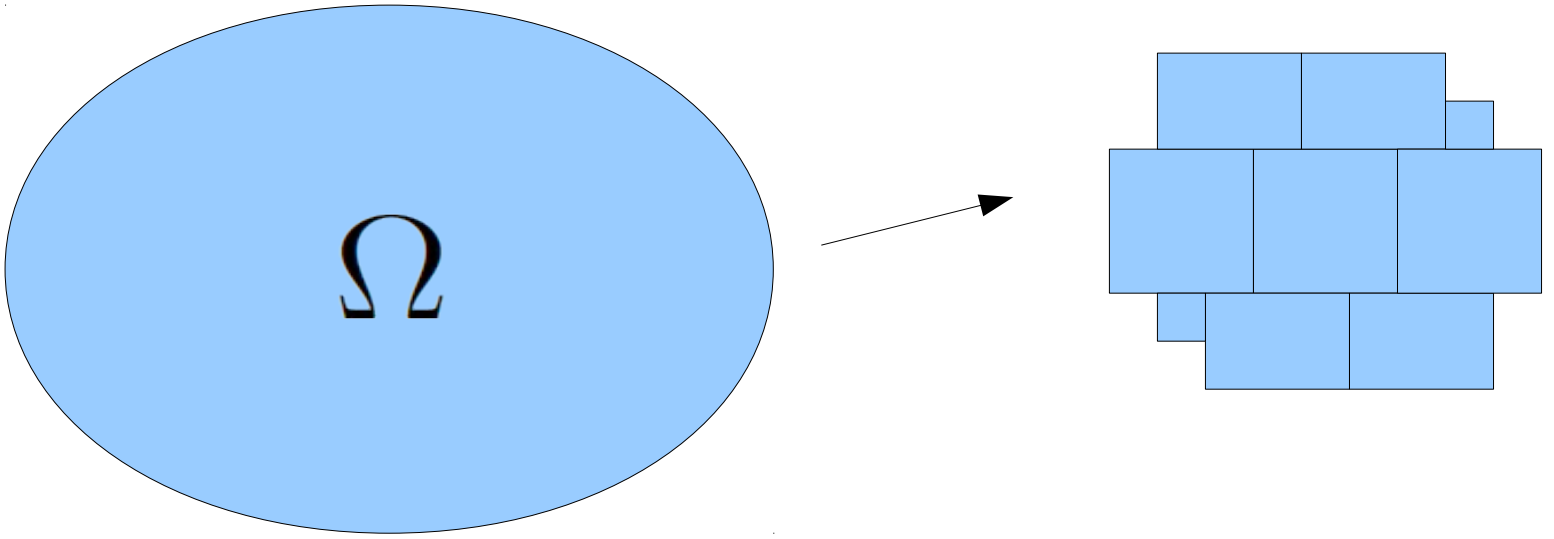
Chombo BootCamp

Tools for Drivers

A driver conveys problem-specific information to Chombo



$$\nabla^2 \phi = \rho$$

- Describes a problem or a family of problems (e.g. Poisson's equation, Euler equations)

A driver performs 3 basic tasks

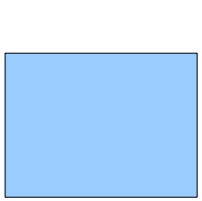
- Gets input data from a file
  - Parameters for problem (coefficients)
  - Initial conditions, boundary conditions
  - Discretization parameters (grid spacing, etc)
  - Parameters for running simulations (plots, etc)
- Translates data into Chombo objects
- Runs the simulation

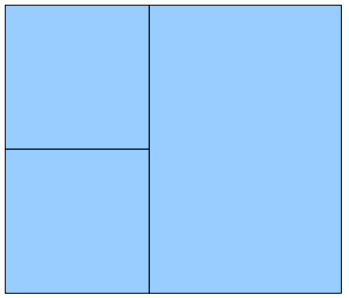
Parsing data is the hard part of writing a driver

- Mathematical objects

$$a, b, \mathbf{x}, \mathbf{T}, f(\mathbf{x}), \mathbf{u}(\mathbf{x}, t) \qquad (s_0, s_1, s_2, \ldots, s_N)$$

- AMR





Higher-level languages are good at this stuff!

Enter Python (www.python.org)

- Open-source
- Widely adopted by scientific community
  - Similar in syntax to Matlab, Fortran
- Interpreted, not compiled

```
a = 1
def dist(x1, x2):
    return sqrt((x1[0]-x2[0])**2 + (x1[1]-x2[1])**2)
x = (1.0,2.0)
y = (3.0,4.0)
print dist(x,y)
```

PyParse parses Python scripts

In Python:

(contents of problem.inputs)

```
name = 'problem'
a = 12
omega = 4
def f(x, t):
   return a * \
        exp(-x[0]*x[0]) * \
        cos(omega*t)

refRatios = [2,2,4,2]

b1 = Box(lo=(0,0),hi=(31,31))
b2 = Box((0,32),(64,32))
boxes = [b1,b2]
```

In C++:

```
PyParse parser("problem.inputs");

string name;
parser.get("name", name);
Real a, omega;
parser.get("a", a);
parser.get("omega", omega);

RefCountedPtr<ScalarFunction> f;
parser.get("f", f);

vector<int> refRatios;
parser.get("refRatios", refRatios);

Box b1;
parser.get("b1", b1);
vector<Box> boxes;
parser.get("boxes", boxes);
```

PyParse allows quicker development

- replaces ParmParse, an older key-value parser
- can do math – no need for preprocessing
- requires less C++, less compilation
- supports nested sequences

  [[(1,2,3), (4,5,6)], (7,8,9)], 10]
- supports higher-level Chombo data types (Boxes, scalar/vector functions, implicit functions)
- is easily extendable

Examples and documentation are forthcoming

- PyParse will get an appendix in the Chombo document
- Most drivers/examples currently use ParmParse

If you're interested in using PyParse in your work, email jnjohnson@lbl.gov.