water approximation", *Journal of Hydrology*

2015

E. P. Beetham et al., “Wave transformation and shoreline water level on Funafuti Atoll, Tuvalu", *Journal of Geophysical Research*

S. Popinet, “A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations", *Journal of Computational Physics*

Lines of code