Numerical simulation of reacting flows using an unstructured adaptive mesh refinement based code HAMISH

R. Stewart Cant, Jian Fang, Umair Ahmed, Girish Nivarti, Charles Moulinec, David R. Emerson, Nilanjan Chakraborty

UNIVERSITY OF CAMBRIDGE
DEPARTMENT OF ENGINEERING
Science & Technology Facilities Council
Newcastle University

Outlines

- Background
- Introduction to the HAMISH Code
- Code Tests
  - 1-D thermal diffusion
  - 1-D planar flame
  - 1-D HOQ
  - 2-D thermal diffusion
  - 3-D TGV
  - 3-D isotropic decaying turbulence
- Scalability and Code Profiling
- Summary and Perspectives

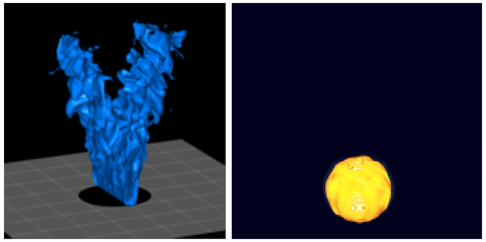
Background

- **Adaptive Mesh Refinement (AMR)**
  - AMR is a method of dynamically adapting the accuracy of a solution within certain regions of a simulation, i.e. during the time the solution is being calculated.
  - The main idea of AMR is to enable higher accuracy at lower cost, through an automatically optimal distribution of mesh cells in the concerned computational region.

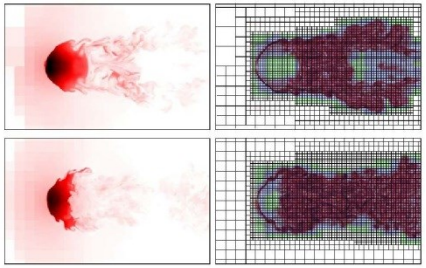
Background

- **Advantages of AMR**
  - Computational savings over a static mesh approach.
  - Storage savings over a static mesh approach.
  - Full control of the mesh resolution.
  - More detailed physics for the same number of cells as for a static mesh.
- **Main Applications**
  - To solve hyperbolic systems of conservation laws.
  - Problems with large dynamic range of scales.
  - Local enhancement of resolution is sufficient.

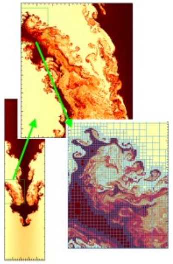
Background



Flames (Boxlib)

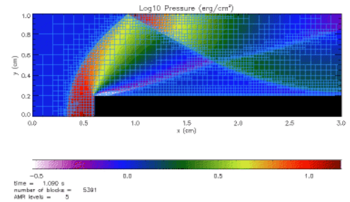


R-T instability (ENZO Code)

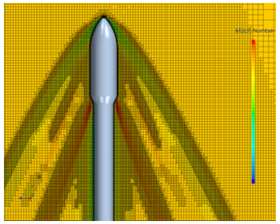


(FLASHCode)

Problems with interfaces

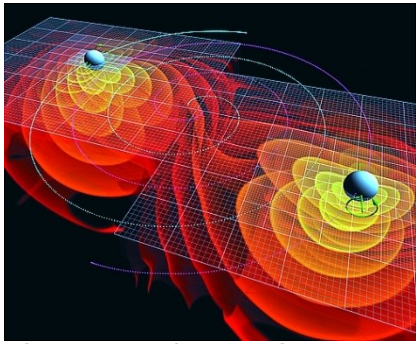


Moving Shock-Wave (PARAMESH)

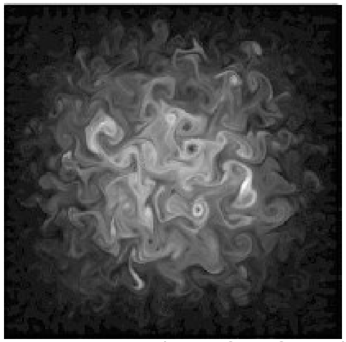


Supersonic Vehicle

Problems with discontinuities

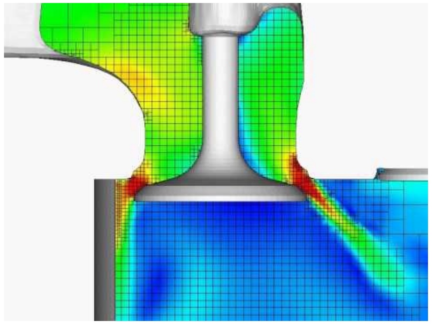


Computing Cosmic Cataclysms

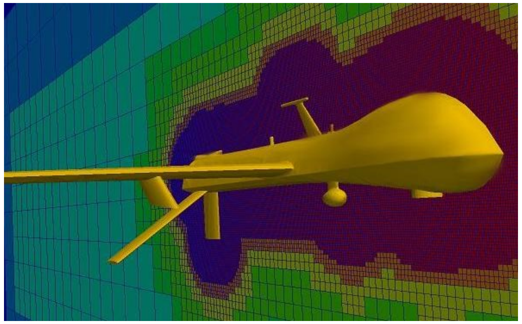


Turbulence (FLASH Code)

Problems with great variety of scales



Engine Combustion



Drone

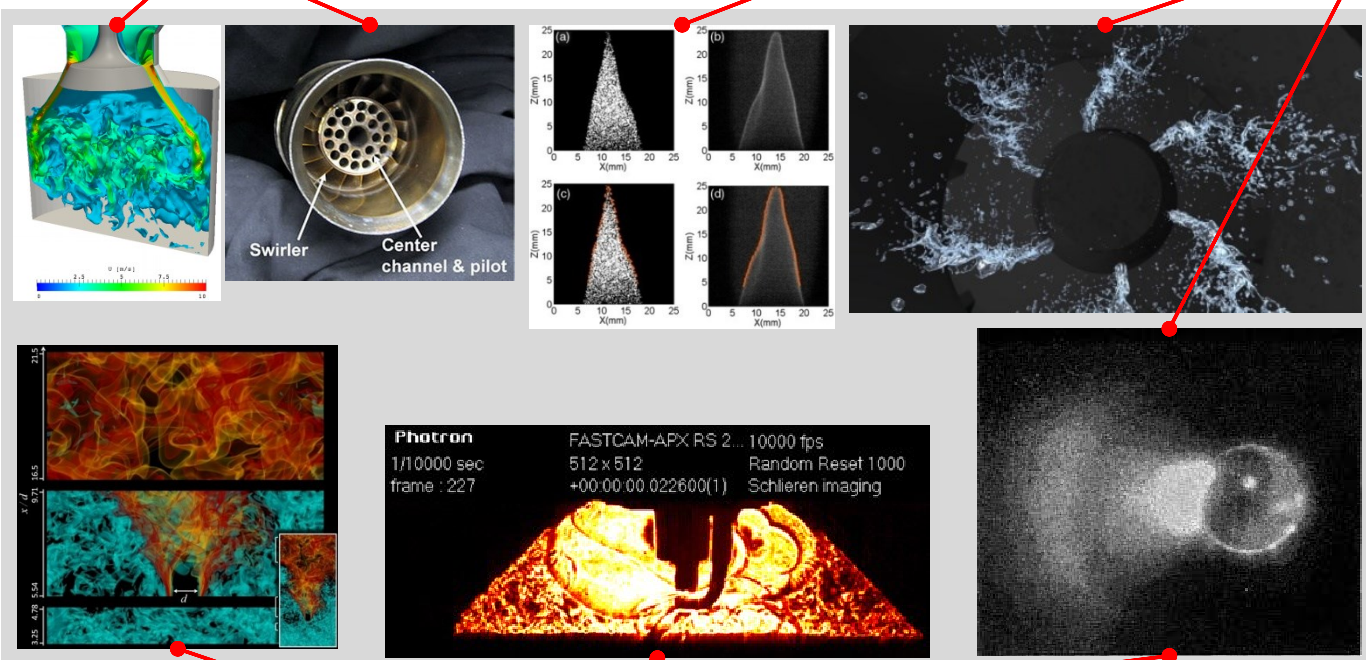
Problems with complex geometries

Background

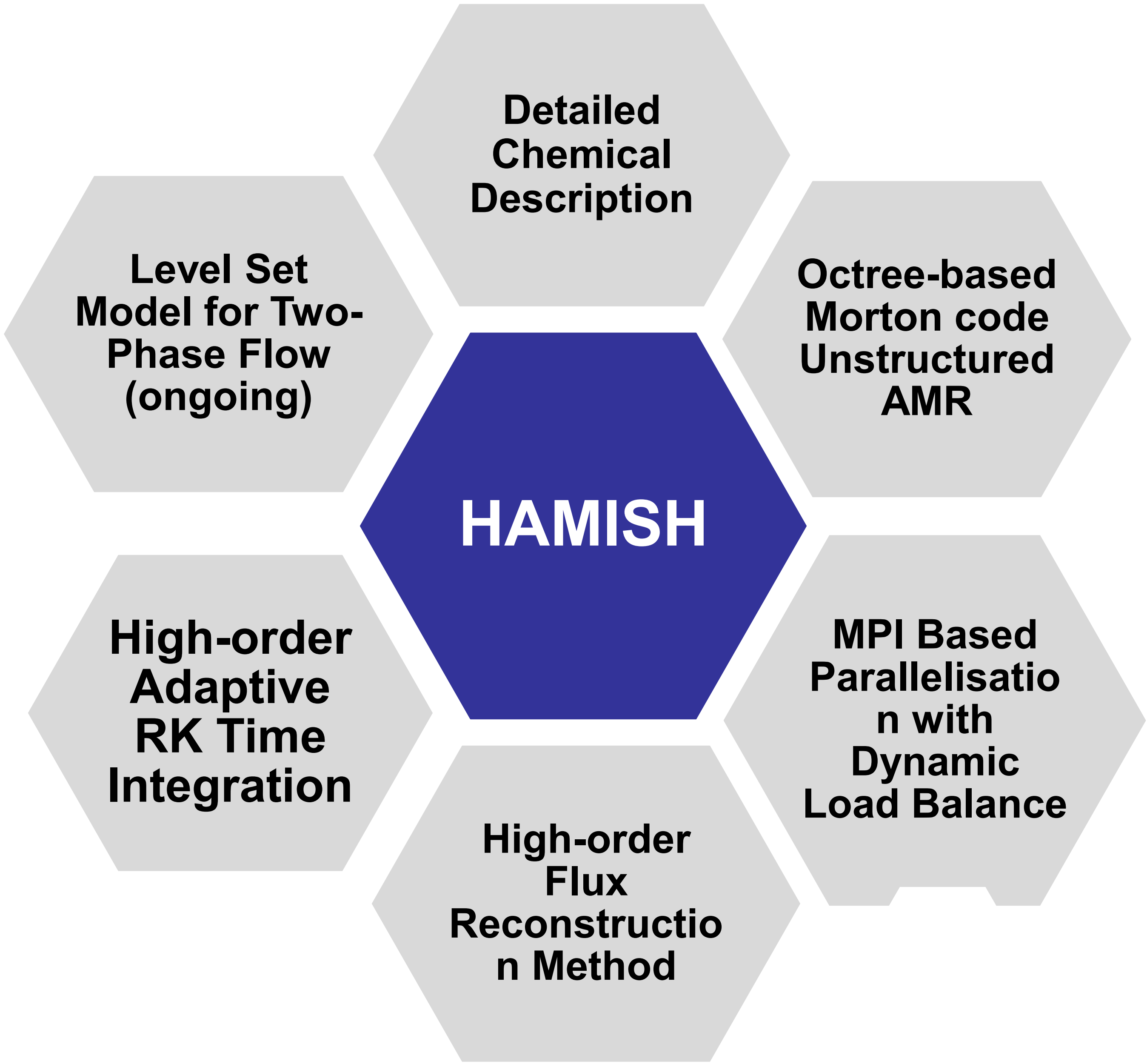
- **Flow dynamics in combustion**

Geometry Complexity
- •Swirler
- •Piston
- •Igniter
- •Spray

Interface
- •Flame front
- •Gas/Liquid interface
- •Droplets surface

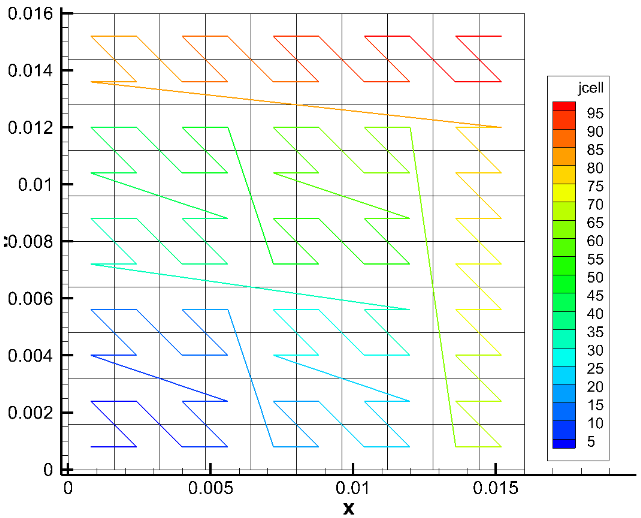


Scale-variety
- •Turbulence
- •Flame
- •Shock-wave
- •Droplets

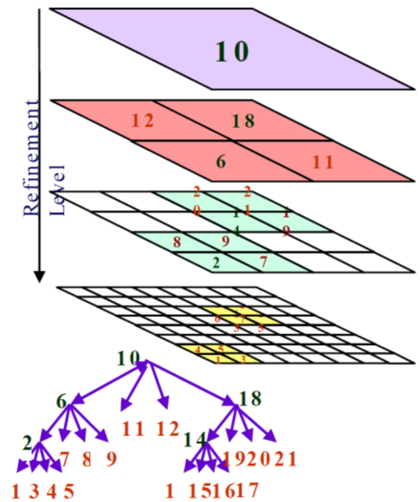

Detailed Chemical Description
Level Set Model for Two-Phase Flow (ongoing)
Octree-based Morton code Unstructured AMR
HAMISH
High-order Adaptive RK Time Integration
MPI Based Parallelisation with Dynamic Load Balance
High-order Flux Reconstruction Method

Introduction to the HAMISH Code

- **AMR in HAMISH (h-refinement)**

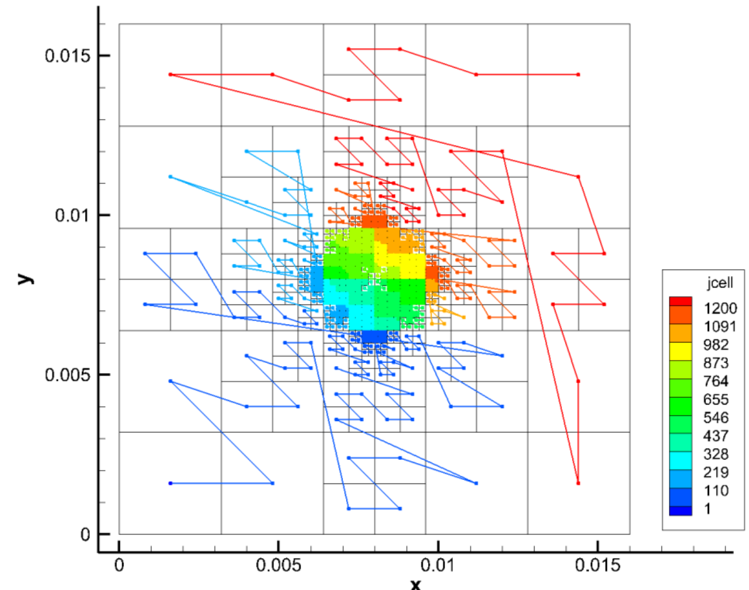


M-Code



Tree data structure

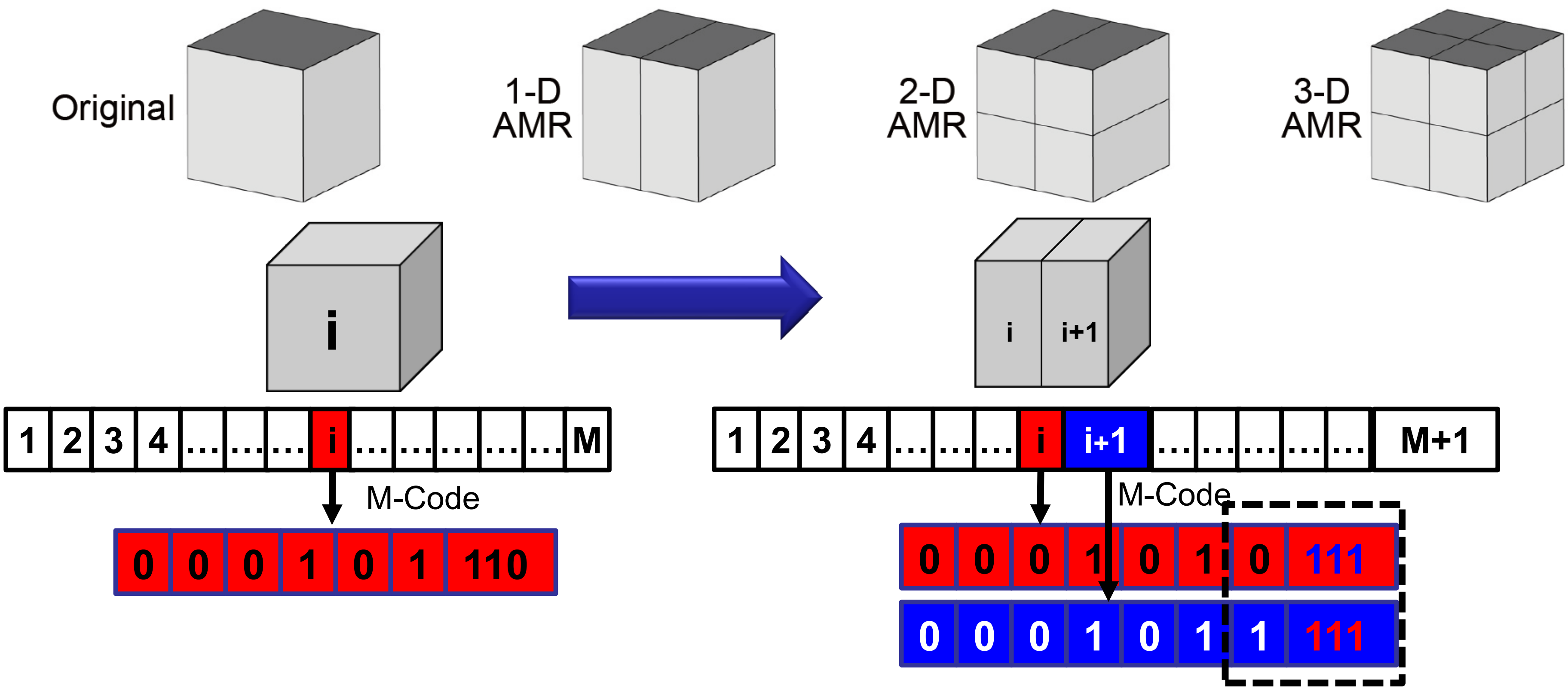
Binary tree/Quadtree/Octree



AMR cell structure

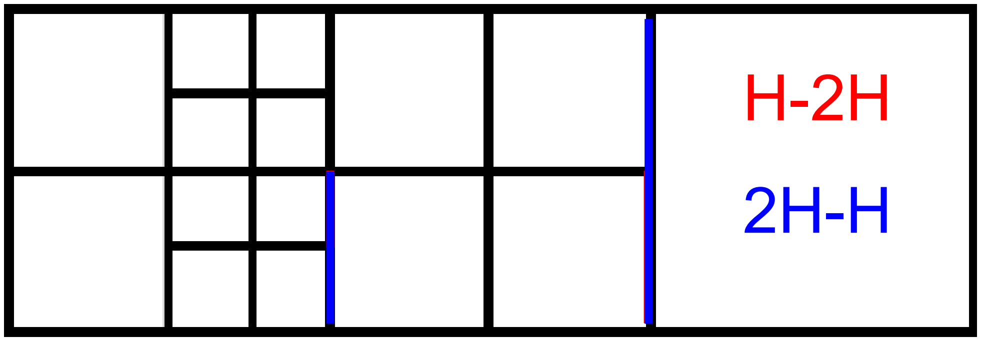
Introduction to the HAMISH Code

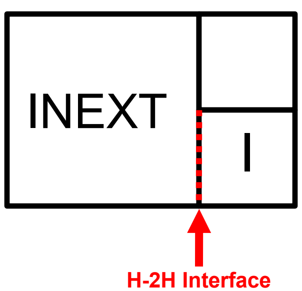
- **Data Structure**



Introduction to the HAMISH Code

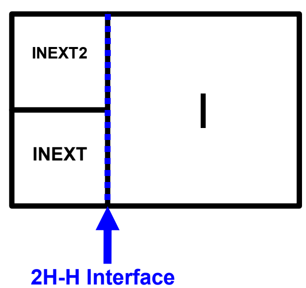
- **Flux Calculation (linear scheme)**





$$f_{Interface} = \frac{1}{2}(f_I + f_{INEXT})$$

$$\left.\frac{\partial f}{\partial x}\right|_{Interface} = \frac{(f_I - f_{INEXT})}{0.5(L_I + L_{INEXT})}$$



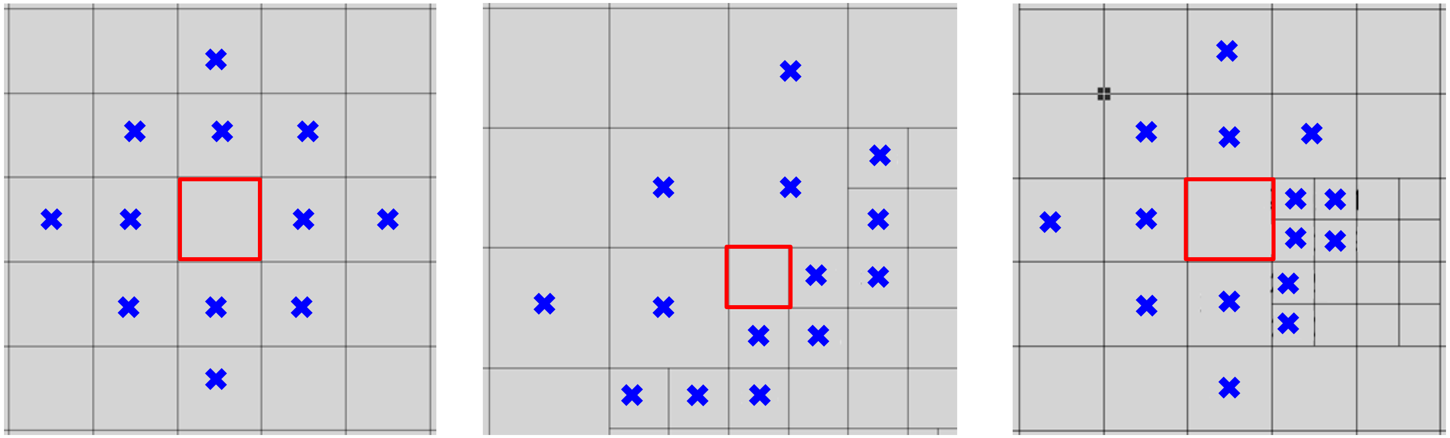
$$f_{Interface} = \frac{1}{2}(f_I + 0.5(f_{INEXT} + f_{INEXT2}))$$

$$\left.\frac{\partial f}{\partial x}\right|_{Interface} = \frac{(f_I - 0.5(f_{INEXT} + f_{INEXT2}))}{0.5(L_I + L_{INEXT})}$$

Conservation is preserved

Introduction to the HAMISH Code

- **Flux Calculation (CENO Flux Reconstruction)**



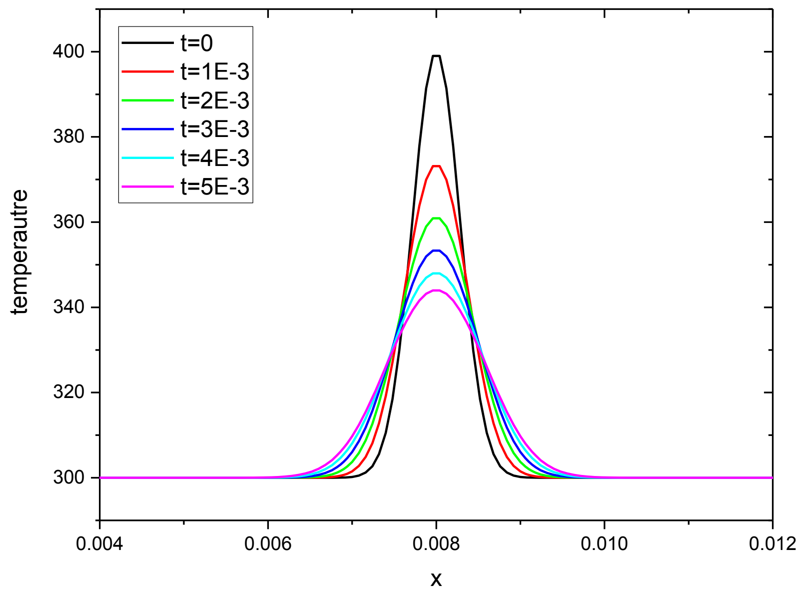
Stencil in 2D case

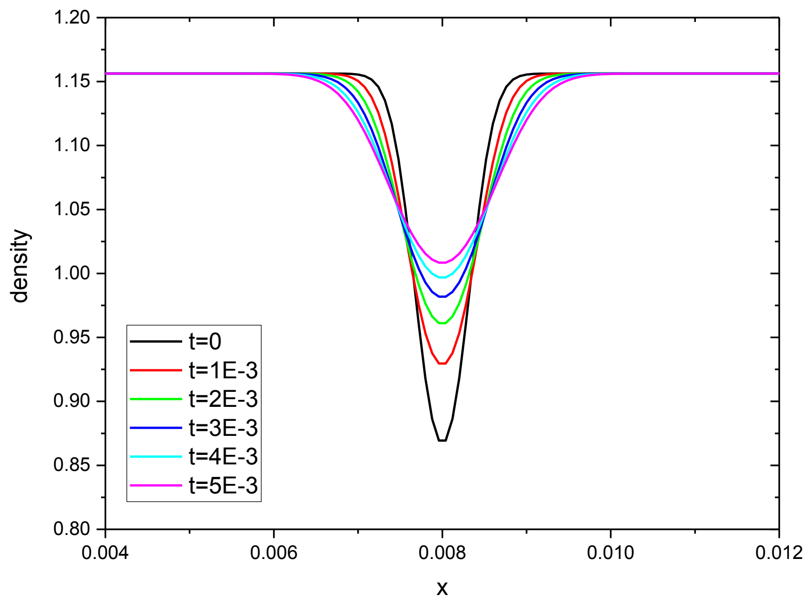
Code Tests

- **1-D thermal diffusion problem**

•Pure Diffusion Case
•Periodic Condition for All Boundaries
•No Chemical Reaction
•Initial condition

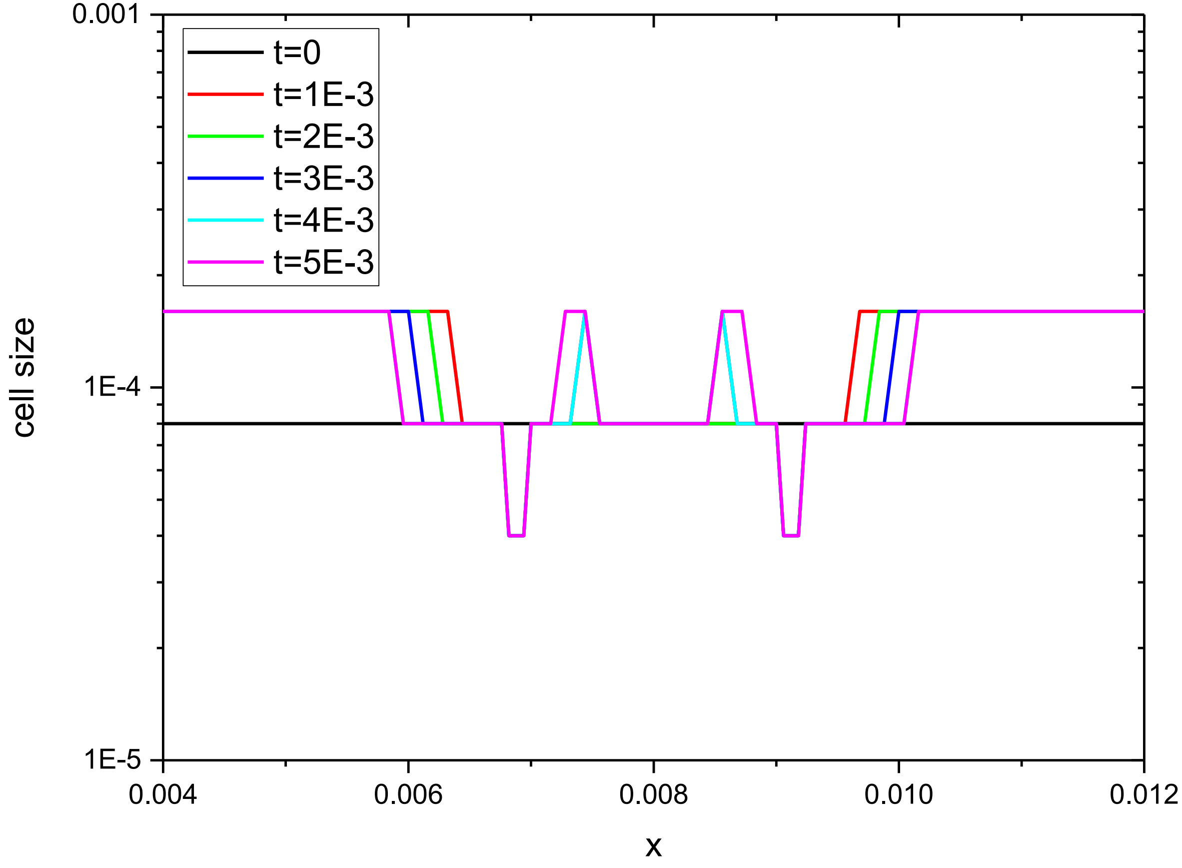
$$T = 300 + 100exp\left(-\frac{(x - x_0^2)}{4\delta}\right)$$





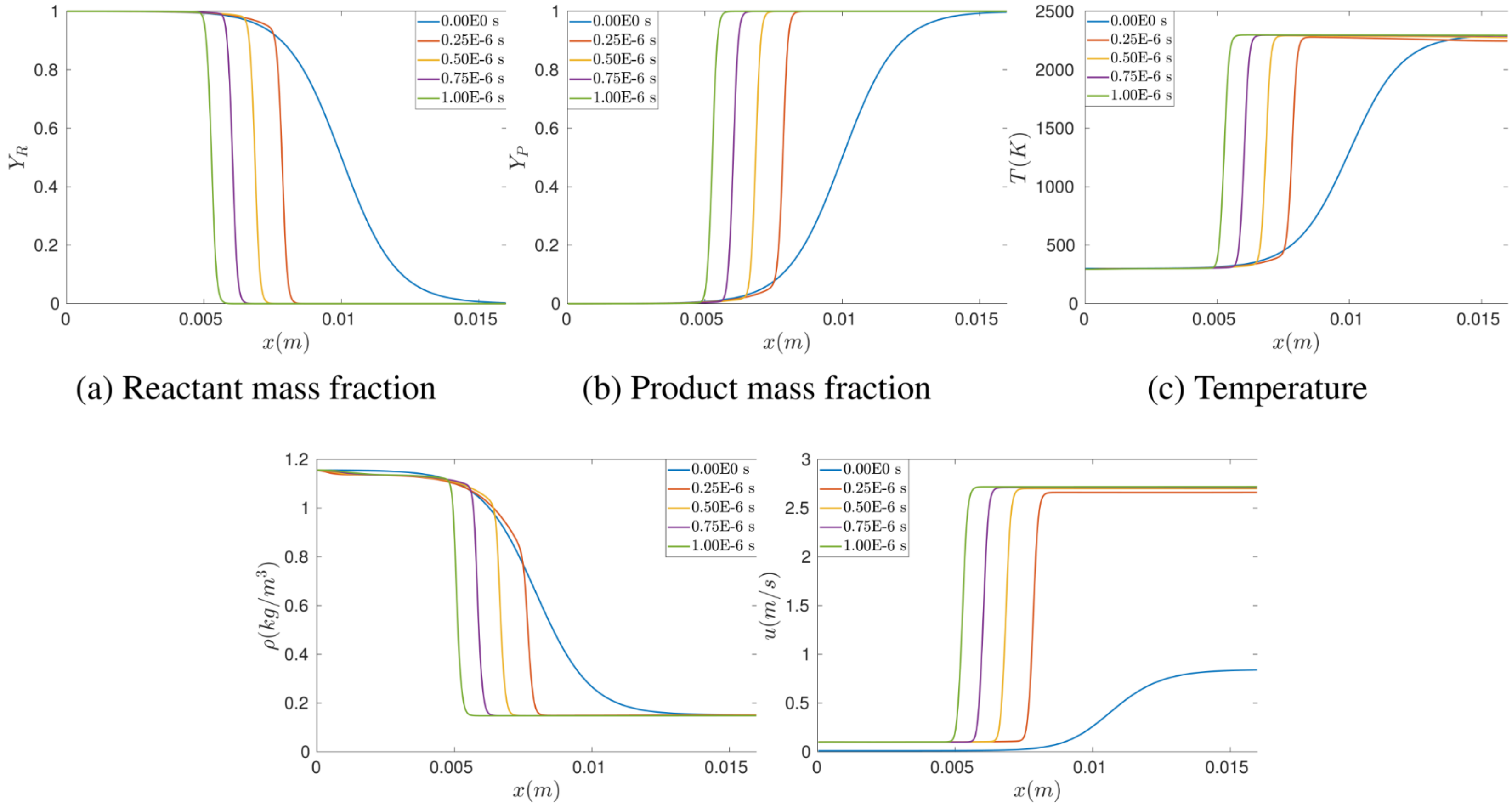
Code Tests

- **1-D thermal diffusion problem**



Code Tests

- **1-D planar flame results**



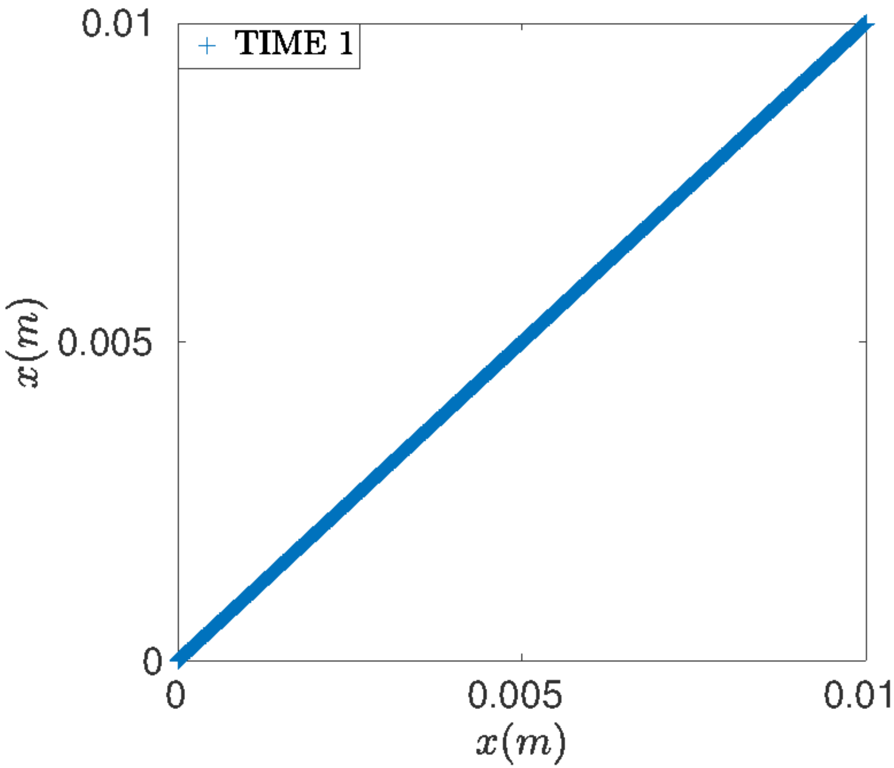
(a) Reactant mass fraction (b) Product mass fraction (c) Temperature

(d) Density (e) Velocity

Code Tests

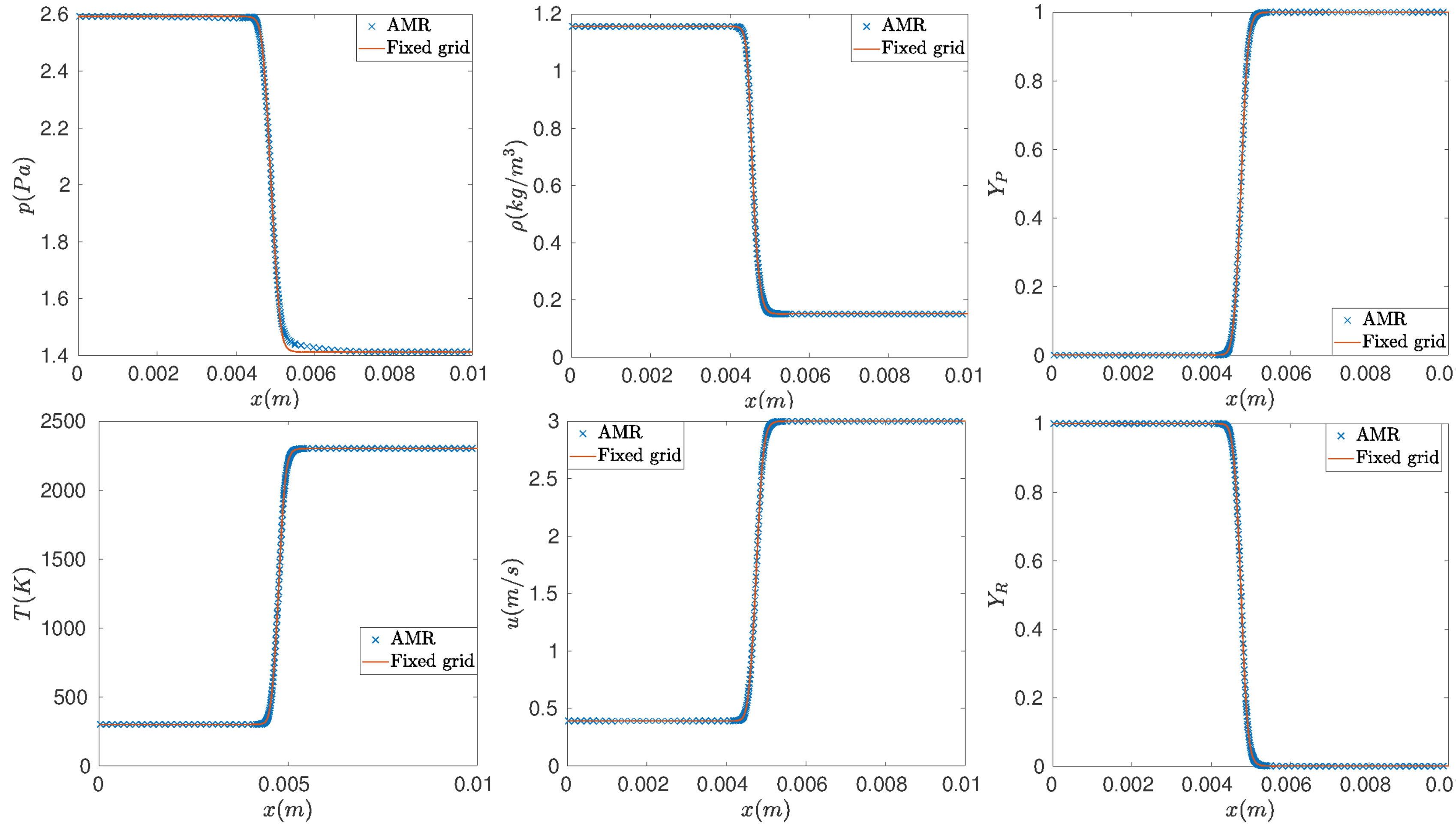
- **1-D planar flame results with AMR**

[redacted]



Code Tests

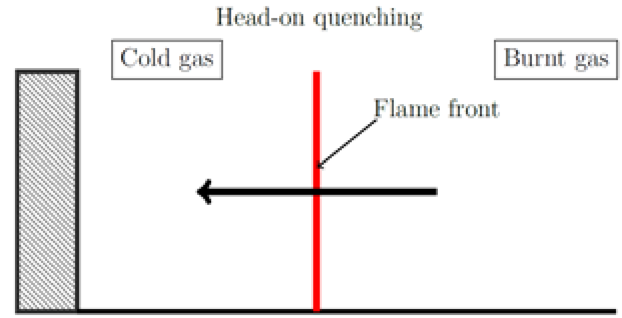
- **1-D planar flame results with AMR**

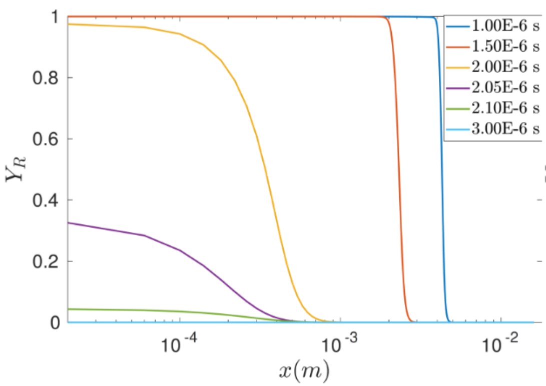


- Fixed grid simulation with 2048 cells.
- AMR simulation finished with 157 cells.

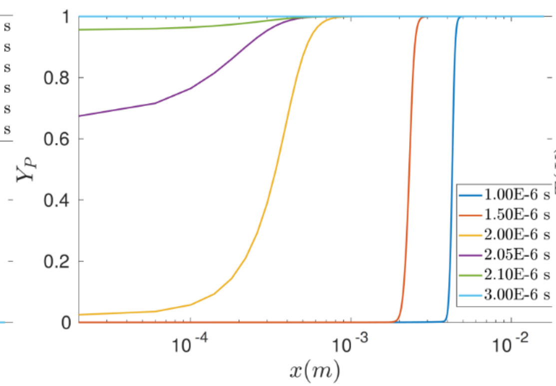
Code Tests

- **1-D HOQ flame results**

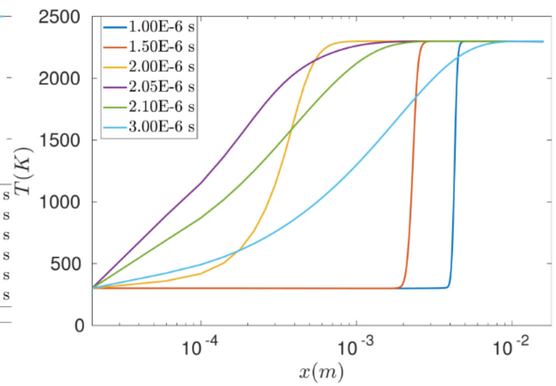




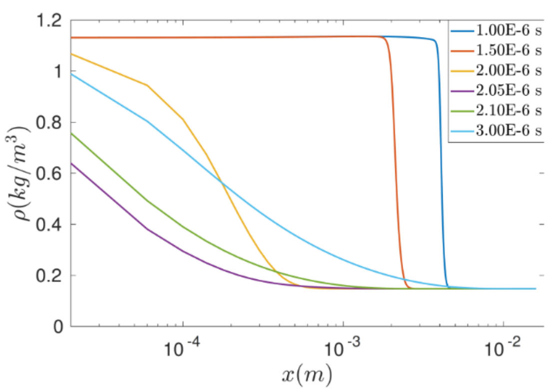
(a) Reactant mass fraction



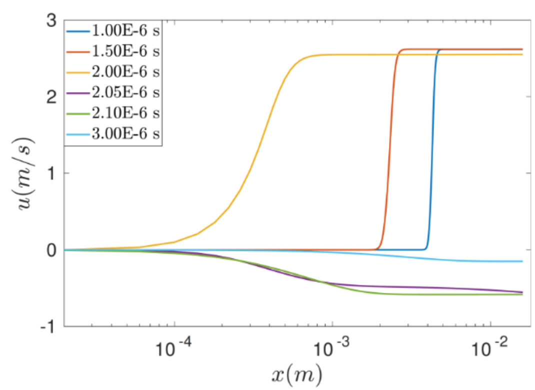
(b) Product mass fraction



(c) Temperature



(d) Density



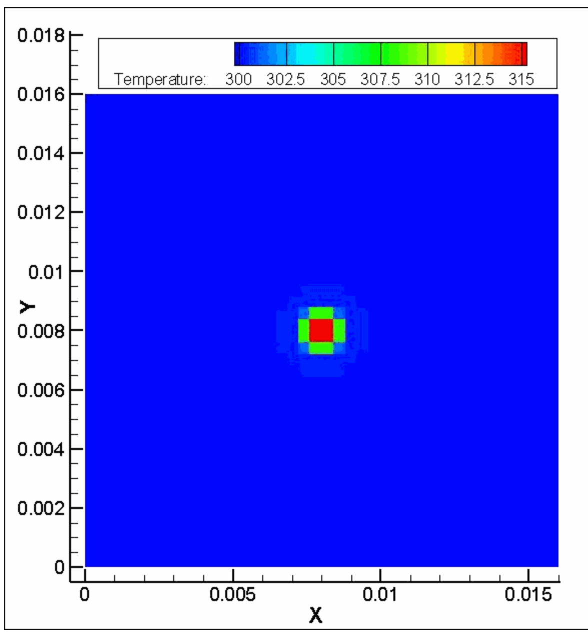
(e) Velocity

Code Tests

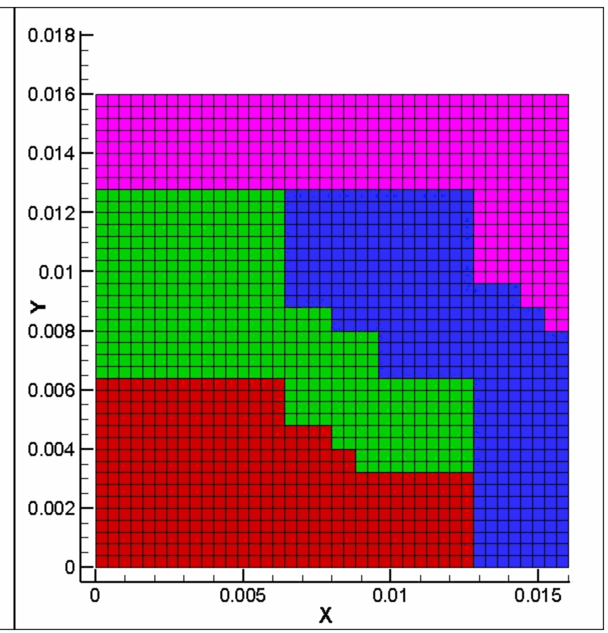
- **2-D thermal diffusion problem**

•Pure Diffusion Case
•Periodic Condition for All Boundaries
•No Chemical Reaction
•Initial condition

$$T = 300 + 100exp\left(-\frac{(x - x_0^2) + (y - y_0^2)}{4\delta}\right)$$



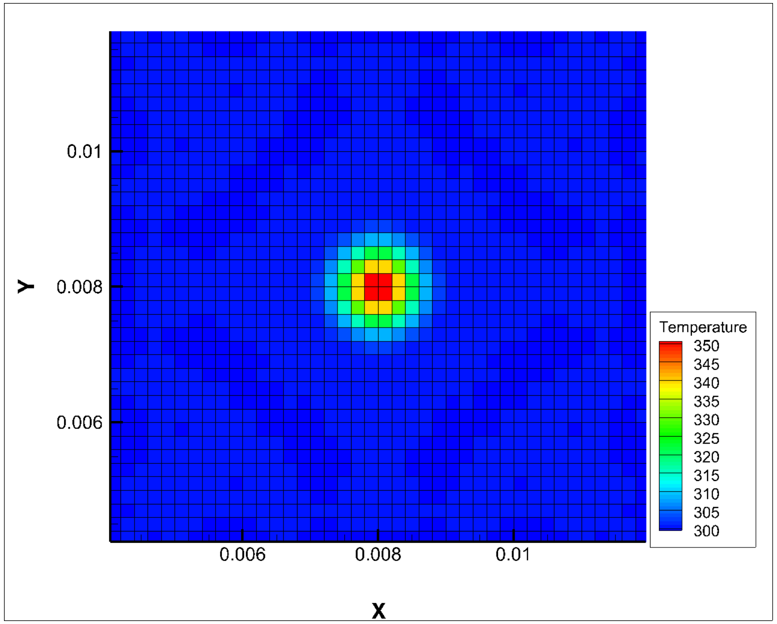
Temperature



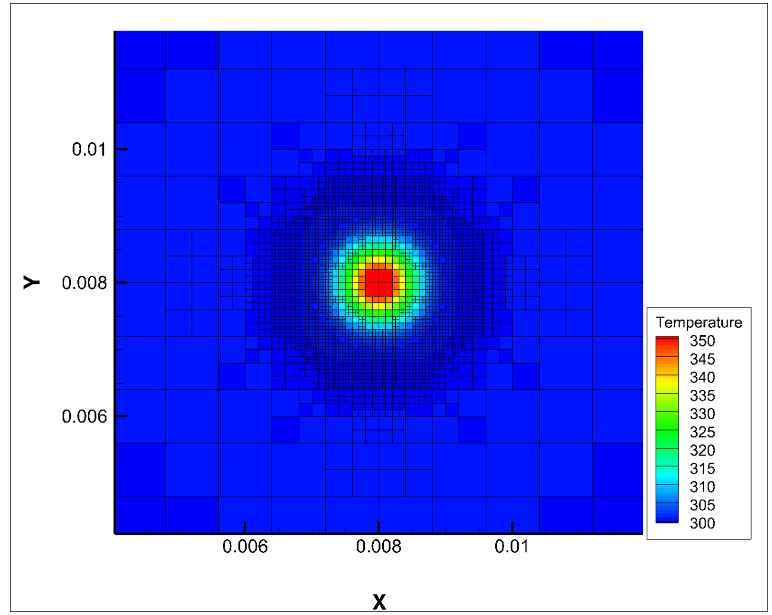
Dynamic load balance

Code Tests

- **2-D thermal diffusion problem**



HAMISH with Fixed Mesh of **6400** cells

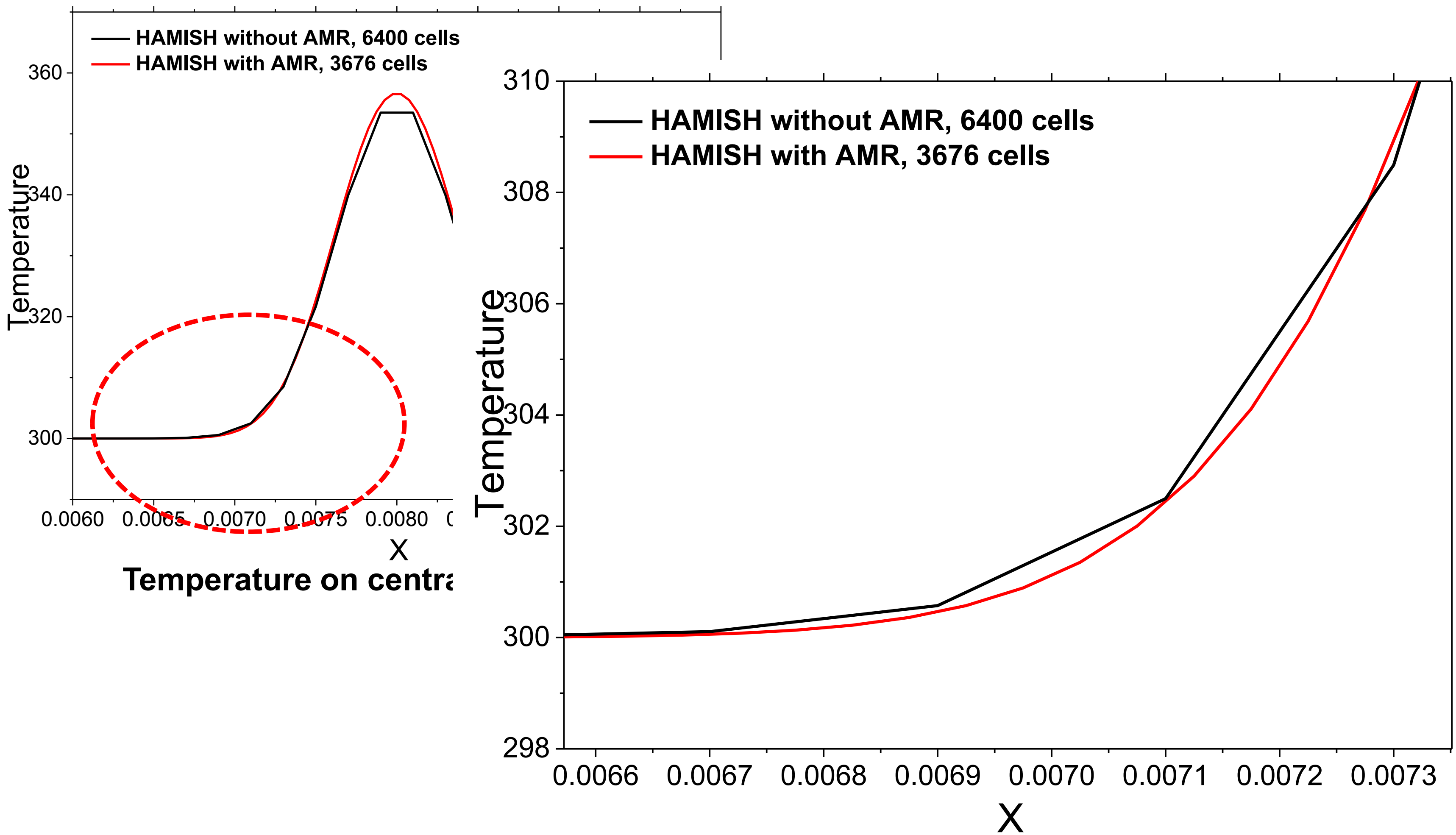


HAMISH with Adaptive Mesh of **3676** cells

Temperature at t=0.001

Code Tests

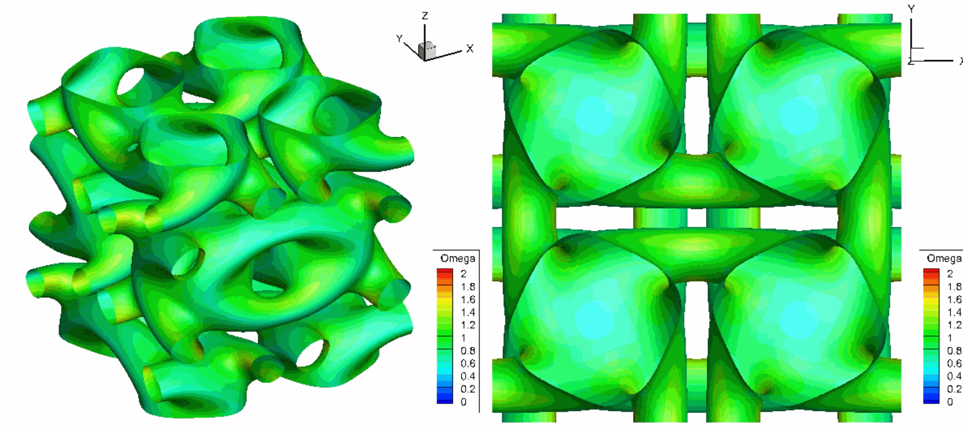
- **2-D thermal diffusion problem**



Code Tests

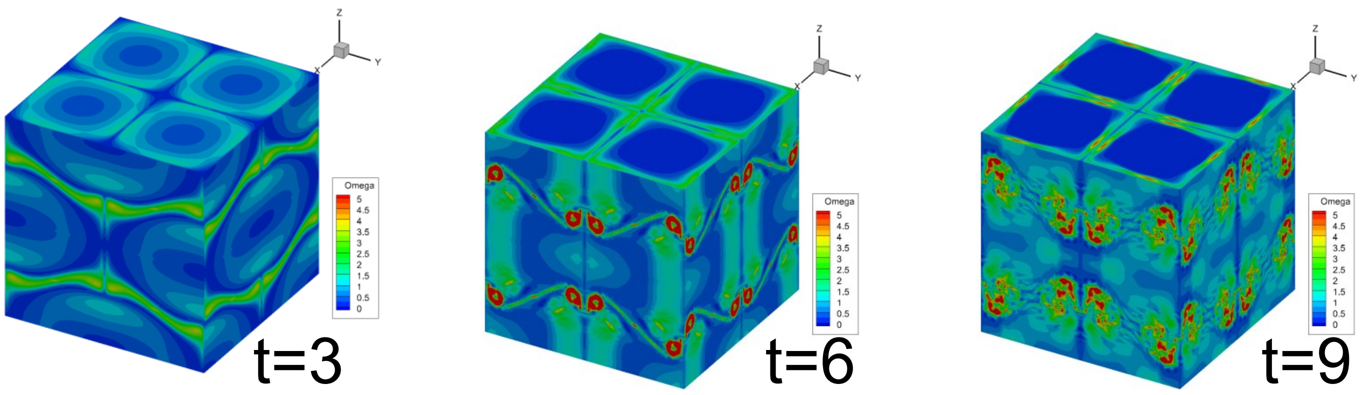
- **3-D Taylor-Green Vortex**

$$u = U_0 \sin(x/L)\cos(y\,/\,L)\cos(z\,/\,L)$$
$$v = -U_0 \cos(x/L)\sin(y\,/\,L)\cos(z\,/\,L)$$
$$w = 0$$
$$p = p_0 + \frac{\rho_0 U_0^2}{16}[\cos(2x/L) + \cos(2y/L)][\cos(2z\,/\,L) + 2]$$
$$\rho = \rho_0$$
$$T = \frac{p}{\rho R}$$



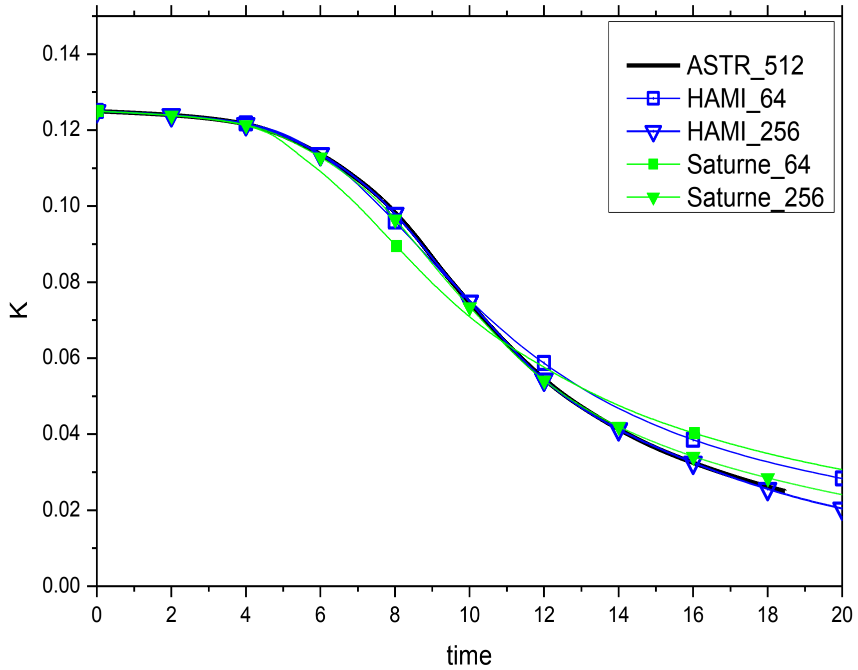
$2\pi \times 2\pi \times 2\pi$, Re=1600, Ma=0.1

J. R. Bull and A. Jameson. "Simulation of the Taylor–Green Vortex Using High-Order Flux Reconstruction Schemes", AIAA Journal, Vol. 53, No. 9 (2015), pp. 2750-2761.

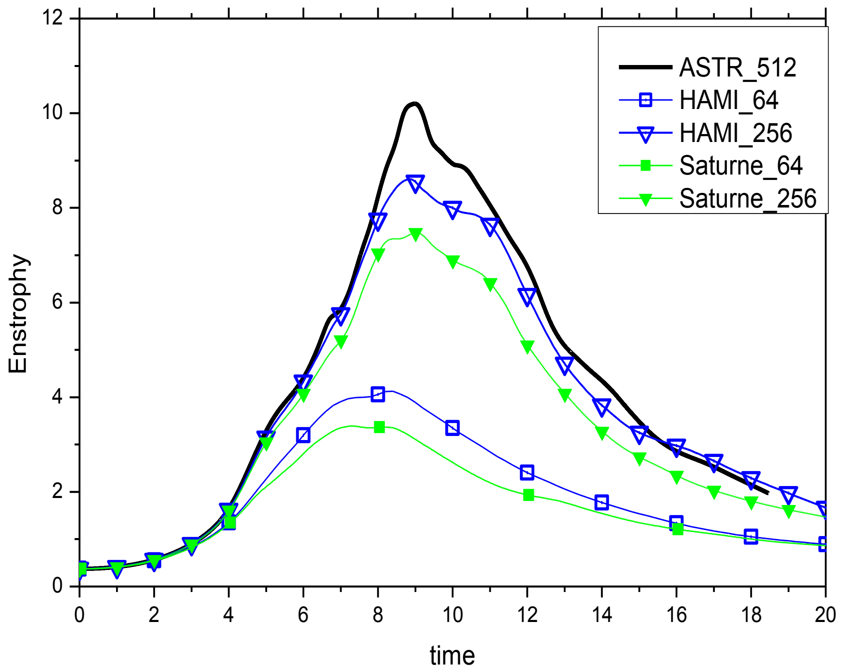


Code Tests

- **3-D Taylor-Green Vortex**



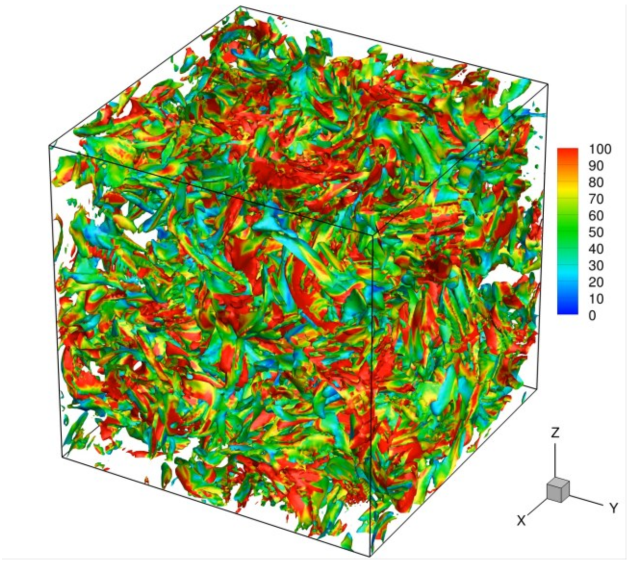
KE temporal evolution



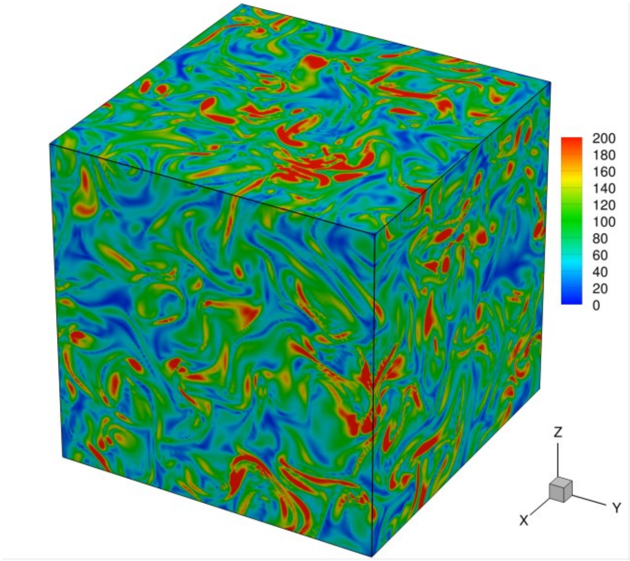
Enstrophy temporal evolution

Code Tests

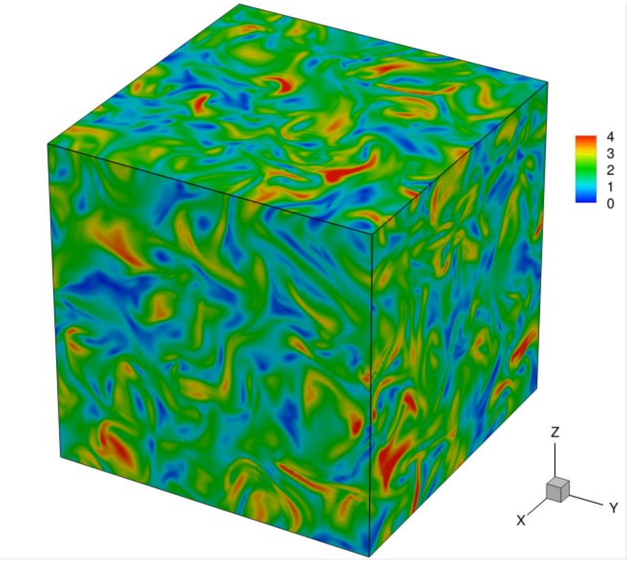
- **3-D Isotropic decaying grid turbulence**



Q-Criterion



Vorticity magnitude



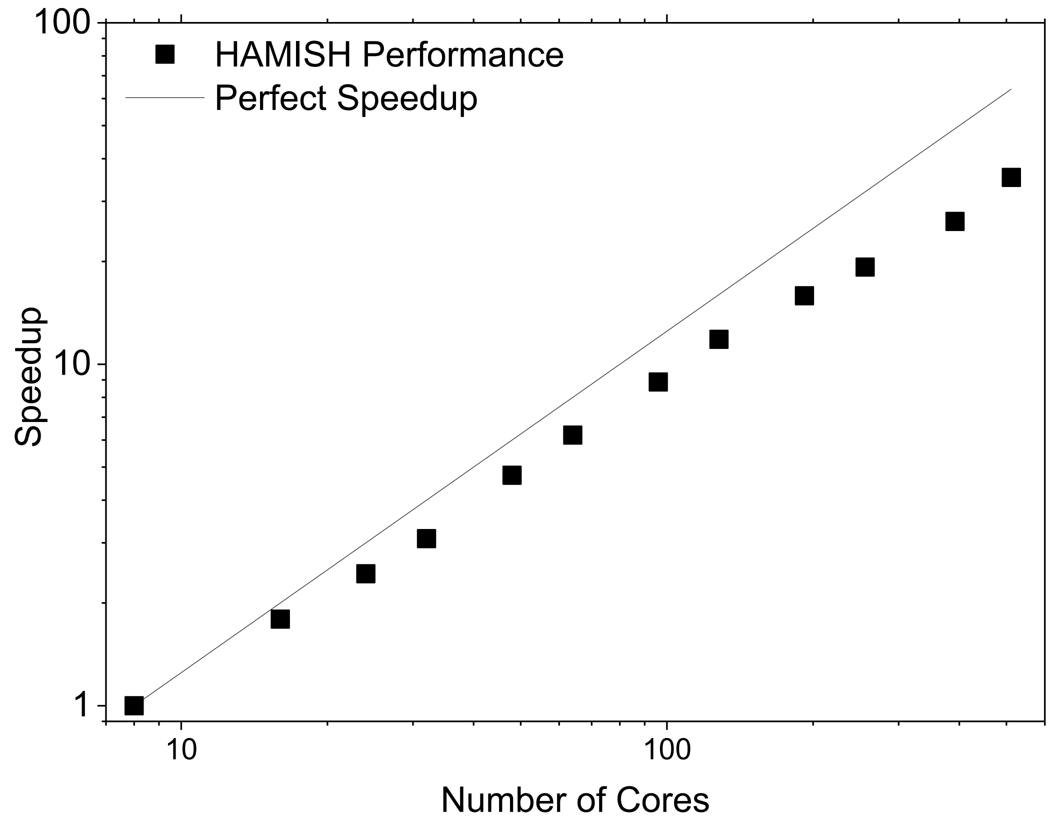
Velocity magnitude

- Fixed grid 128x128x128
- The choice of AMR criterion needs to be adjusted.
- Currently enstrophy based AMR criterion is being tested.

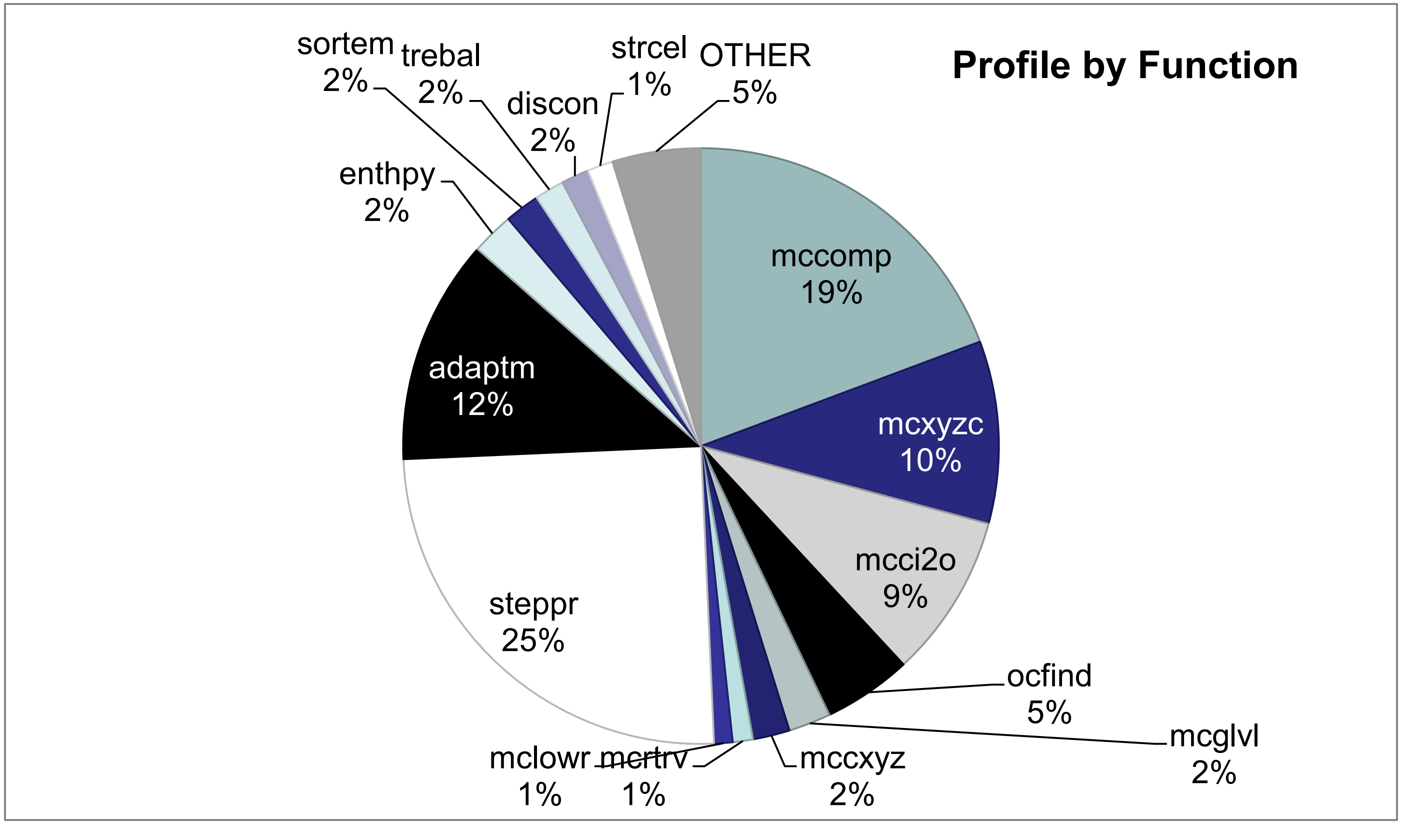
Scalability and Code Profiling

- Scalability of HAMISH with AMR off (128^3 cells)

Cores	Runtime / s	Speedup
8	1632.8	1.00
16	910.6	1.79
24	670.5	2.44
32	529.1	3.09
48	345.1	4.73
64	263.2	6.20
96	184.0	8.87
128	138.0	11.83
192	102.9	15.87
256	84.82	19.26
392	62.3	26.20
512	46.3	35.26



Scalability and Code Profiling



More than 60% of the computations of one step is cost for M-Code, Octree, AMR related subroutines.

MCCOMP	Compares two Morton codes in their entirety
MCXYZC	Converts xyz coordinates into a Morton code at the specified level
MCCI2O	Converts an encoded integer array to an octal string
OCFIND	Searches the local Octree using a given Morton code
STEPPR	Time stepping of the solution, including calculating RHS
ADAPTM	Adapts the spatial mesh

Summary and Perspectives

- The HAMISH code was tested and its accuracy was assessed.
- Good performance and scalability were observed.
- Adaptive mesh refinement presents a better capability in resolving flame front, using fewer computational cells, compared to a simulation using the fixed mesh-based solver.
- More than 60% of the CPU resource is spent in the mesh adaptation-related computations when AMR is activated at every time-step.
- The current HAMISH code shows its capability in capturing small-scale structures and interfaces in turbulent reacting flows.

- Code optimisation.
- Level set model for two-phase flow.
- Post-processing module for turbulence simulation.
- OpenMP support.

Acknowledgements

The project is supported by the UKCTRF (EP/K024574/1) and EPSRC Software Flagship Project ((EP/P022286/1).)