Scientific Computing in Shallow Water Flows

Nicole Pitre

Supervisor: Dr. Francis Poulin

Research Project Goal

- The main goal of the project was to learn how to use a powerful scientific computing library called FEniCS
- A sub-goal was to learn how to use a finite element grid generation software called Gmsh to create regular and irregular geometries
- Another sub-goal was to learn how to efficiently integrate Gmsh meshes into FEniCS
- After understanding the framework of FEniCS, the goal was to adapt the library to study the Rotating Shallow Water model, which is important in studying large-scale oceanographic and atmospheric flows

What is FEniCS?

- Scientific computing is one of the most powerful tools available to solve complex scientific problems
- FEniCS is a free scientific computing library based on the Finite Element method
- It can be used to solve a variety of different systems of partial differential equations in an automated, efficient way
- It is very versatile and is able to parallelize very efficiently on a large number of computational cores

What is Gmsh?

- Gmsh is a free 3D finite element grid generator with a built-in CAD engine and post-processor
- Its goal is to provide a fast and user-friendly meshing tool with parametric input and advanced visualization capabilities

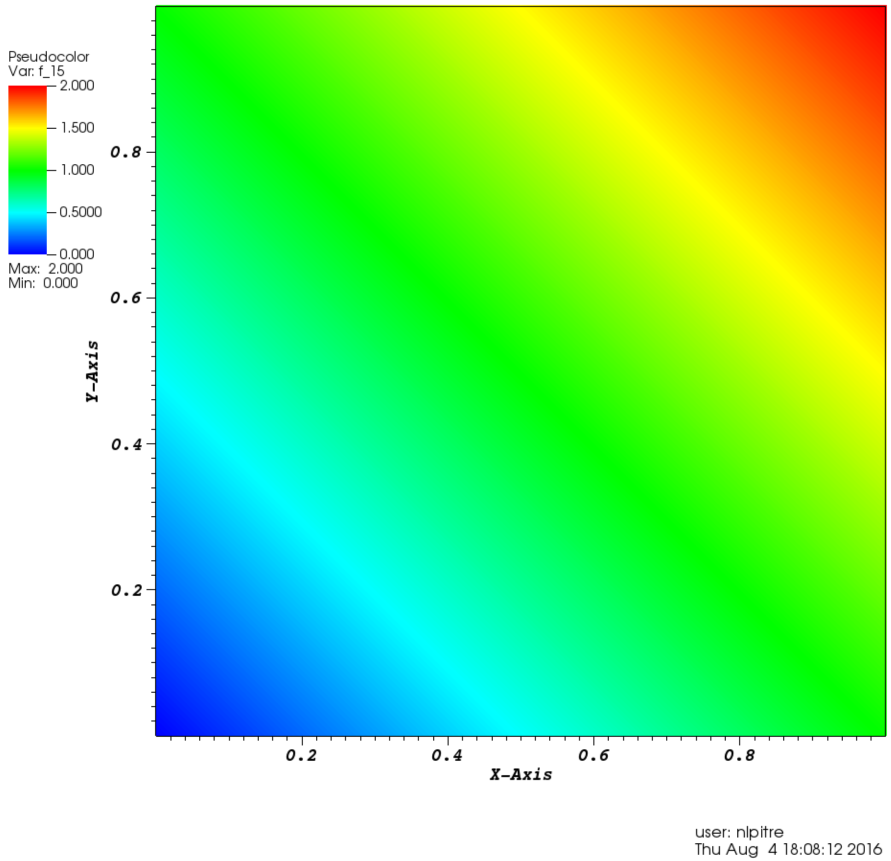
Steps to create a FEniCS Demo for a particular geometry

1. Create a geometry file in the Gmsh script language.
2. Use Gmsh to generate a finite element mesh from the geometry file.
3. Convert the mesh into XML format using the dolfin-convert script (also produces an XML subdomains file).
4. Write a Python demo script using the FEniCS DOLFIN module.
5. Add the XML mesh file into the Python demo script.
6. Open FEniCS in a virtual environment such as Docker and then run the demo.
7. View the solution in VisIt.

Demo: Solving Laplace's Equation on a Unit Square Mesh

- In this demo we use FEniCS to solve Laplace's equation on a regular grid geometry (a unit square)
- We could have also used a built-in FEniCS class called UnitSquareMesh() instead of writing our own geometry file – most regular geometries are built in to FEniCS

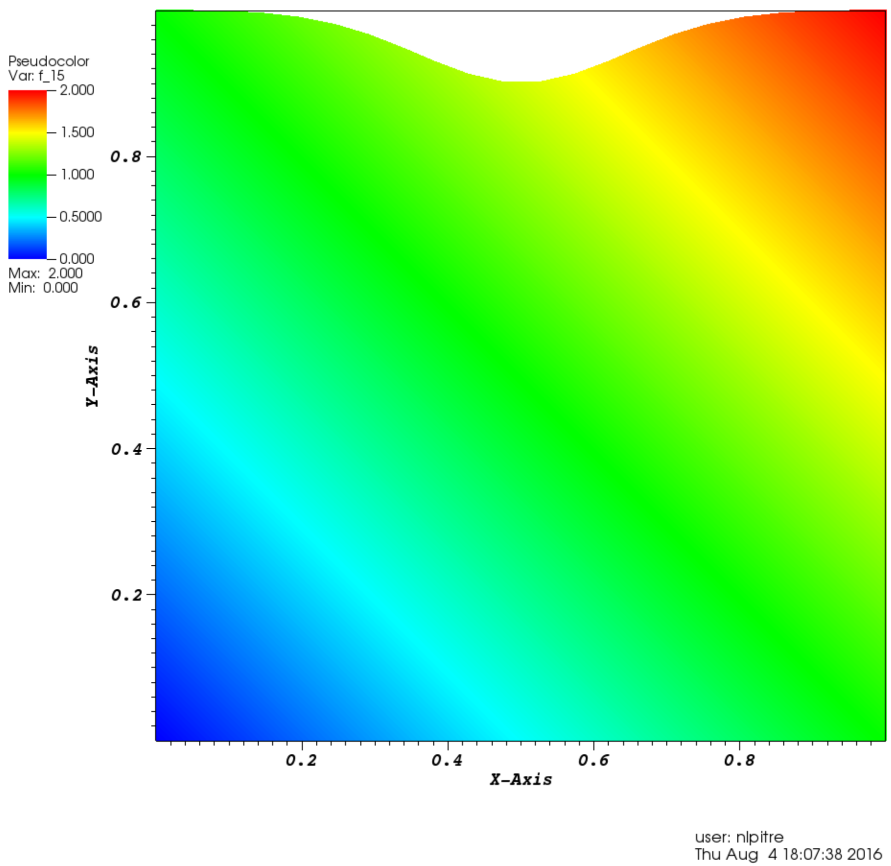
Solution: Solving Laplace's Equation on a Unit Square Mesh



Demo: Solving Laplace's Equation on a Cape Mesh

- Now we use FEniCS to solve Laplace's equation on an irregular geometry: a cape mesh
  - Note: a cape usually indicates a marked change in the trend of a coastline
- Notice how the solution is similar to that of the last demo, but we have used a more suitable mesh to model the changes in a coastline

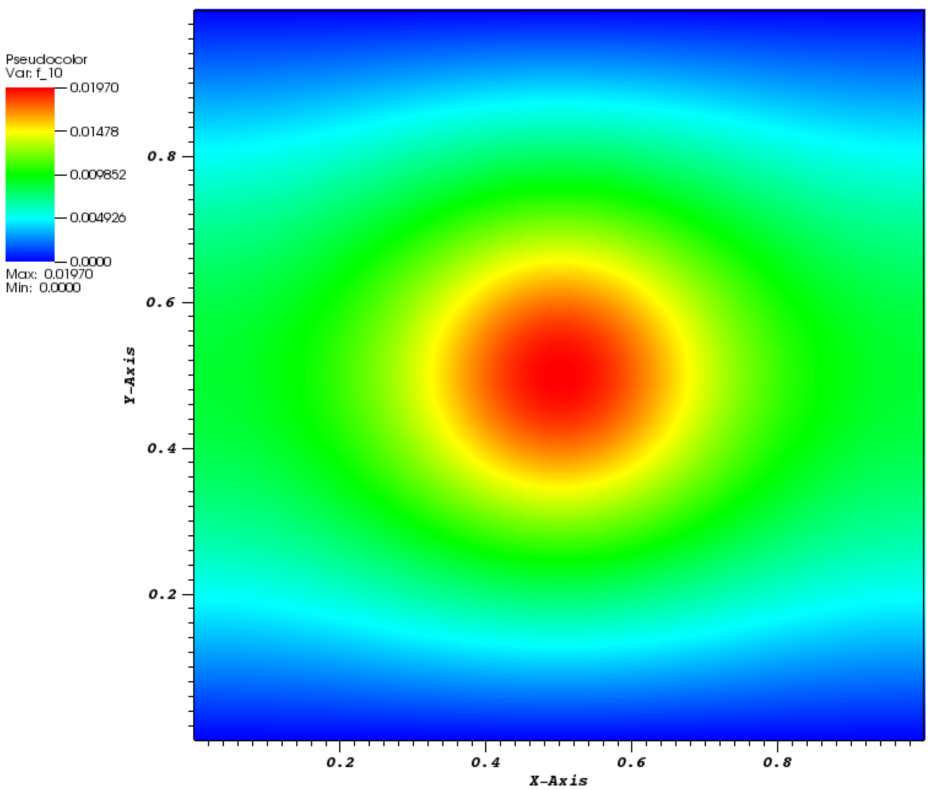
Solution: Solving Laplace's Equation on a Cape Mesh



Demo: Enforcing Periodic Boundary Conditions

- In this demo we solve Poisson’s equation on a square mesh while enforcing periodicity in both the x and y directions
- FEniCS makes it easy to enforce periodic boundary conditions in 1, 2 and 3 directions

Solution: Enforcing Periodic Boundary Conditions



Going Forward…

- We will share our code with the scientific community
- We hope that this code will enable students and researchers to study the underlying physical processes in geophysical flows in regular and irregular geometries more easily since they will not have to develop the tools themselves
- The interaction between FEniCS and Gmsh is not well documented – our research from this summer will hopefully clarify this interaction