AMReX Workshop

James Le Houx

Samvir Thandi

What is AMReX?

- AMReX: Adaptive Mesh Refinement EXascale
- Software framework for building massively parallel block-structured adaptive mesh refinement (AMR) applications.
- Developed at Lawrence Berkeley National Laboratory (LBNL), National Renewable Energy Laboratory (NREL), and Argonne National Laboratory (ANL) as part of the Block-Structured Adaptive Mesh Refinement (AMR) Co-Design Center in the United States Department of Energy’s Exascale Computing Project.
- Exascale computing refers to computing systems capable of one exaFLOPS or a billion billion floating point operations per second
- Allows computational power to be focused on the most critical parts of the problem in the most computationally efficient way.
- Latest development in AMR by Chombo and BoxLib developers
- Available at: https://github.com/AMReX-Codes/amrex

Features

- 200 times faster than previous implementations of AMR [1]
- C++ and Fortran
- 1D, 2D, 3D support
- Support for various node types (cell-centred, face-centred etc)
- Parallelisation support using OpenMP or MPI
- Parallel I/O
- Visualisation support (AmrVis, VisIt, Paraview, and yt)
- Open Source, contributions welcome
- Regular updates (monthly)

[1] https://crd.lbl.gov/news-and-publications/news/2017/berkeley-lab-led-ecp-co-design-center-achieves-orders-of-magnitude-speed-up-in-latest-software-release/

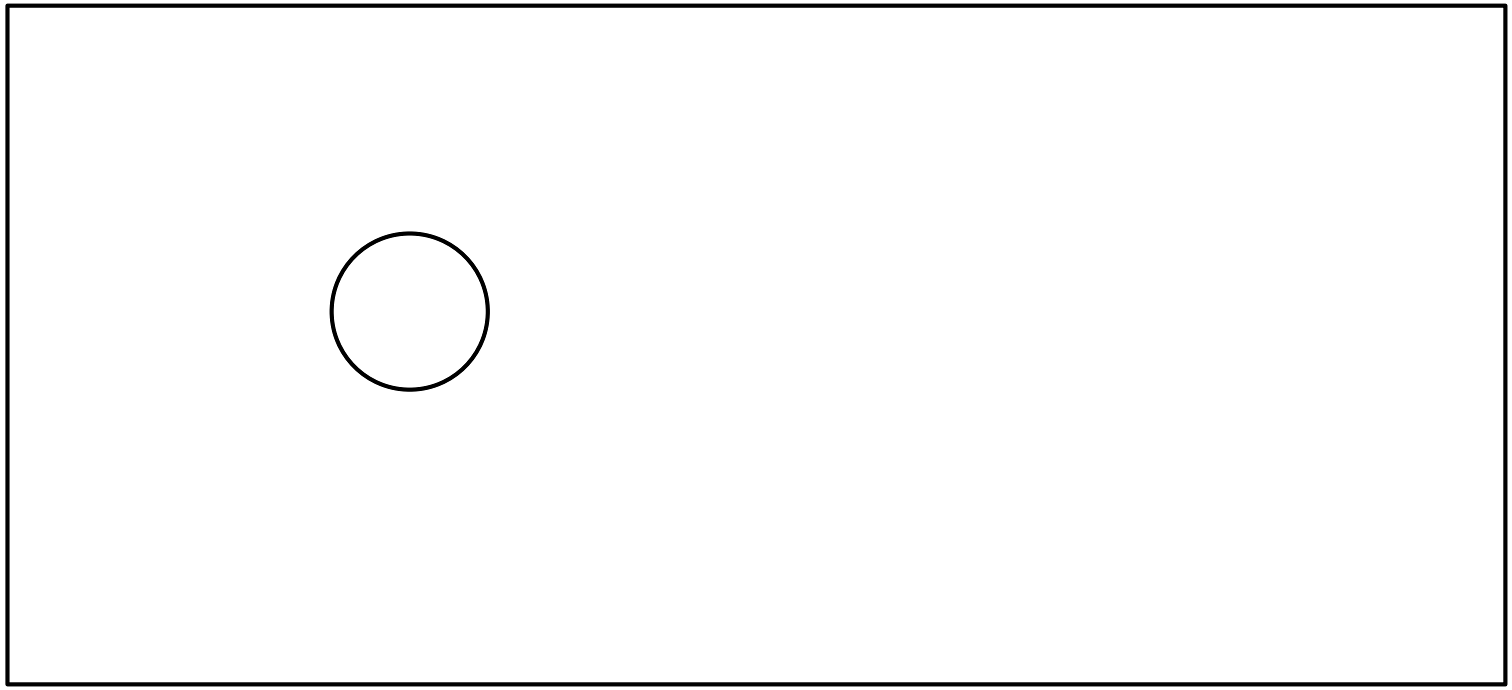
What is a grid and why is it needed?

- A way of turning a physical problem into a mathematical model
- Majority of simulation tools use a grid
- Example is computational fluid dynamics (CFD)
- Grids are a crucial part of a simulation, important to get right
- They need to be adapted around regions of interest or large change in gradients. Can be globally or locally refined
    - AMReX allows this to be done automatically on a local scale

Simulation procedure

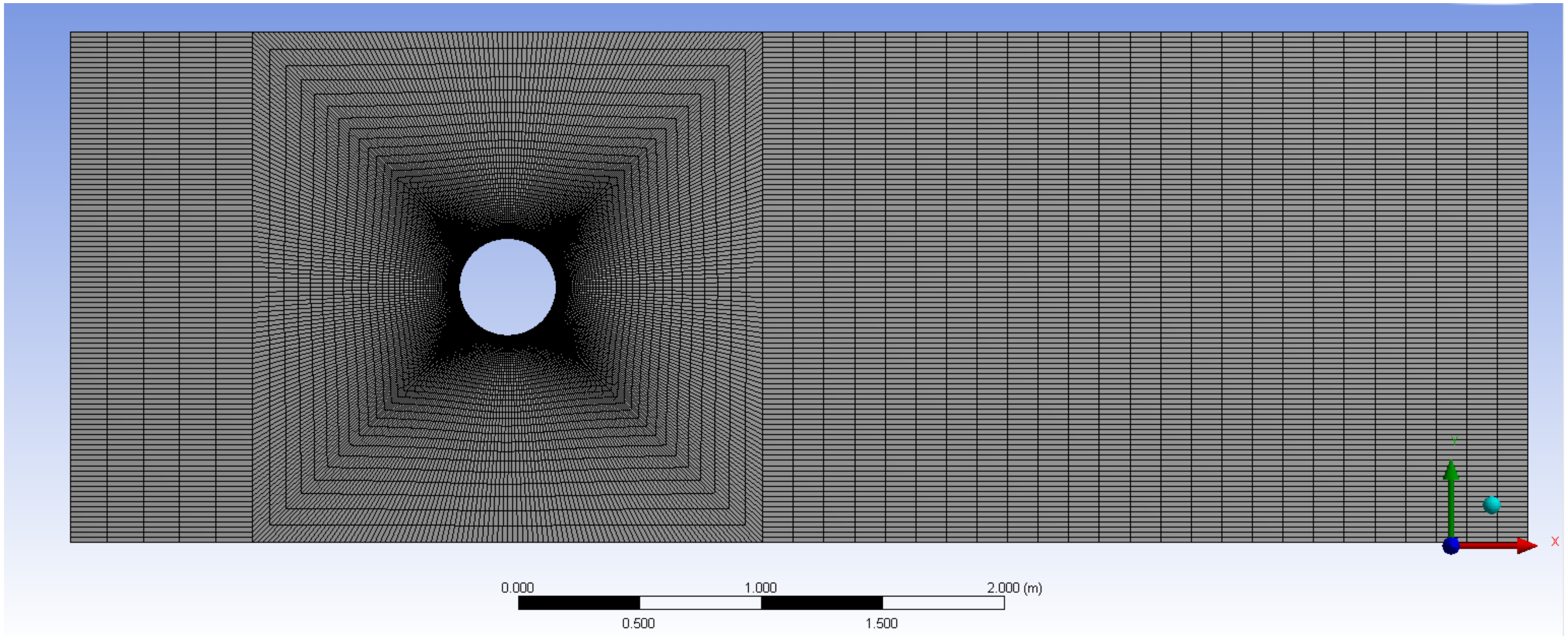
Geometry

- Start with a physical problem you want to analyse (e.g. vortex shedding over a cylinder in a wind tunnel



Meshing

- Convert the physical problem into a mesh/grid. Known as discretisation
- This allows the problem to be represented as a finite number of control volumes
- Apply various conservation law equations to control volumes
- Run for a certain number of iterations and time steps.
- Ensure the problem is independent of the mesh (grid refinement)
  - This is usually done before or after the simulation has run

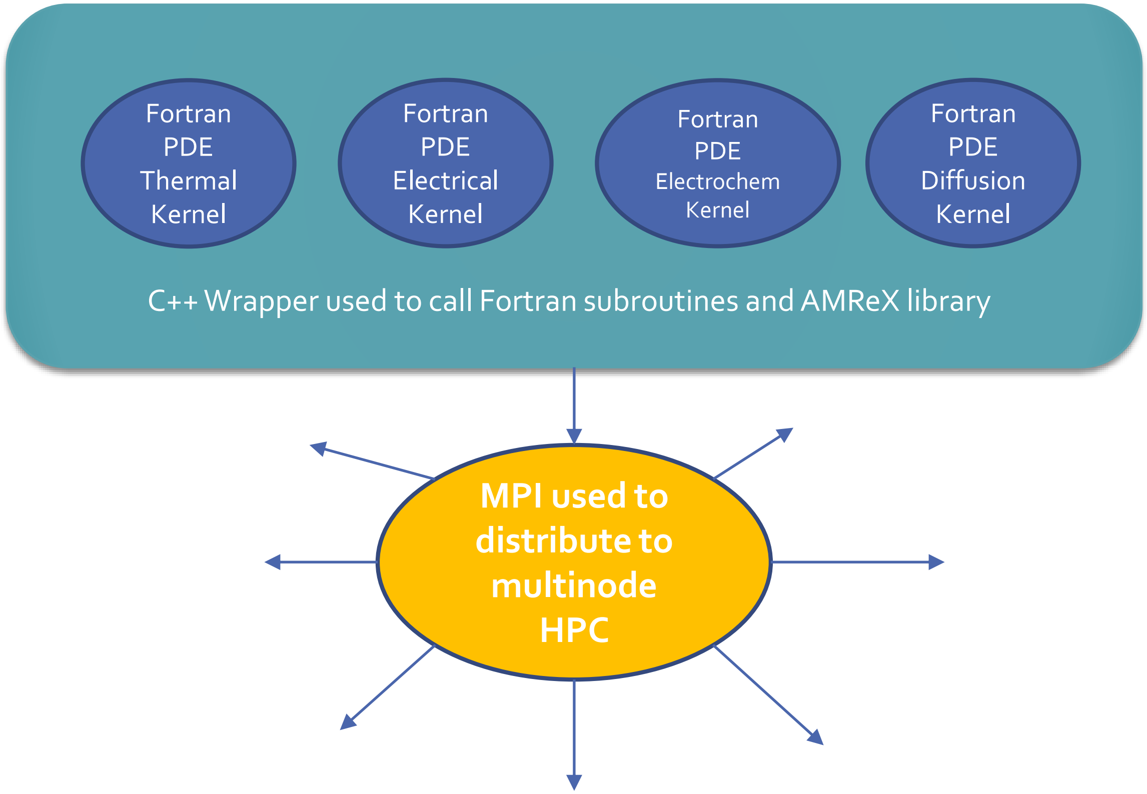


0.000
1.000
2.000 (m)
0.500
1.500
Y
X

Why use AMReX

- Allows the grid to be locally refined only in areas of interest
- Refinement happens as simulation progresses
- Can save computational power
- Can reduce the amount of time spent on grid refinement

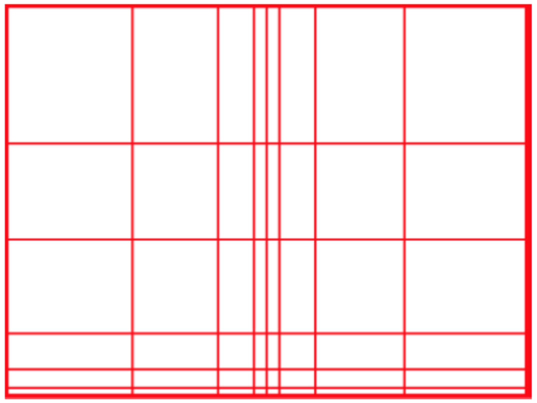
How does AMReX work?


Fortran PDE Thermal Kernel
Fortran PDE Electrical Kernel
Fortran PDE Electrochem Kernel
Fortran PDE Diffusion Kernel
C++ Wrapper used to call Fortran subroutines and AMReX library
MPI used to distribute to multinode HPC

PDE – Partial Differential Equation
MPI – Message Passing Interface
HPC – High Performance Computing

Types of Structured Mesh

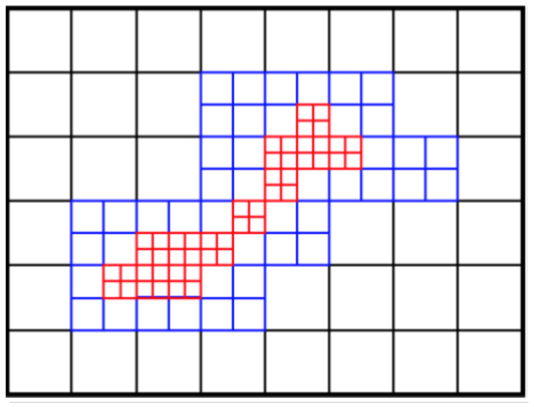
- Mesh Distortion (Biasing)



EPSRC
Engineering and Physical Sciences
Research Council

Types of Structured Mesh

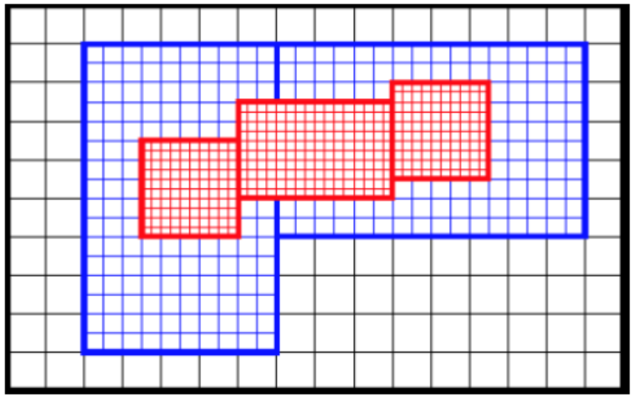
- Mesh Distortion (biasing)
- Point-wise (tree based)



EPSRC
Engineering and Physical Sciences
Research Council

Types of Structured Mesh

- Mesh Distortion (biasing)
- Point-wise (tree based)
- Block-structured (AMReX)



In block-structured AMR the solution is defined on a hierarchy of levels of resolution – a union of rectangular grids which can change dynamically.

EPSRC
Engineering and Physical Sciences
Research Council

Kelvin-Helmholtz Instability

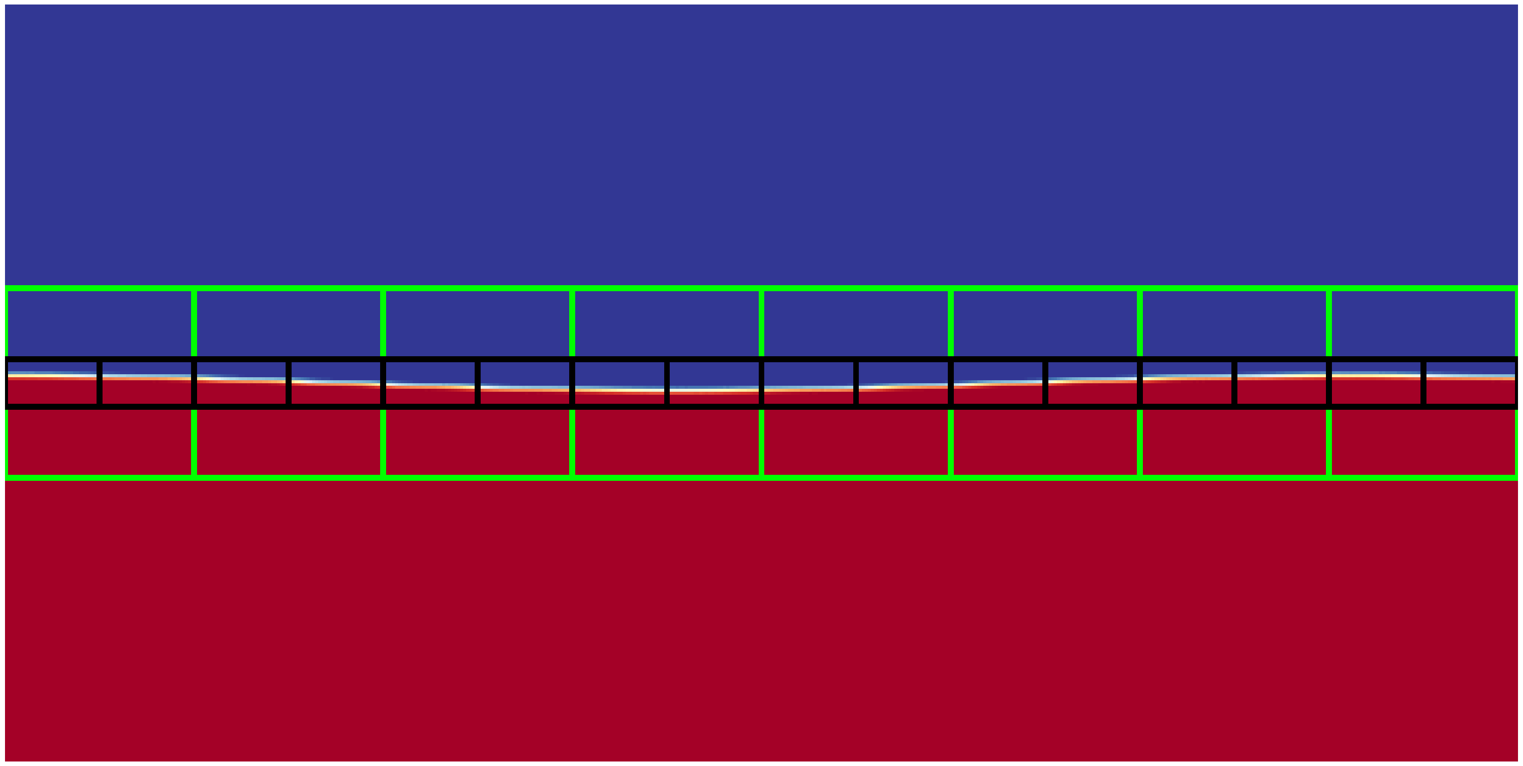


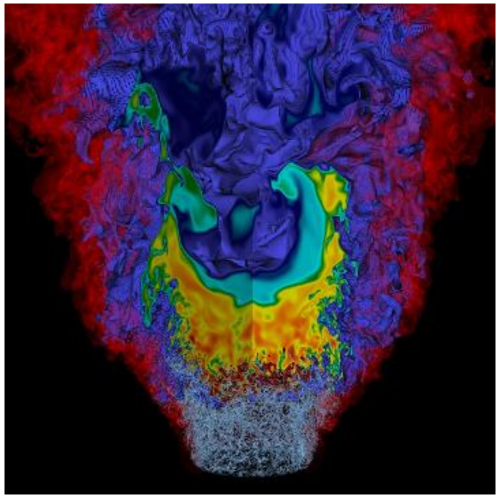
Figure 1: A depiction of the Kelvin-Helmholtz instability using a 3 levelled adaptive mesh solving incompressible Navier-Stokes equations [1]

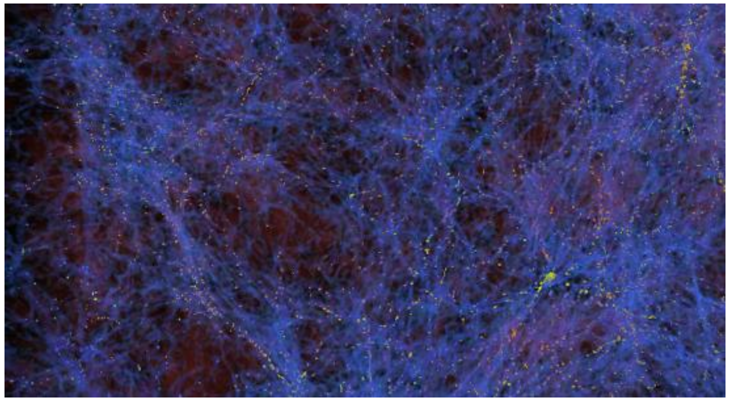
[1] Why Block Structured AMR?, AMReX, https://amrex-codes.github.io/overview.html Date Accessed: 19/03/18

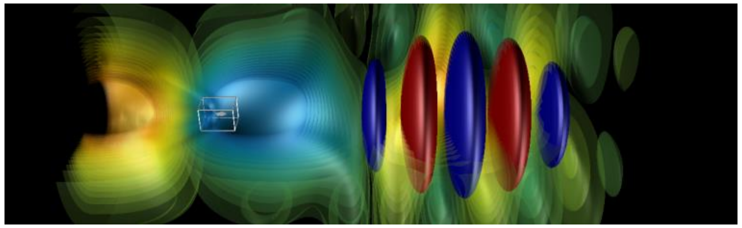
Adaptive Mesh Isn't New

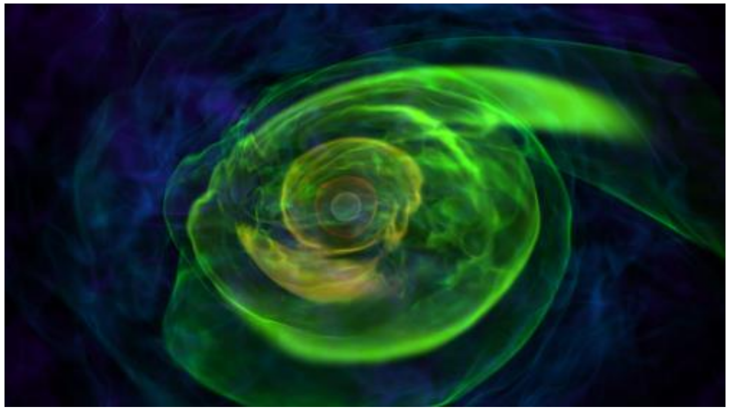
- First proposed back in the 1980's, there have been a multitude of different frameworks and languages
- As we have seen, there are a number of ways of refining the grid (distortion, point-wise, block-structured)
- The AMReX 'family tree'
    - Boxlib -> Chombo -> AMReX

Current Applications Using AMReX


Combustion


Cosmology


Accelerators


Astrophysics

[1] Why Block Structured AMR?, AMReX, https://amrex-codes.github.io/overview.html Date Accessed: 30/04/18

Getting started

Code can be downloaded from GitHub

```
sudo apt install git
cd /home/feeg6003/Documents/
git clone https://github.com/AMReX-Codes/amrex.git
```

Hello World exercise

```
#include <AMReX.H>  // AMReX header file
#include <AMReX_Print.H> // allows the AMReX print command

int main(void)
{
    amrex::Initialize(argc,argv);
    amrex::Print() << "Hello world from AMReX version "
                   << amrex::Version() << "\n";
    amrex::Finalize();
}
```

Building

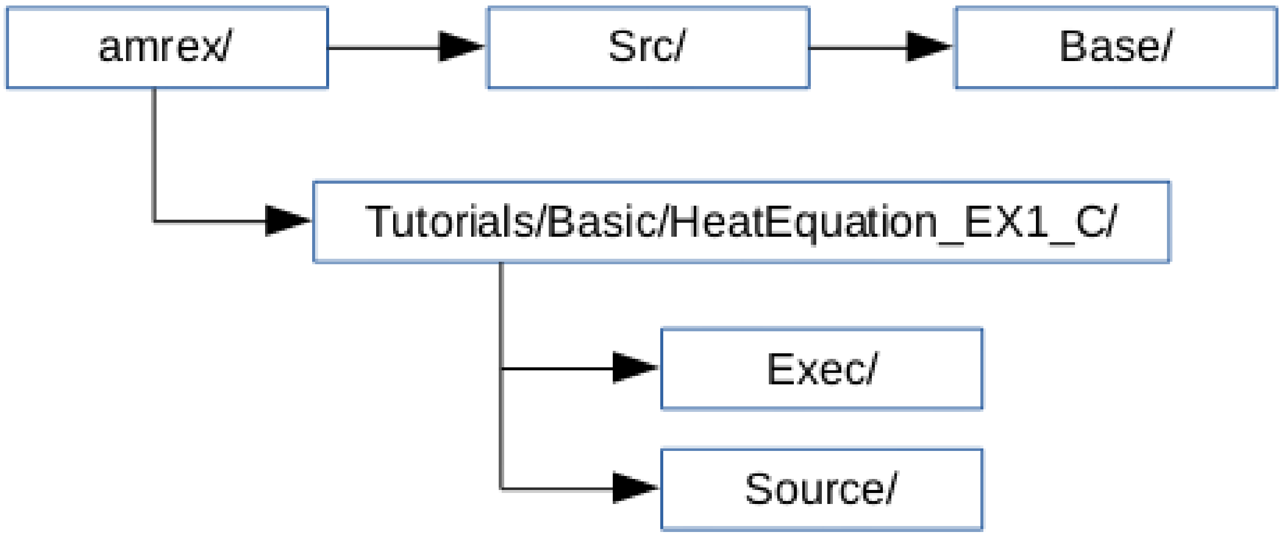
```
make
```

- This will start the compilation process
- Output:

```
main3d.gnu.DEBUG.ex
```

Change variables in GNUMake

```
DIM = 2
USE_MPI = TRUE
```


amrex/
Src/
Base/
Tutorials/Basic/HeatEquation_EX1_C/
Exec/
Source/

Velocity Example

Now we will perform a simple advection example, this first example applies a uniform velocity to a particle cluster.

The code is structured into two directories: 'Source' and 'Exec'.

First, we need to create an executable in the working directory, this is done using a makefile like in HelloWorld.

Next, in order to run the executable, we need to specify an inputs file:

```
./main2d.gnu.MPI.ex inputs
```

Velocity Example

Next, we want to view our output plot files from the simulation, in this virtual machine we will be using AMRVis. We need to call the amrvis executable so we will use the alias command to create a shortcut:

```
alias amrvis2d='/home/feeg6003/Documents
/Amrvis/amrvis2d.gnu.ex'
```

You can now open up our simulation results by running:

```
amrvis2d –a plt*
```

Have a look through the output plots. Now try changing variables in the inputs file and re-running the simulation.

Can you increase the number of adaptive mesh levels?

Vortex Example

Now we know how to run the velocity simulation, can you do the same with the vortex example?

Takeaway

- AMReX is a framework that alows you to build massively parallel multi-physics applications on block-structured grids
- A large amount of what you would normally have to write is supported within AMReX, speeding up development time:
  - MPI is as simple as USE_MPI=TRUE
  - Domain decomposition
  - Tiling of grids
  - Ghost cells
  - Multi-level meshing