Imperial College London

A (very) brief introduction to numerical simulation & Fluidity

Matthew Piggott
m.d.piggott@imperial.ac.uk

Fluidity training workshop 11 – 13 Nov 2015

This presentation

- A highly cut-down version of 2012's presentation:
  http://amcg.ese.ic.ac.uk/~piggt01/Piggott-FluidityTraining-061112.pdf

- Please refer to the 2012 document for:
  - further examples of Fluidity applications
  - brief comments on some of the things to consider when conducting numerical simulations (of any type – not Fluidity-specific)
  - issues you need to think about
  - things you need to be wary of
  - i.e. modelling **'best practice'**

- and please feel free to ask or email me questions on any of these

What is Fluidity?

- A whole bunch of Fortran, C/C++, Python
- … linked to a number of external libraries, e.g. PETSc
- … to solve a variety of PDE systems
- … using control volume / finite element (CV/FEM) discretisation methods
- … on structured and unstructured (and possibly anisotropically adapting) meshes
- … in parallel using MPI and OpenMP
- … coupled to external models/ codes as well as internal physics/ chemistry/ biology/ sediment/ turbulence modules

- NB. remember that a fixed structured mesh is a sub-case of an adapting unstructured mesh, but our implementation of CV/FEM methods always assumes our meshes are unstructured – this has implications on coding and CPU costs

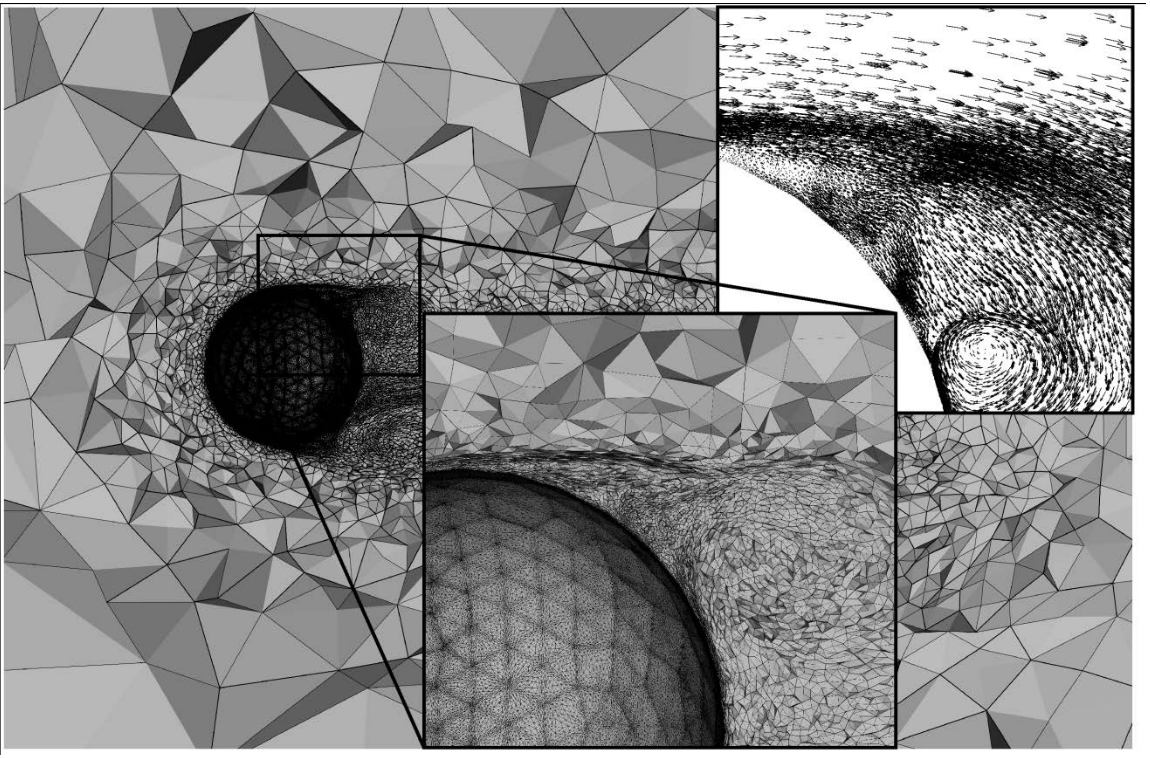
- NB. In our future roadmap we aim to make more and more use of automatic code generation techniques and increased levels of abstraction within the Fluidity framework (cf. the Firedrake project: http://www.firedrakeproject.org/)

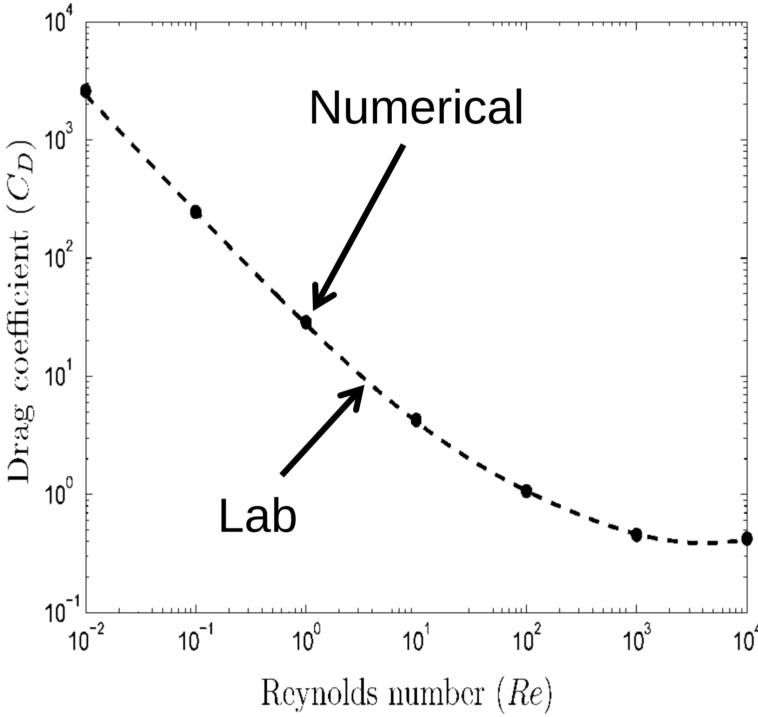
Model verification

is absolutely key and we have put a lot of effort into this.

Please refer to 2012's material for more on our **buildbot** system, test cases, and in particular the very valuable **method of manufactured solutions**.

CFD **verification/validation**: Comparisons between drag calculation in flow past a sphere at a range of Re numbers





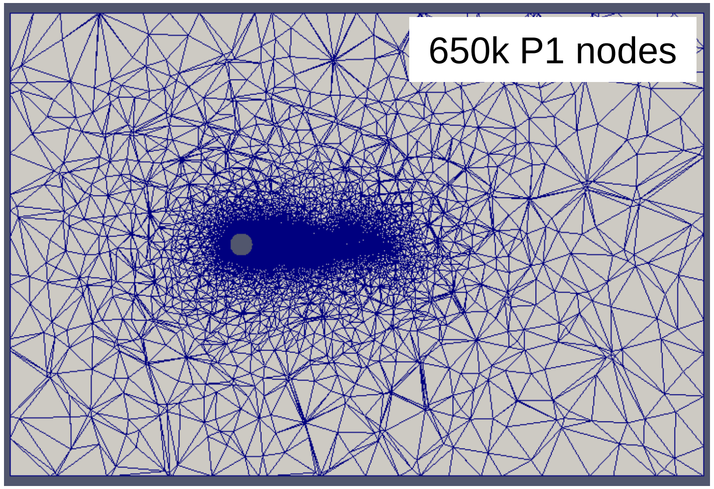
Computed drag coefficient compared against correlation (from Brown and Lawler, 2003) with lab data:

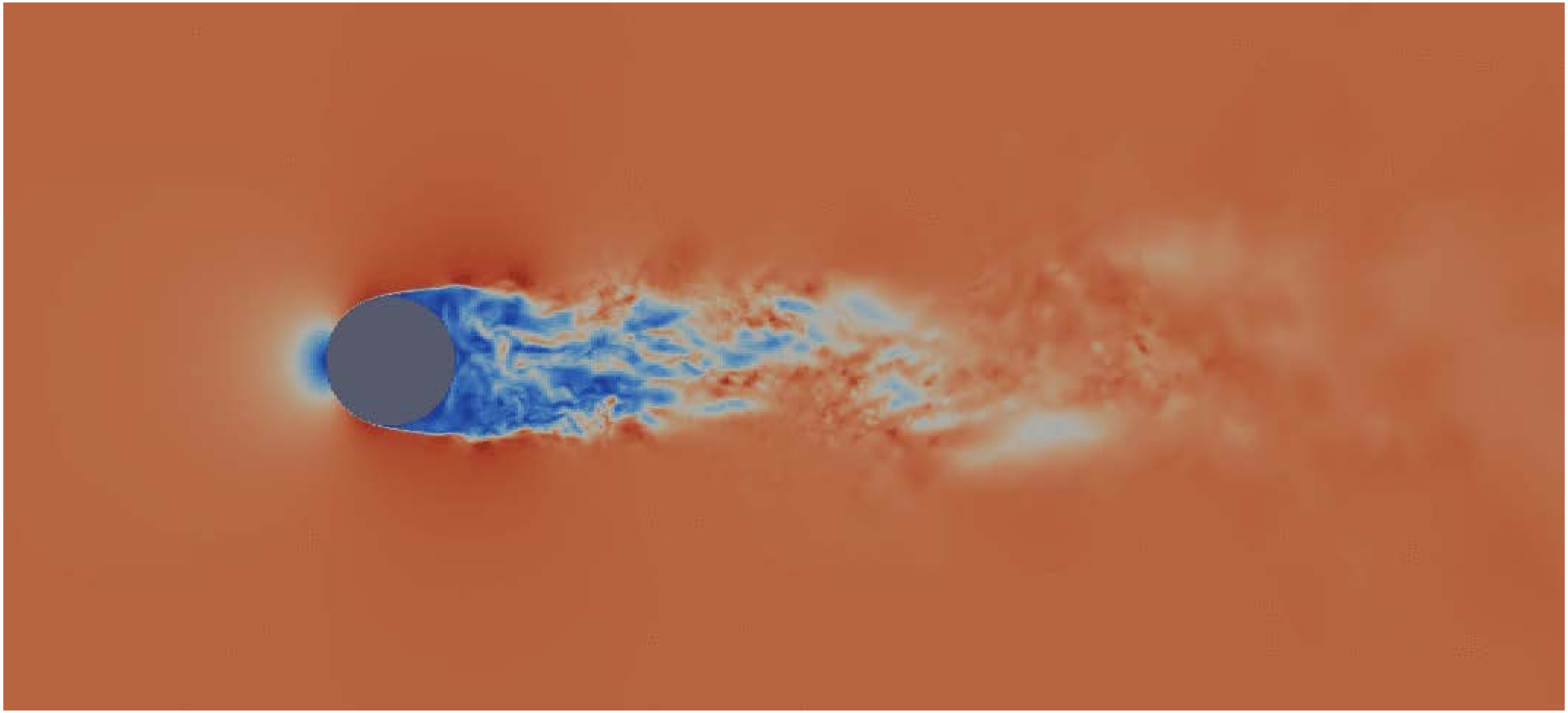
$$C_D = \frac{24}{Re}\left(1 + 0.15 Re^{0.681}\right) + \frac{0.407}{1 + \frac{8710}{Re}}$$

cf. ‘flow past sphere’ examples in manual/code

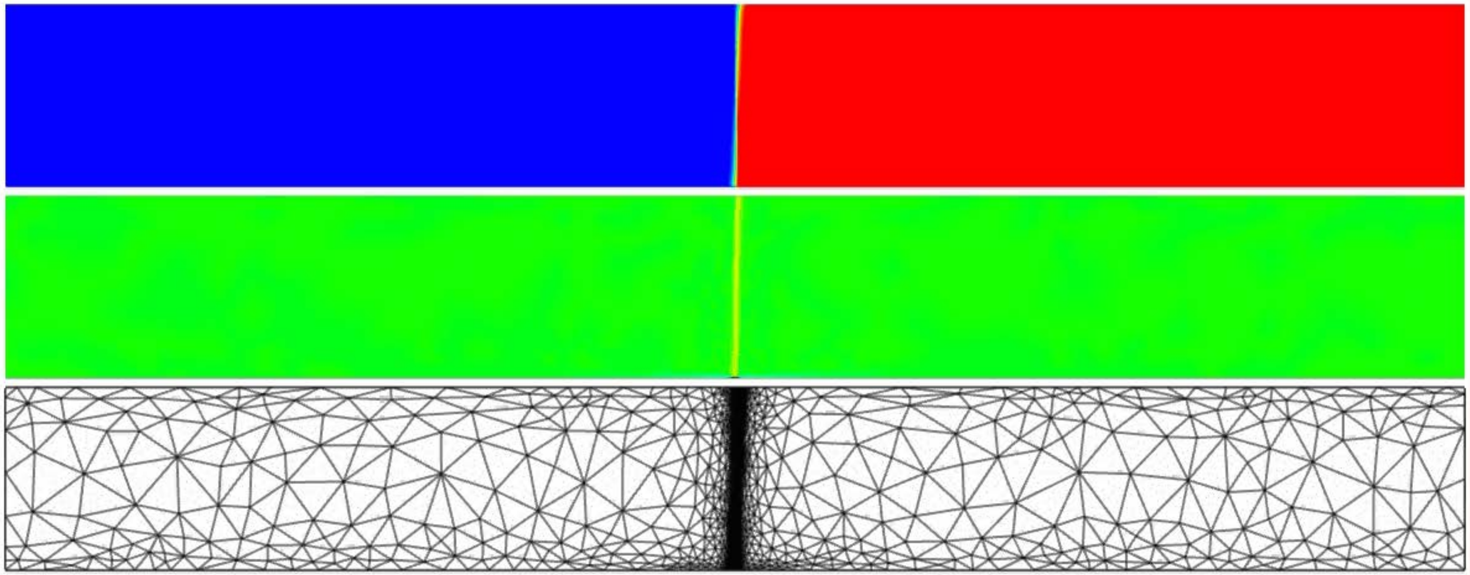
A movie showing the magnitude of velocity in a slice through the 3D domain

Notice that the problem is using an adaptive mesh and with the settings used the wake becomes under-resolved downstream of the sphere – we don't really care for this drag calculation problem and have chosen an error metric to guide adaptivity appropriately

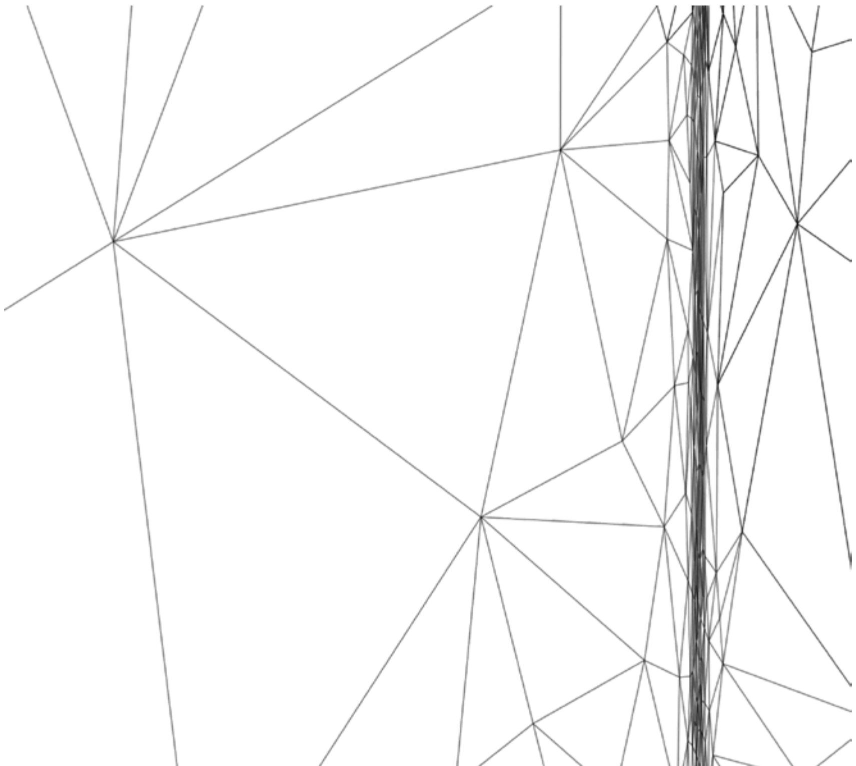




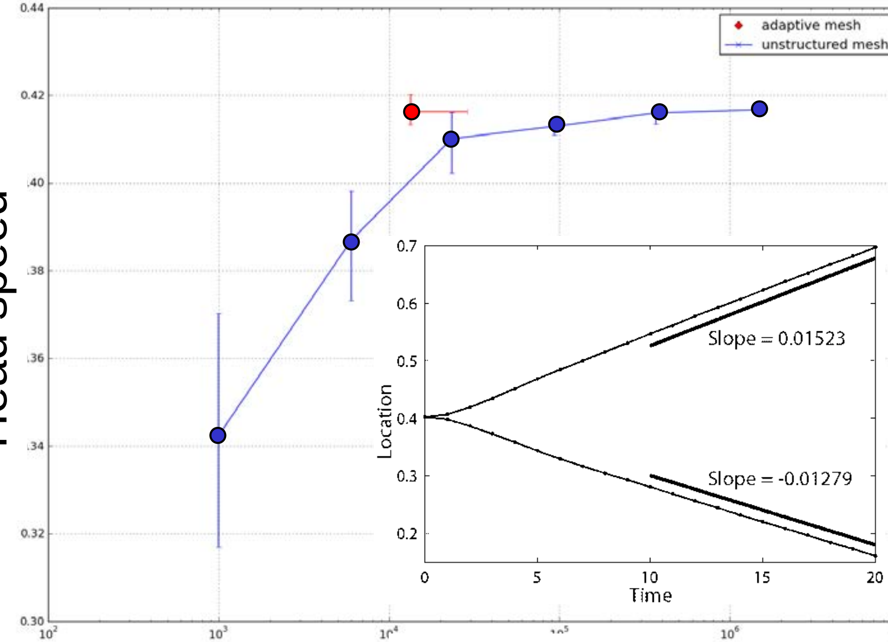
GFD **verification/validation**: 2D example with buoyancy… head speed



cf. 'lock exchange' example in manual/code



Head speed



10^3 Nodes 10^6

Hiester HR, Piggott MD, Allison PA, The impact of mesh adaptivity on the gravity current front speed in a two-dimensional lock-exchange, Ocean Modelling 38, 1-21, 2011.

Realistic domains: multi-scale tides, fixed but highly variable mesh

Final comments

- Fluidity has a lot of features, is very flexible and powerful
- This gives a lot of scope for things going wrong if you’re not careful!
- Always start simply…
- e.g. only turn on mesh adaptivity if your problem really needs it, and after you are sure from fixed mesh simulations that your problem is set up correctly and robustly
- Always check for sensitivity of your results to mesh resolution, time step, and even some numerical discretisation options … a single simulation really doesn’t tell you very much, it’s possible to be right for the wrong reasons

Wednesday

- 10:30: Welcome
- 10:30-11:00: Introduction to Fluidity
- 11:00-12:00: Finite element theory
- 12:00-13:00: *Lunch*
- 13:00-14:00: Intro to adaptivity
- 14:00-14:15: *Break*
- 14:15-15:15: Numerical considerations I
- 15:15-15:30: *Break*
- 15:30-16:30: Obtaining and compiling Fluidity

Thursday

- 10:00-11:00: Numerical considerations II
- 11:00-12:00: Meshing
- 12:00-13:00: *Lunch*
- 13:00-14:00: Diamond
- 14:00-14:15: *Break*
- 14:15-15:15: Running and visualisation
- 15:15-15:30: *Break*
- 15:30-16:30: Parallel simulations and clusters

Friday

- 10:00-10:30: What to do when your simulation "breaks“
- 10:30-11:30: Automatic code generation
- 11:30-12:00: Introduction to the examples
- 12:00-13:00: *Lunch*
- 13:00-16:00: Hands-on session