MPI — via Python (mpi4py)

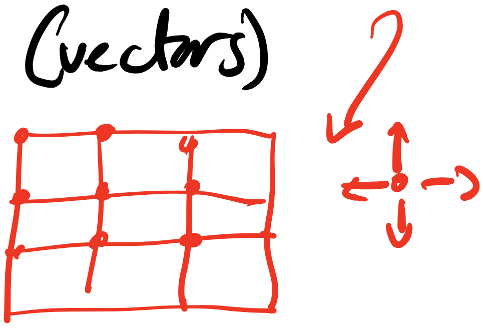
Parallel datastructures for PDEs

$\partial_t u - \nabla^2 u = f$

$A x = b$ ← vector

← matrix

- → linear algebra (vectors)
- → grids/meshes



multigrid

- → Computing in parallel

```
for (i ... )
    $a[i] = \frac{b[i-1] - b[i+1]}{2}$
```

- → Operators (matrices)
- → Data (re)distribution

Implementation → hands on stuff

→ we'll do some live coding.

Today

— set up environment (hopefully)

— Why does parallel work?

Coursework: implementation of some of these ideas + use to solve some PDE.

→ Large scale parallel ⇒ Hamilton.

Day-to-day parallel / development / debugging
⇒ on your machine.

- → An MPI library
- → A python venv with mpi4py & numpy scipy . . .

Windows: WSL → then ubuntu
or (maybe?) → conda
mpi4py package.

MacOS: I recommend homebrew
→ brew install mpich.

Debian-based: libmpich-dev

Fedora
Arch | similar?