Assignment 2

On Sunday, 30th October 2005, the Mountbatten building at the University of Southampton burnt to the ground. The Mathematics building is just south of this: however, the fire did not spread so it was untouched.

The Mountbatten building contained a lab housing a range of hazardous materials. The polluted air column rapidly rose and spread.

Assignment

1. Choose a way of modelling the pollutant spread.
2. Choose a method to numerically solve your model on an unstructured grid.
3. Write a code to solve the model using the chosen method.
4. Find how much of the pollutant passed over the University of Reading, using some of the provided grids.
5. Write a report of no more than ten pages (not including references and appendices) explaining your results and their accuracy, both in terms of modelling and numerics.

Grids

Grids provided

Two sets of unstructured grids are provided, using different reference lengthscales.

1. England, Scotland, Wales
2. Southern Local Authorities

The expectation is that only one set of grids will be used. Different lengthscales can be used for grid convergence studies.

Conventions and locations

The coordinates used are given in metres (the projected coordinate reference system is EPSG:27700). In this coordinate system relevant locations are

- University of Southampton, Mountbatten Building: (442365, 115483)
- University of Reading, Brian Hoskins Building: (473993, 171625)

File usage

The grids are defined in terms of three files:

1. `<name>_nodes_<scale>.txt`: the location of the nodes of the grid;
2. `<name>_IEN_<scale>.txt`: the mapping from global element number to global node numbers;
3. `<name>_bdry_<scale>.txt`: which global node numbers lie on the boundary.

These can be imported and used with standard `numpy` and `matplotlib` functions. For example,

```
nodes = np.loadtxt('data/esw_nodes_100k.txt')
IEN = np.loadtxt('data/esw_IEN_100k.txt', dtype=np.int64)
boundary_nodes = np.loadtxt('data/esw_bdry_100k.txt',
                            dtype=np.int64)

plt.triplot(nodes[:,0], nodes[:,1], triangles=IEN)
plt.plot(nodes[boundary_nodes, 0], nodes[boundary_nodes, 1], 'ro')
plt.axis('equal')
```

will load and plot the coarsest grid, marking the boundary points.

Models

Equations

The minimal model would solve for the pollutant as a scalar field indicating the pollutant concentration normalized to one over Southampton. This could be done using a time independent or time dependent model. The velocity field could be imposed (in the simplest model) or solved for (much more complex). A reasonable approximation to the wind conditions that day would be a constant 10 metre per second wind to the north.

Source or boundary conditions

The pollutant was sourced by the fire at a very specific and narrow location. However, on all practical grids this source will be within a single element, and that element will be very close to the boundary. The pollutant can therefore be injected into the domain either using a localised source function (local in space for stationary models; for time dependent models, the fire lasted about 8 hours) or by setting Dirichlet boundary conditions over some relevant boundary nodes.

Methods

In this section we have seen essentially three methods that would work on unstructured meshes;

1. time independent Galerkin finite element solutions of diffusion equations (advection terms can be added, with care);
2. time dependent Galerkin finite element solutions of advection equations (diffusion terms can be added, with care);
3. time dependent Discontinuous Galerkin finite element solutions of advection equations (adding diffusive terms requires careful work).

The choice of method should be linked to the model chosen.

Code from standard modules (eg `numpy`, `scipy`, `quadpy`) can be used, but full PDE solving packages (eg `FEniCS`, `dedalus`) cannot.

Report

The report should be between 6 and 10 pages and should include

1. the model used and the equations to be solved;
2. the method used, appropriately referenced. If terms are needed extending beyond those given in notes or lectures they should be defined in full (calculations can be deferred to appendices);

3. the choice of initial and boundary conditions;
4. the choice of output quantity to represent “pollutant over Reading”;
5. the results, with appropriate figures;
6. a discussion/critique of the accuracy of all choices made and methods used.

Code

- The report should include a link to an accessible version control repository containing the code.
- The code should be runnable in “finite time” - less than 10 minutes maximum on a laptop.
- Code documentation is essential.
- Code tests are useful.

Submission

Email a PDF version of the report together with a link to the version control repository by the end of Friday 20th December 2024 to I.Hawke@soton.ac.uk.