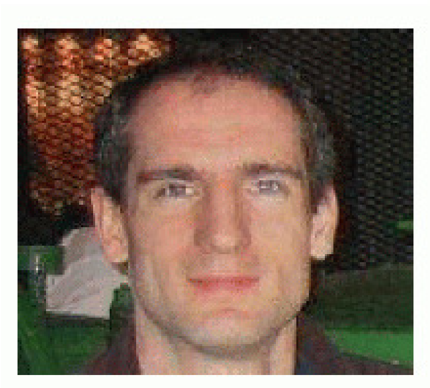
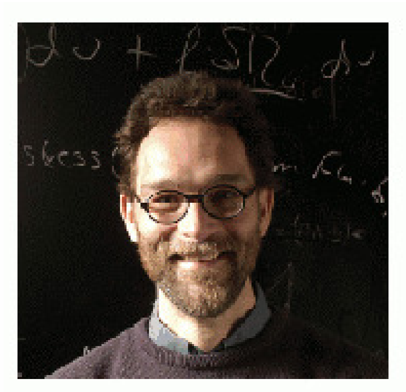
Chapter 1

People

The `oomph-lib` "architects" are (in no particular order)



Andrew Hazel



Matthias Heil

...assisted by former/current project/MSc/PhD students and collaborators who made (or are still making) significant contributions to the development of the library (listed in reverse chronological order):

- **Thierry Gonon** implemented the methodology to subtract singular (or non-singular) functions off solutions to the Poisson and Navier-Stokes equations.
- **Chris Johnson** has provided many bug fixes.
- **Puneet Matharu** works on the implementation of geometric multigrid solvers, particularly for Helmholtz equations.
- **Chihebeddine Hammami** worked on the implementation of Yulii Shikhmurzaev's interface formation theory.
- **Narjes Akriche** worked on pseudo-resonances in acoustic fluid-structure interaction problems.
- **Aman Rajvardhan** worked on implementing surfactant transport equations in two- and three-dimensional geometries.
- **Jordan Rosso** worked on topological fluid mechanics of the the Karman vortex street.
- **Florian Molinier** did some early work on the coupled solution of the axisymmetric free-surface Navier-Stokes equations and the axisymmetric Foeppl von Karman equations.

- **Jonathan Deakin** worked on a glaciology-related melt problem (and has since returned as PhD student to work on the numerical solution of acoustic fluid-structure interaction problems).
- **Draga Pihler-Puzovic** worked on the the coupled solution of the Foeppl von Karman equations and the Reynolds lubrication equation to model wrinkling/fingering in elastic-walled Hele-Shaw cells.
- **Joris Ferrand** worked on the solution of the Foeppl von Karman equations.
- **Harsh Ranjan** worked on multiple solutions of Navier–Stokes flows in curved tubes.
- **Anton Martinsson** implemented the machinery required to output `oomph-lib` data in paraview format, bypassing the need for running the time-consuming tecplot to paraview conversion scripts. He also implemented the displacement-based axisymmetric Foeppl von Karman equations.
- **André Von Borries** is working on free-surface Navier–Stokes and lubrication theory problems.
- **Matthew Walker** implemented PML methods for the azimuthally Fourier-decomposed Helmholtz equations.
- **Joris Ferrand** implemented the axisymmetric Foeppl von Karman equations.
- **Philippe Mesnard** worked on acoustic FSI problems and introduced many improvements to `oomph-lib's` machinery for handling such problems.
- **Florian Molinier** worked on the coupling of the free surface Navier-Stokes equations and the axisymmetric Foeppl von Karman equations (in the context of simulating flows in elastic-walled Hele-Shaw problems).
- **David Nigro** developed and implemented much of the machinery for acoustic fluid-structure interaction problems.
- **Matthew Russell** implemented the Foeppl-von-Karman equations; he now continues to work on poro-elastic FSI problems.
- **Raphael Perillat** worked on the simulation of flows in elastic-walled Hele-Shaw cells.
- **Robert Harter** works on acoustic fluid-structure interaction problems.
- **Radu Cimpeanu** implemented the PML boundary conditions for the Helmholtz equations and the time-harmonic equations of linear elasticity.
- **Julio Perez Sansalvador** works on parallel unstructured mesh adaptation.
- **David Shepherd** works on the numerical solution of micromagnetic problems.
- **Ray White** is working on block preconditioners.
- **Nico Bergemann** made (and continues to make) significant contributions to the adaptive unstructured mesh (re-)generation capabilities for free-surface problems.
- **Ben Saxby** works on hp adaptivity and XFEM.
- **Michael Crabb** worked on Discontinuous Galerkin (DG) methods.
- **Peter Ashcroft** worked on eigenvalue problems.
- **Jeremy van Chu** contributed to the completion the tecplot to paraview conversion scripts and significantly extended the `the paraview tutorial.` He also developed the `LineVisualiser` machinery (which allows the extraction of computational data along lines in a higher-dimensional domain) and wrote the domain-based tube mesh.
- **Guilherme Rocha** developed elements to simulate Hele-Shaw problems (by solving the free-surface Reynolds lubrication equations).
- **Ahmed Wassfi** extended **Tarak Kharrat's** work on the Helmholtz equation and implemented the Fourier-decomposed version of this equation.
- **Alexandre Raczynski** keeps providing bug fixes and contributed to the completion the tecplot to paraview conversion scripts discussed in the `the paraview tutorial.`

- **David Rutter** wrote the `tutorial for the linear elasticity equations.`
- **Tarak Kharrat** implemented the Helmholtz elements and the methodology to apply the Sommerfeld radiation condition.
- **Luigi Collucci** continued Benjamin Metz's work and developed the interface from `oomph-lib` to `Triangle`.
- **Francisco Jose Blanco Rodriguez** worked on free-surface problems and wrote the driver code that simulates the Rayleigh instability of an axisymmetric jet.
- **Wassamon Phusakulkajorn** worked on C1-continuous triangular finite elements for shell, beam and biharmonic problems.
- **Benjamin Metz** worked on adaptivity and solution transfer for unstructured meshes.
- **Amine Massit** worked on outflow boundary conditions for Navier-Stokes problems and physiological FSI problems based on meshes generated by `vmtk`.
- **Patrick Hurley** works on free surface Navier-Stokes problems.
- **Andy Gait** worked on parallelisation, in particular the problem distribution and the subsequent distributed mesh adaptation.
- **Angelo Simone** wrote python scripts that convert `oomph-lib's` output to the vtu format that can be read by `paraview`; see `the paraview tutorial` for details.
- **Sophie Kershaw** worked on the Navier-Stokes equations in spherical coordinates.
- **Floraine Cordier** developed the driver codes and tutorials for the `flow past the elastic leaflet` and `Turek & Hron's FSI benchmark`. In the process she significantly extended `oomph-lib's` FSI capabilities.
- **Stefan Kollmannsberger** and his students Iason Papaioannou and Orkun Oezkan Doenmez developed early versions of the code for `Turek & Hron's FSI benchmark` and its `non-FSI counterpart.`
- **Cedric Ody** developed the `YoungLaplace` elements and their refineable counterparts to study capillary statics problems.
- **Alice Gaertig** developed interfaces to the third-party mesh generators `Triangle,` `TetGen,` , `Geompack++,` and `CQMesh.`
- **Claire Blancon** developed the demo drivers for the collapsible channel problem (with and without fluid-structure interaction).
- **Nick Chapman** worked on the implementation of triangular and tet-elements.
- **Chris Gold** implemented explicit timestepping schemes.
- **Phil Haines** worked on bifurcation detection and tracking for the Navier-Stokes equations and developed the formulation of the equations in plane polar coordinates.
- **Richard Muddle** worked on the block preconditioning techniques for the biharmonic (and many other) equations, and parallel solvers.
- **Glyn Rees** worked on iterative linear solvers and multigrid
- **Alberto de Lozar** worked on 3D free-surface Navier-Stokes problems.
- **Jonathan Boyle** developed the initial interfaces to third-party iterative solvers and is now involved in the further parallelisation of the code and the implementation and application of block-preconditioning techniques for Navier-Stokes and fluid-structure interaction problems.
- **Renaud Schleck** completed the octree-based mesh refinement procedures and wrote the MPI-based parallel assembly routines and the interfaces to SuperLU_dist.

- **Sharaf Al-Sharif** provided the initial implementation of nodal spectral elements.
- **Daniel Meyer** used oomph-lib to study a variety of axisymmetric Navier-Stokes problems, with and without free surfaces, and developed drafts for many of our tutorials.
- **Alexandre Klimowicz** worked on block-preconditioning methods.
- **Jean-Michel Lenoir** implemented the first part of the octree-based 3D mesh refinement procedures.
- **Gemma Barson** provided the initial implementation for the 2D Delaunay mesh generation procedures.

We're always looking for more help! Get in touch if you're interested in joining the team.

1.1 PDF file

A `pdf version` of this document is available.