Imperial College London

Using Diamond

Christian Jacobs

Context

- To run a simulation in Fluidity, we need to give it some **input**. For example:
  - The path to the mesh file
  - What fields we want to solve for
  - Initial conditions, boundary conditions
  - What spatial and temporal discretisations will be used
  - Solver settings
- All of these need to be **specified by the user in a file**, which is then given to Fluidity.
- This is where **Diamond** comes in...

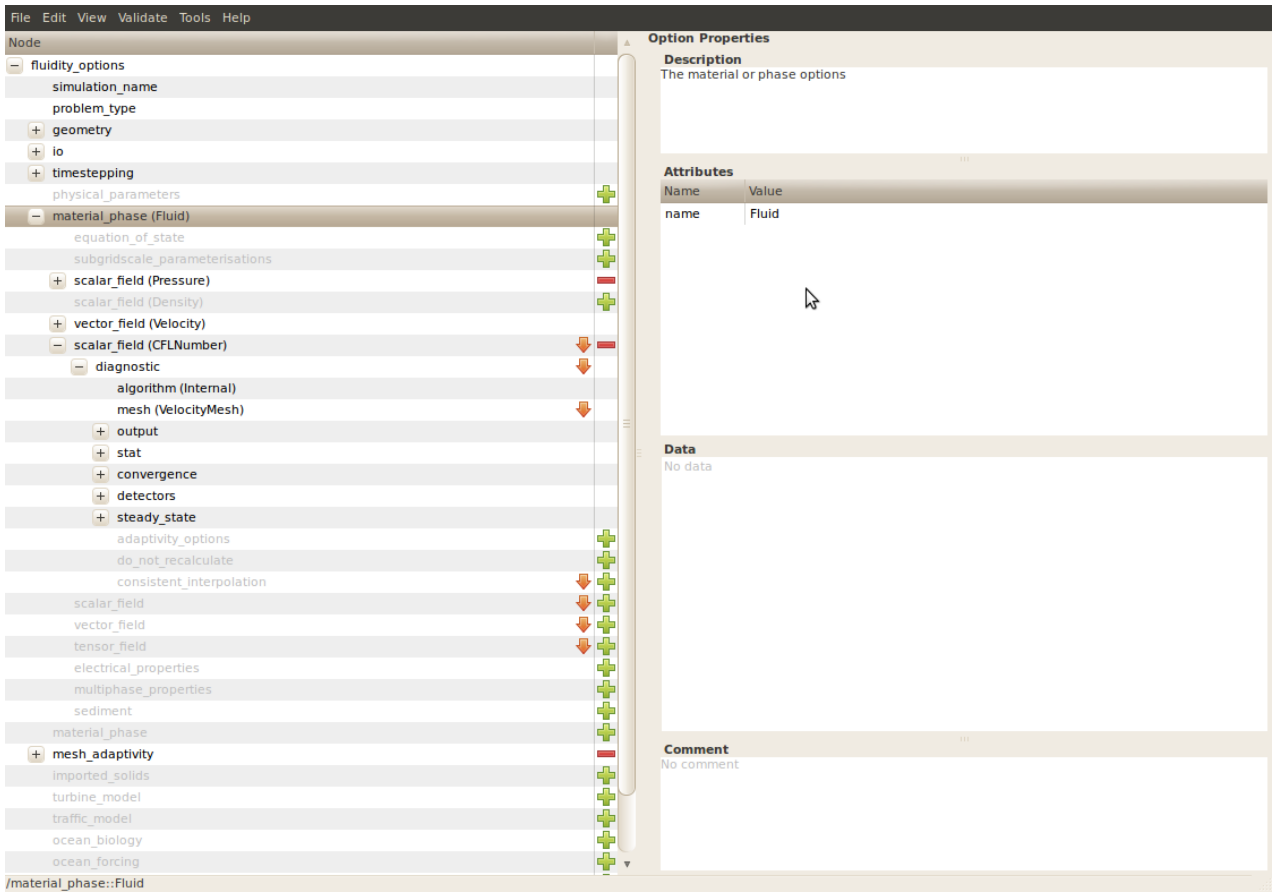
Diamond and FLML

- Diamond is an XML editor, used to **create/edit simulation configuration files**.
- These files have a ".flml" (FLuidity Markup Language) file extension, but are basically XML files with elements pre-defined...
- ...in another XML file called a **schema**. Schemas contain all the available options that the user can choose from, and act like a **blueprint** or **template** from which .flml files can be derived.

- Diamond loads the schema and gives you all the options contained within.

Imperial College London

Diamond's User Interface

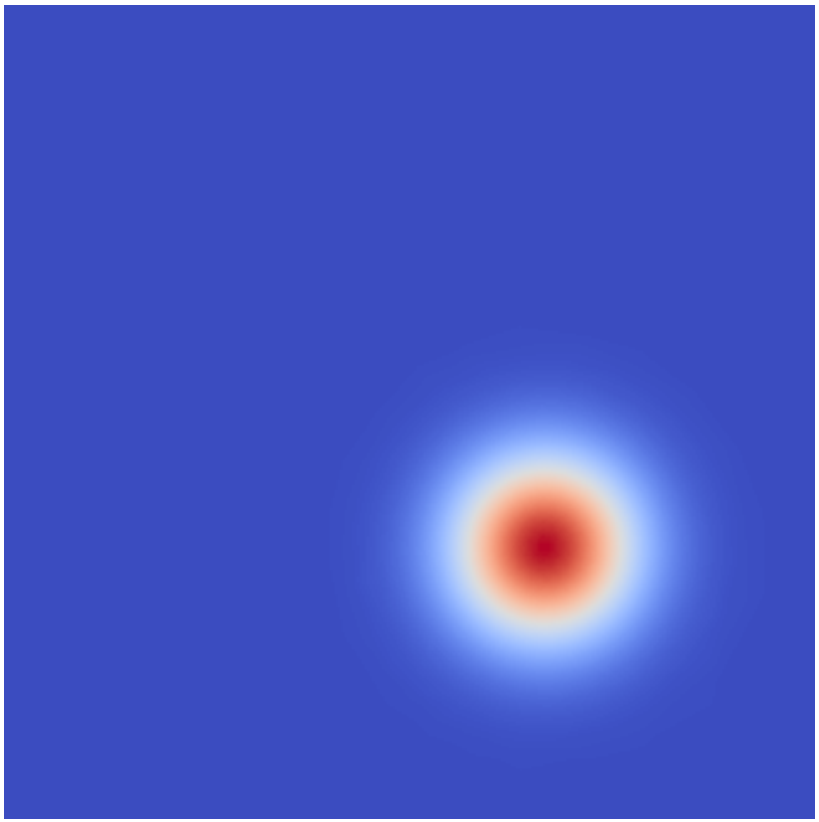


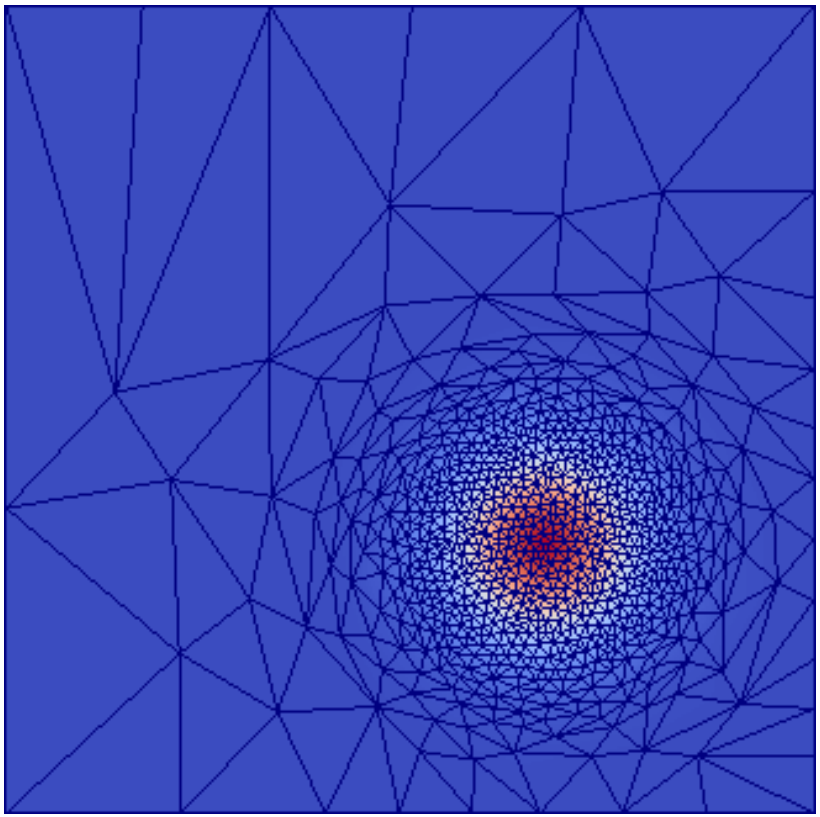
Ham et al., 2010

Live demo

- Stommel gyre
- Prescribed velocity
- Adaptive mesh
- Advect a tracer (temperature) and measure mixing

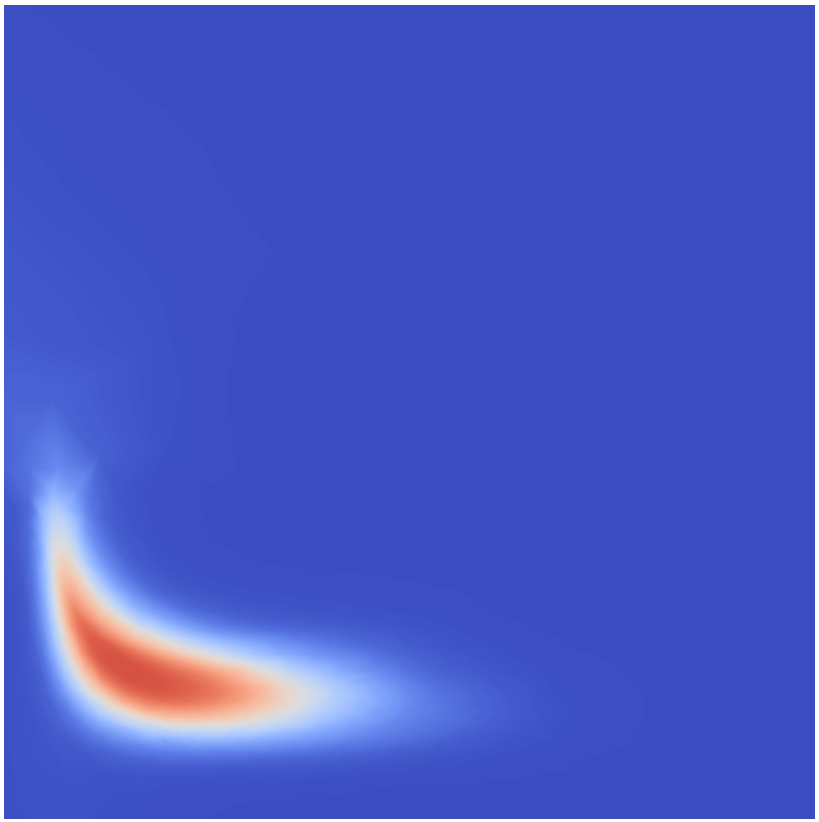
Output

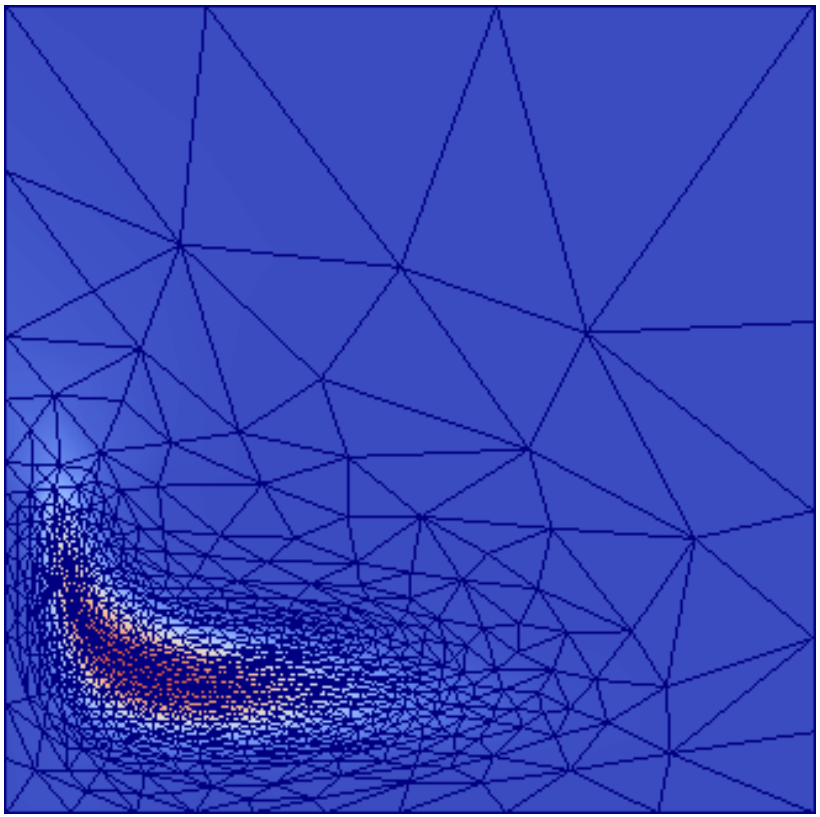




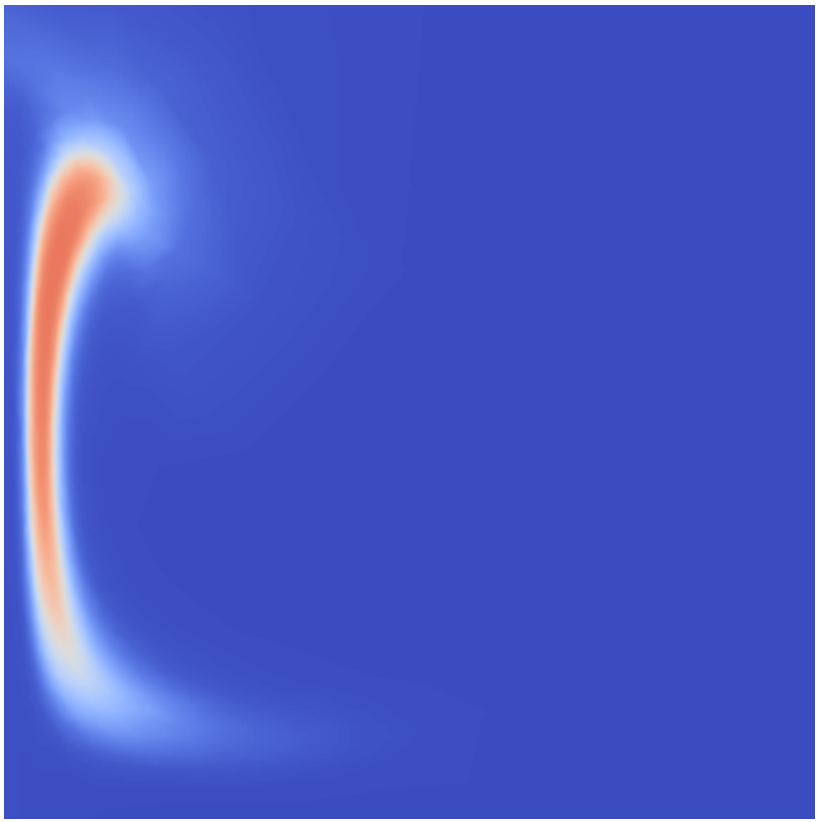
Imperial College London

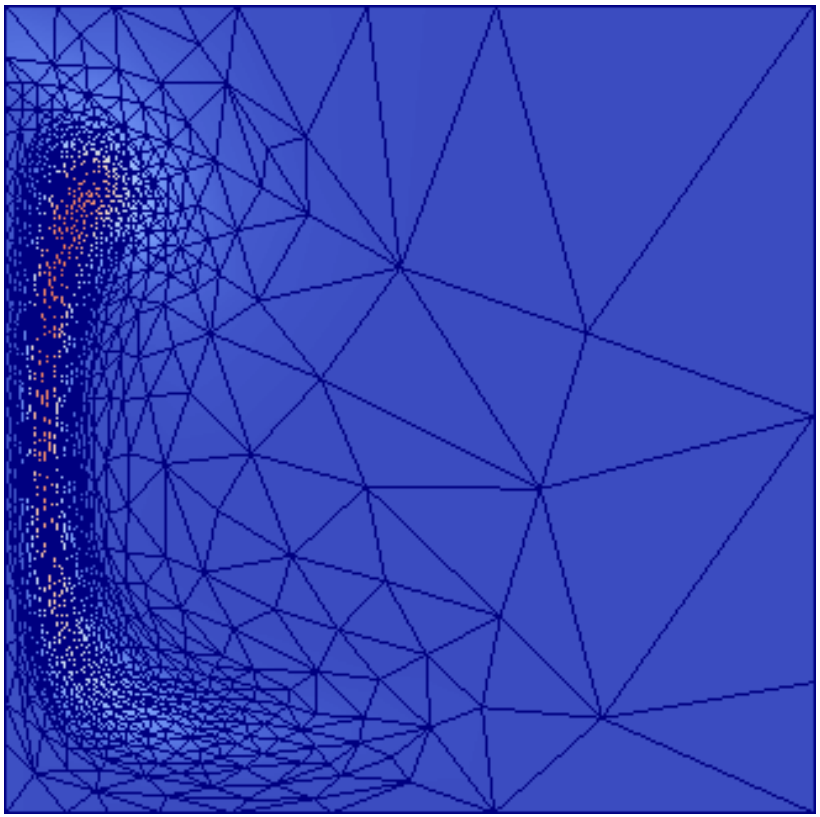
Output



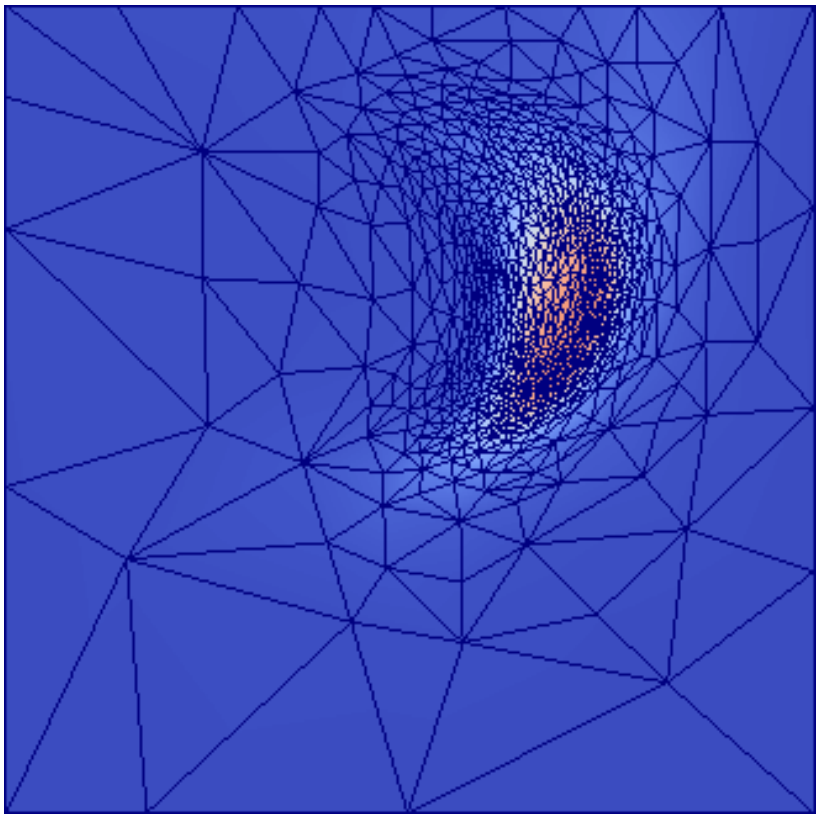


Output





Output



Before we start...

Make a directory in your data/<username> directory:

```
mkdir stommel
cd stommel
cp /scratch/Stommel2_adapt.* .
cp /scratch/stommel.pvsm .
cp /scratch/Stommel_function.py .
```

Imperial College London

Create a FLML

Live demo

```
diamond -s
/data/<username>/fluidity/schema/fluidity_options.rng
my.flml
```

Imperial College London

Running Fluidity

```
/path/bin/fluidity my.flml
/data/<username>/fluidity/bin/fluidity my.flml

/home/<username>/fluidity/bin/fluidity -l -v2 my.flml
```

Imperial College London

Visualising your output

```
paraview --state=stommel.pvsm
```