Lecture 6: PDEs in higher dimensions

Advection

1D: $u_t + au_x = 0$.

In 2D:

$$u_t + a(x,y)u_x + b(x,y)u_y = 0.$$

Or more generally, we can write this as:

$$u_t + \nabla \cdot (\vec{w}u) = 0,$$

with a vector field $\vec{w}(x,y)$.

▶ [m25_2d_adv.m]

- Advection and the wave equation are quite different from diffusion: they are hyperbolic and “information” about the solution travels along characteristics. These are the lines traced out my the vector field $w(x, y)$.
- The numerics are a bit different too: this code uses “upwinding” finite differences which are appropriate for advection-dominited problems, but we haven’t talked about them in this course.
- But we could look more carefully about constructing the matrices in this code…

Heat equation

$$u_t = \nabla^2 u = u_{xx} + u_{yy}$$

on a square or rectangle. We can apply centered 2nd-order approximation to each derivative.

In the method of lines approach, we write

$$u_{xx} + u_{yy} \approx \frac{v^n_{i-1,j} - 2v^n_{ij} + v^n_{i+1,j}}{h^2} + \frac{v^n_{i,j-1} - 2v^n_{ij} + v^n_{i,j+1}}{h^2}.$$

This gives a stencil in *space* (then still need to deal with time).

```
             (1)
              |  (i,j+1)
              |
              |         (i+1,j)
(1)------(-4)-----(1)
              |(i,j)
              |
              |
             (1)
```

- Using forward or backward Euler, accuracy is $O(k + h^2)$.
- And a stability restriction for FE of $k < h^2/4$.
- In principle, our "finite difference Laplacian" maps a matrix of 2D grid data to another such, and is thus a "4D tensor". However, in practice we stretch out 2D to 1D, so that the tensor becomes a matrix:

$$\frac{v}{dt} = Lv.$$

- How does this "stretch" work? It defines an ordering of the grid points. In Matlab: meshgrid() and (:), see later.

Matrix structure

Let's look at the structure of L. We choose an ordering for the grid points and assume zero boundary conditions:

```
^ y
|       0     0     0
|
|  0   x4    x8    x12   0
|
|  0   x3    x7    x11   0
|
|  0   x2    x6    x10   0
|
|  0   x1    x5    x9    0
|
|       0     0     0
+----------------------------> x
```

The discrete Laplacian now looks like this:

```
              |-4   1   0   0    1   0   0   0    0   0   0   0|  |v1 |
              | 1  -4   1   0    0   1   0   0    0   0   0   0|  |v2 |
              | 0   1  -4   1    0   0   1   0    0   0   0   0|  |v3 |
              | 0   0   1  -4    0   0   0   1    0   0   0   0|  |v4 |
              |                                                |  |   |
        1     | 1   0   0   0   -4   1   0   0    1   0   0   0|  |v5 |
Lv  =  ---    | 0   1   0   0    1  -4   1   0    0   1   0   0|  |v6 |
       h^2    | 0   0   1   0    0   1  -4   1    0   0   1   0|  |v7 |
              | 0   0   0   1    0   0   1  -4    0   0   0   1|  |v8 |
              |                                                |  |   |
              | 0   0   0   0    1   0   0   0   -4   1   0   0|  |v9 |
              | 0   0   0   0    0   1   0   0    1  -4   1   0|  |v10|
              | 0   0   0   0    0   0   1   0    0   1  -4   1|  |v11|
              | 0   0   0   0    0   0   0   1    0   0   1  -4|  |v12|
```