UNIVERSITY OF CAMBRIDGE

Validation of HAMISH: DNS of Combustion with Adaptive Mesh Refinement

R.S. Cant, U. Ahmed, J. Fang, N. Chakraborty, G. Nivarti, C. Moulinec and D. R. Emerson

Department of Engineering

Acknowledgements

The Hamish team:

Cambridge: Stewart Cant, Girish Nivarti

Newcastle: Umair Ahmed, Nilanjan Chakraborty

Daresbury: Jian Fang, Charles Moulinec, David Emerson

Funding from EPSRC/UKCTRF and Daresbury Laboratory

Early inspiration from Bill Dawes and Simon Harvey

Newcastle University

Science & Technology Facilities Council

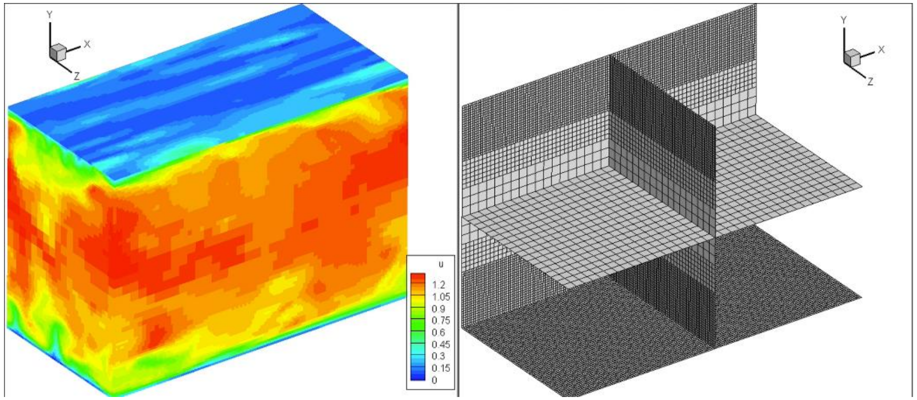
Outline

- Background
- Introduction to the HAMISH Code
- Code testing and validation
  - 1-D laminar flame
  - 2-D channel flow
  - 2-D flame propagation
  - 3-D homogeneous isotropic turbulence
  - 3-D Taylor-Green vortex
  - 3-D turbulent flame
- Accuracy and scalability
- Summary and next steps

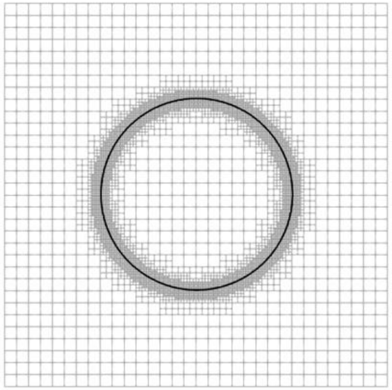
R.S. Cant, U. Ahmed, J. Fang, N. Chakraborty, G. Nivarti, C. Moulinec, D. R. Emerson: "An unstructured Adaptive Mesh Refinement approach for Computational Fluid Dynamics of reacting flows", Journal of Computational Physics, 2022.
doi: https://doi.org/10.1016/j.jcp.2022.111480

Background to AMR

- **Adaptive Mesh Refinement**
  - Dynamic adaption of the mesh
  - Based on the solution
  - Local in space and time

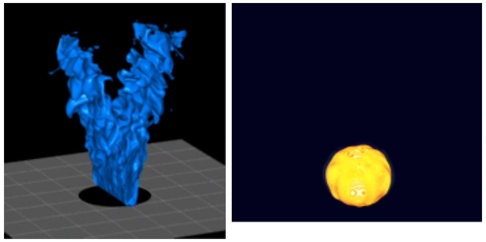


- **Advantages of AMR**
  - Higher accuracy and lower cost compared with a static mesh
  - Savings in both CPU and memory
  - Full control of the local mesh resolution
  - More detailed physics for the same number of cells

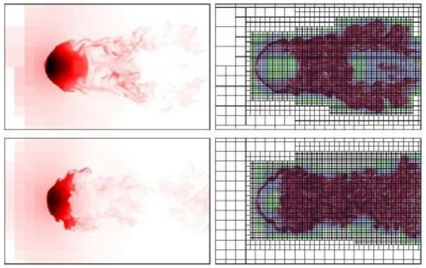


- **Main Applications**
  - Problems with large dynamic range of scales
  - Flames, two-phase flow, boundary layers, shock waves

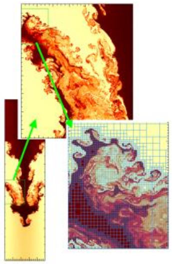
Previous AMR in CFD



Flames (Boxlib)

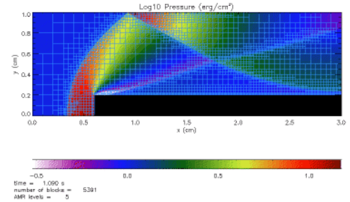


R-T instability (ENZO Code)

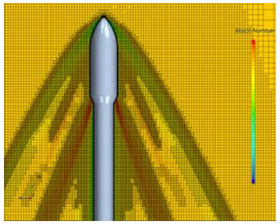


(FLASHCode)

Problems with interfaces

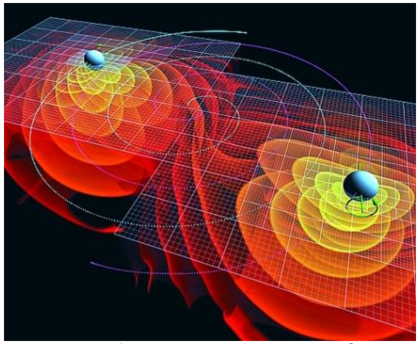


Moving Shock-Wave (PARAMESH)

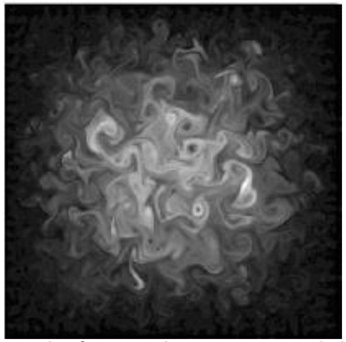


Supersonic Vehicle

Problems with discontinuities

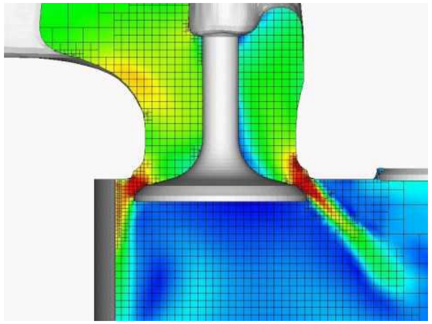


Computing Cosmic Cataclysms

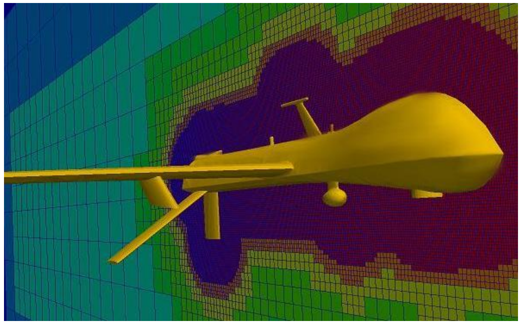


Turbulence (FLASH Code)

Problems with great variety of scales



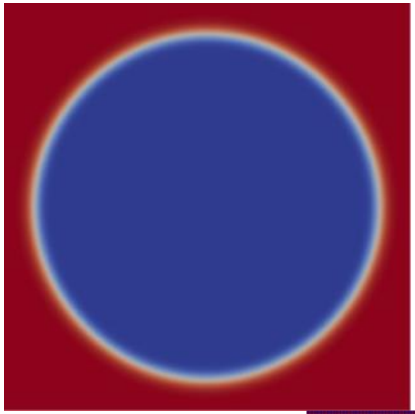
Engine Combustion

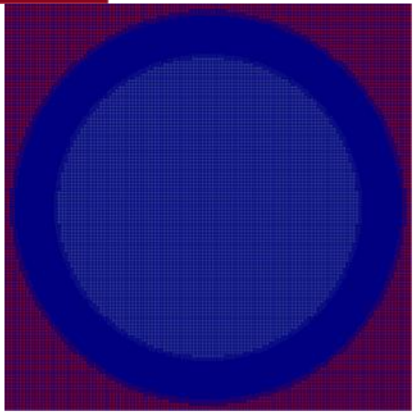


Drone

Problems with complex geometries

So what's new in HAMISH?

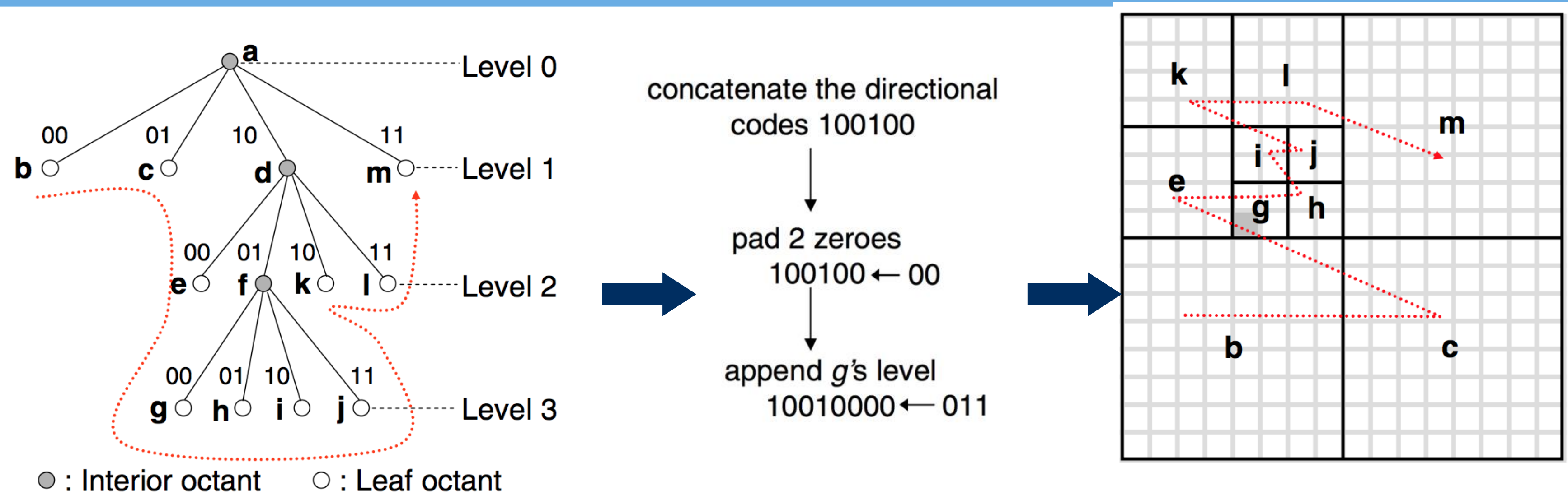




- Pointwise AMR: highly local mesh refinement
- Highly efficient data structure and indexing
- High-accuracy numerics: spatial and temporal
- Fully automatic parallel load-balancing
- Tailored to combustion problems

- Open source code with UK-based development expertise

Data structures in HAMISH



Oct(Quad)tree

Spatial index (Morton code)

Space-filling curve

Refinement criterion based on the Euclidean norm of the local Laplacian

Tree balancing ensures that (at most) h-2h transitions exist

Partition Interval Table stores the highest local Morton code on each processor

RENO scheme

Solution is reconstructed within each cell using polynomial basis functions ϕ

$$u(x,y,z) = \bar{u}_0 + \sum_{k=1}^{K} a_k^{(u)} \phi_k(x,y,z) \qquad \phi_k = \psi_k - \int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \psi_k \, dxdydz$$

Fourth order sweet-spot: monomials ψ are:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
x	x^2	x^3	y	yx	y^2	yx^2	y^2x	y^3	z	zx	zy	z^2	zx^2	zxy	z^2x	zy^2	z^2y	z^3

Integrate over a cell:

$$\bar{u}_j = \bar{u}_0 + \sum_{k=1}^{K} a_k^{(u)} A_{jk} \qquad A_{jk} = \frac{1}{\hat{h}_j}\int_{-\hat{h}_j/2}^{\hat{h}_j/2}\int_{-\hat{h}_j/2}^{\hat{h}_j/2}\int_{-\hat{h}_j/2}^{\hat{h}_j/2} \phi_k(x-\hat{x}_j, y-\hat{y}_j, z-\hat{z}_j)\, dxdydz$$

Solve the linear system:

$$A_{jk} a_k^{(u)} = b_j^{(u)}$$

using Singular Value Decomposition, producing the Moore-Penrose Pseudo-inverse $A_{kj}{}^*$

Note that A_{jk} (and $A_{kj}{}^*$) depend only on the local geometric configuration of the stencil.

Fluxes obtained from the polynomials evaluated at Gauss integration points on each cell face

Fluxes calculated for the same face in adjacent cells reconciled using a Riemann solver

Algorithms

- Time-stepping: 3rd order 3-step TVB Runge-Kutta (Shu+Osher)

  - adaptive time-step using embedded scheme with PI controller

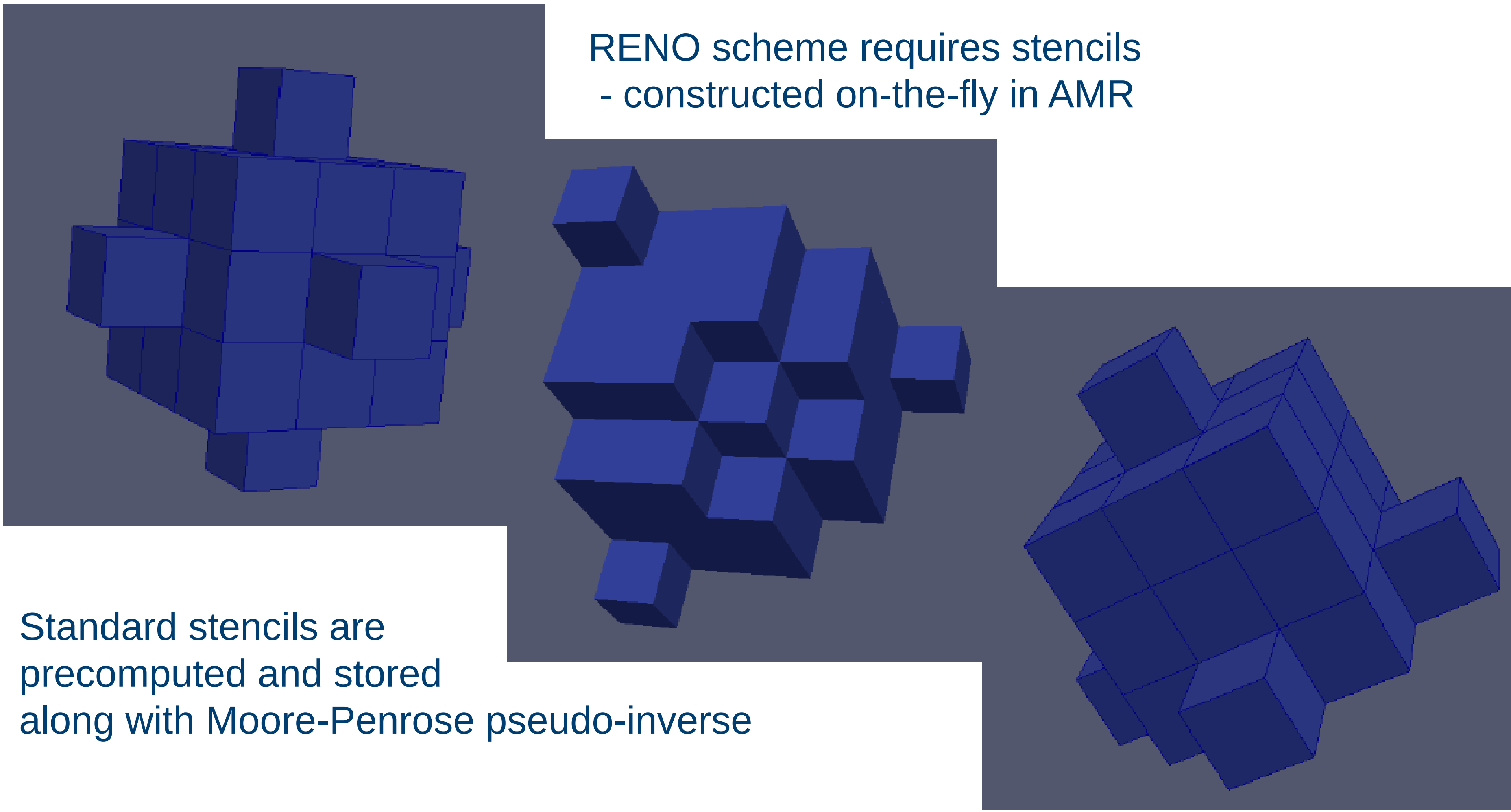
- Automatic parallel load-balancing - cell diffusion approach

$$N_p^{m+1} = N_p^m + \alpha_N\left[\left(N_{p+1}^m - N_p^m\right) - \left(N_p^m - N_{p-1}^m\right)\right]$$

- Stencil construction: layers of face-neighbours with geometrical filtering

- Standard NSCBC boundary conditions

Stencil construction: rogue's gallery



RENO scheme requires stencils
- constructed on-the-fly in AMR

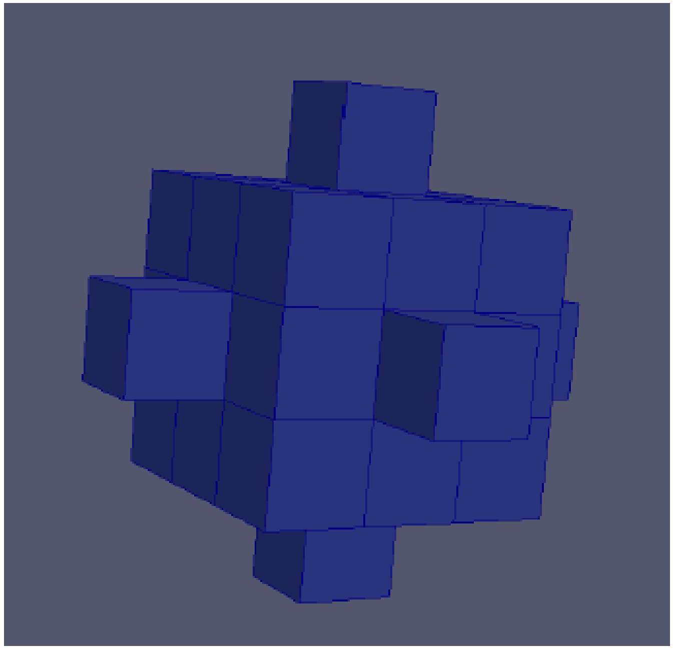
Standard stencils are precomputed and stored along with Moore-Penrose pseudo-inverse

Stencil construction: robustness

Stencil geometry defines the Moore-Penrose Pseudo-Inverse $A_{kj}{}^{*}$

=> determines the coefficients for spatial reconstruction

Largest singular value of $A_{kj}{}^{*}$ (i.e. smallest SV of A_{jk}) controls the stability of the time-stepping scheme



Criterion for robust stencil construction:

SV adjustment by adding cells to stencil
Each cell => new row in A_{jk}
Shifts the SVs in a predictable manner
Improves the stability

Testing: 1D laminar flame propagation

Stoichiometric methane-air flame

- 25-step chemistry
- AMR based on

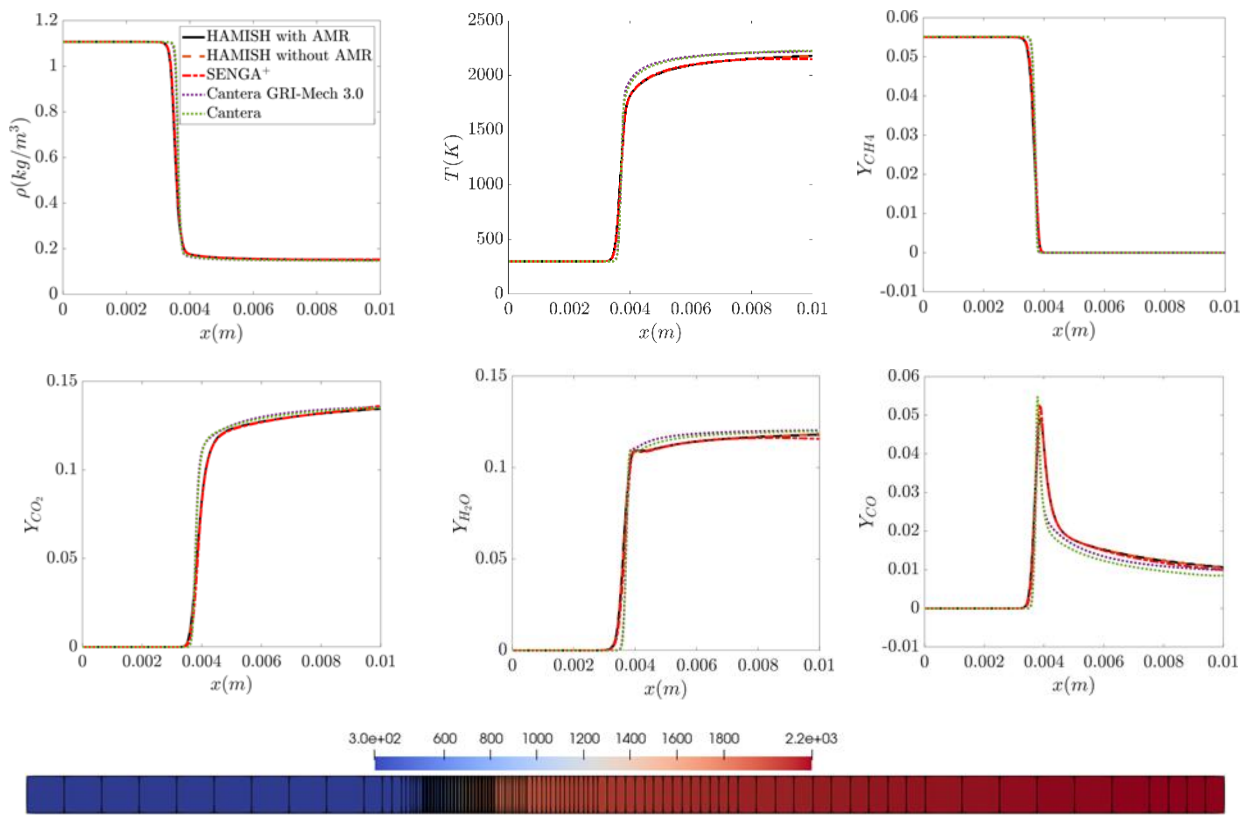
  temperature

  mass fractions

- Comparisons with

  SENGA+

  Cantera



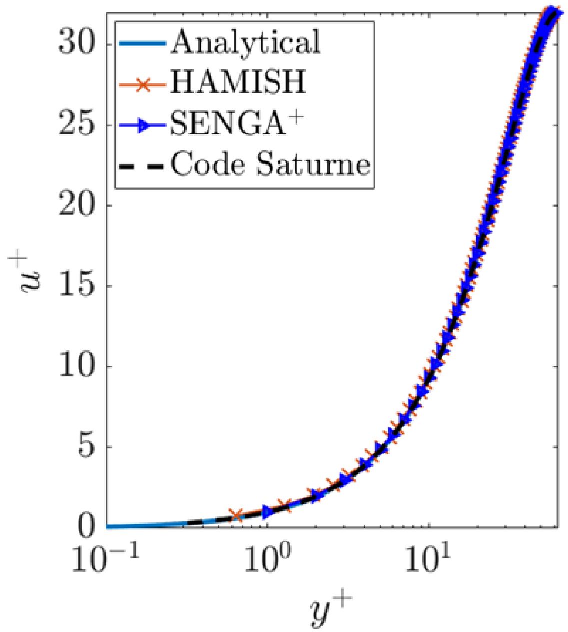
Testing: 2D laminar channel flow

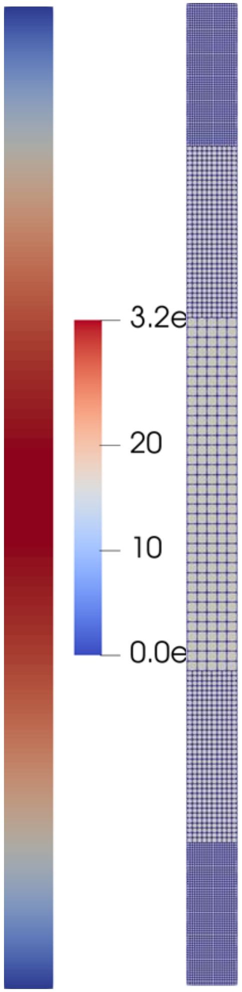
2D channel, $Re_\tau = 64$

- Non-reacting laminar flow
- No-slip BCs on top and bottom walls
- AMR based on velocity
- Comparisons with

  SENGA+

  Code Saturne





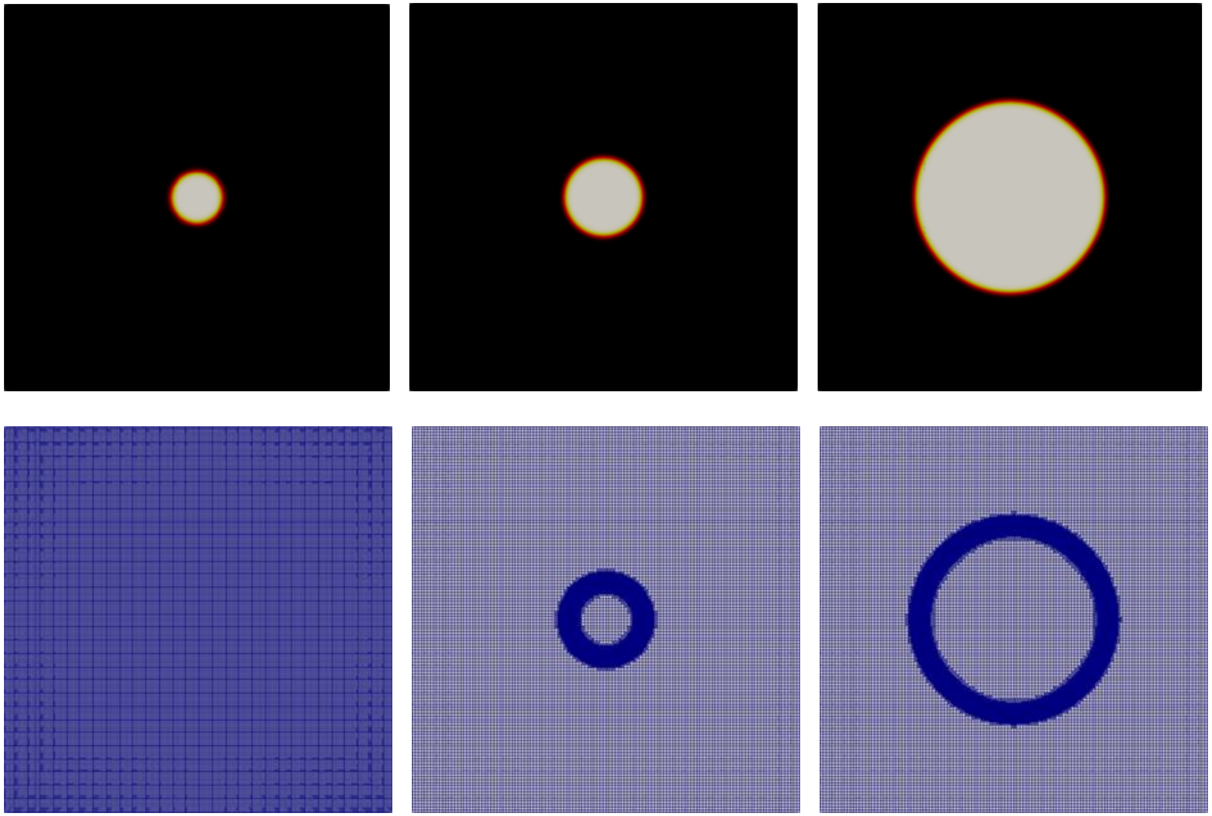
Testing: 2D circular flame propagation

2D circular outwardly-propagating laminar flame

- single-step chemistry
- AMR, initial mesh 64^2

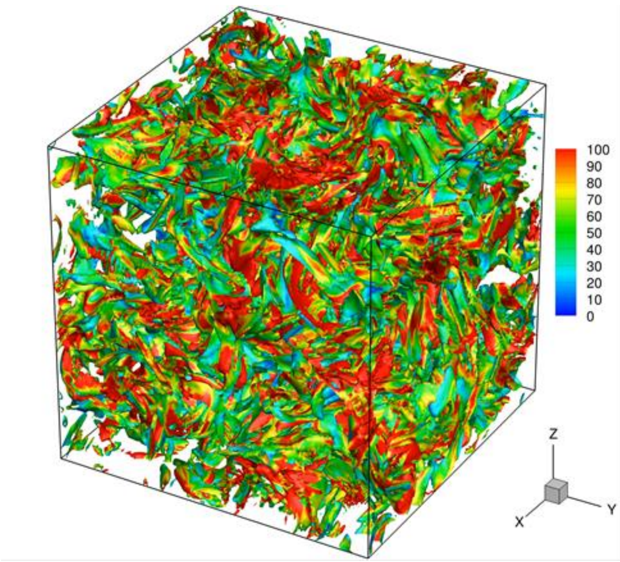
Progress variable:

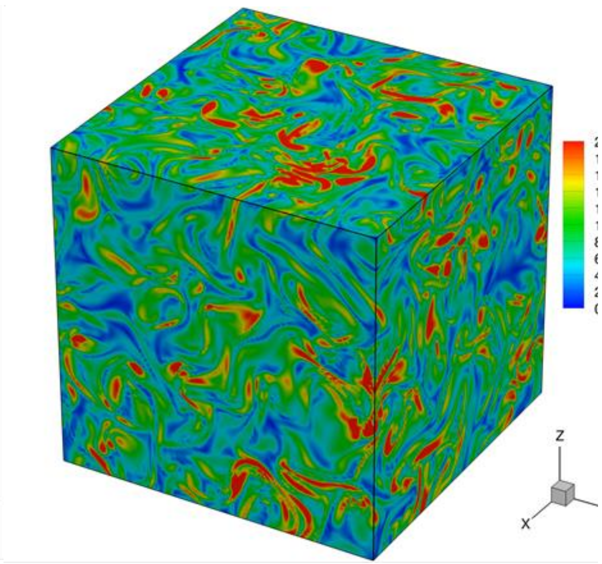
Adapted mesh:

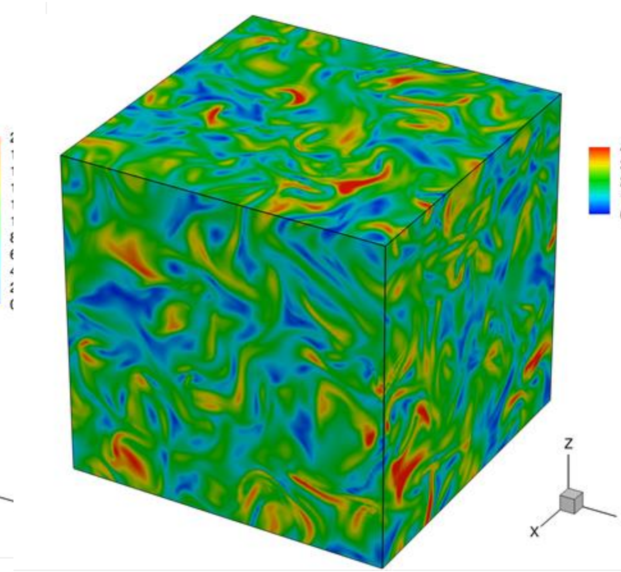


Testing: 3D homogeneous isotropic turbulence

Fixed mesh 128^3







Q-Criterion

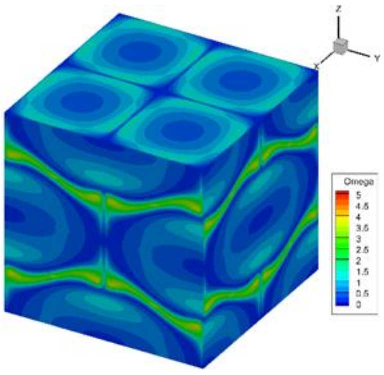
Vorticity magnitude

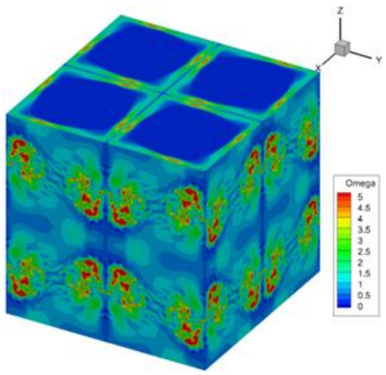
Velocity magnitude

Testing: 3D Taylor Green vortex case

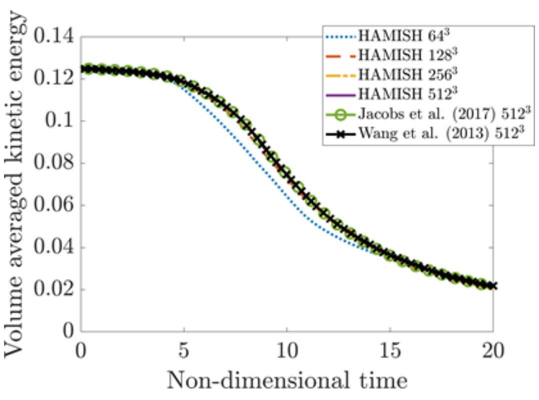
3D TGV initialised with

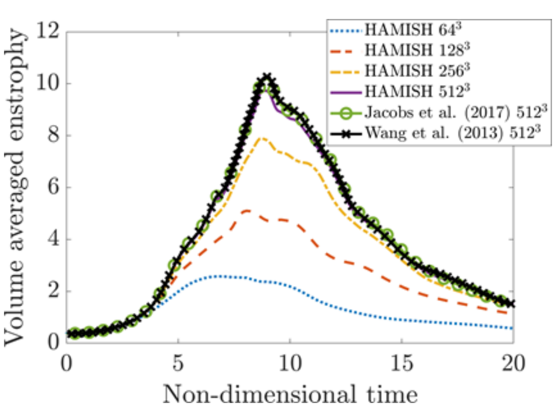
$$u = U_0 \sin(x/L)\cos(y/L)\cos(z/L)$$
$$v = -U_0 \cos(x/L)\sin(y/L)\cos(z/L)$$
$$w = 0$$
$$p = p_0 + \frac{\rho_0 U_0^2}{16}\left[\cos(2x/L) + \cos(2y/L)\right]\left[\cos(2z/L) + 2\right]$$
$$\rho = \rho_0$$
$$T = \frac{p}{\rho R}$$





- Re = 1600, M = 0.1
- cubic domain
- Mesh 64^3 up to 512^3
- No AMR

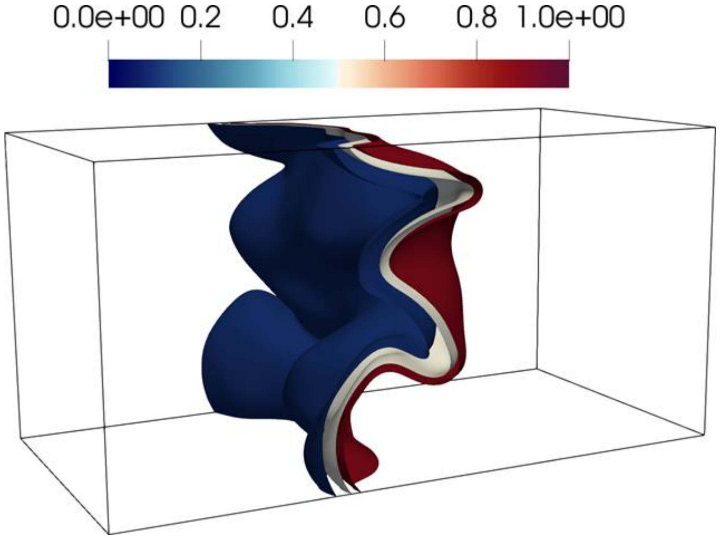


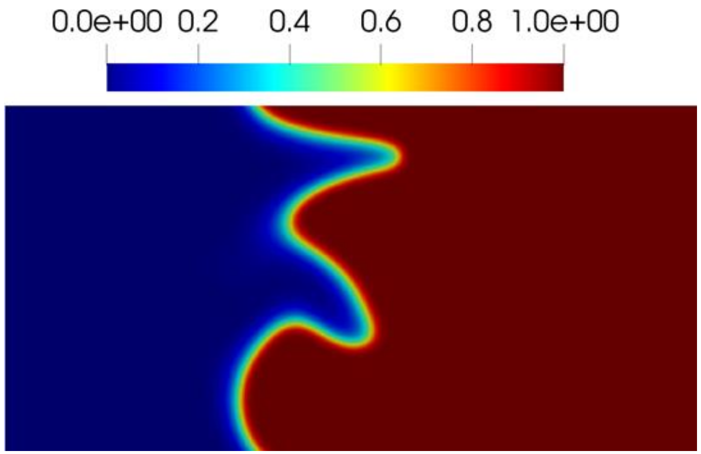


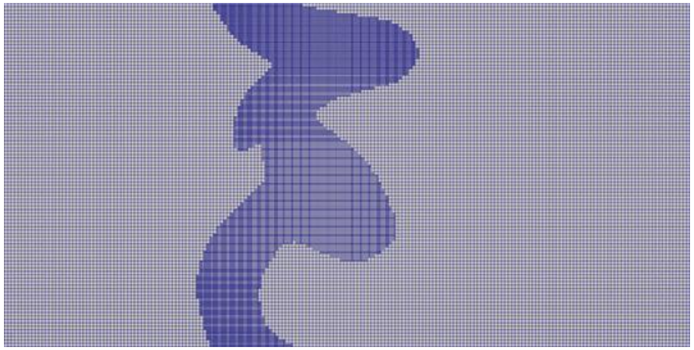
Testing : 3D turbulent flame

3D statistically planar flame

- single-step chemistry
- turbulent flow field u'/s_L=5.0
- AMR: initial mesh 16m cells
  final mesh ~4m cells





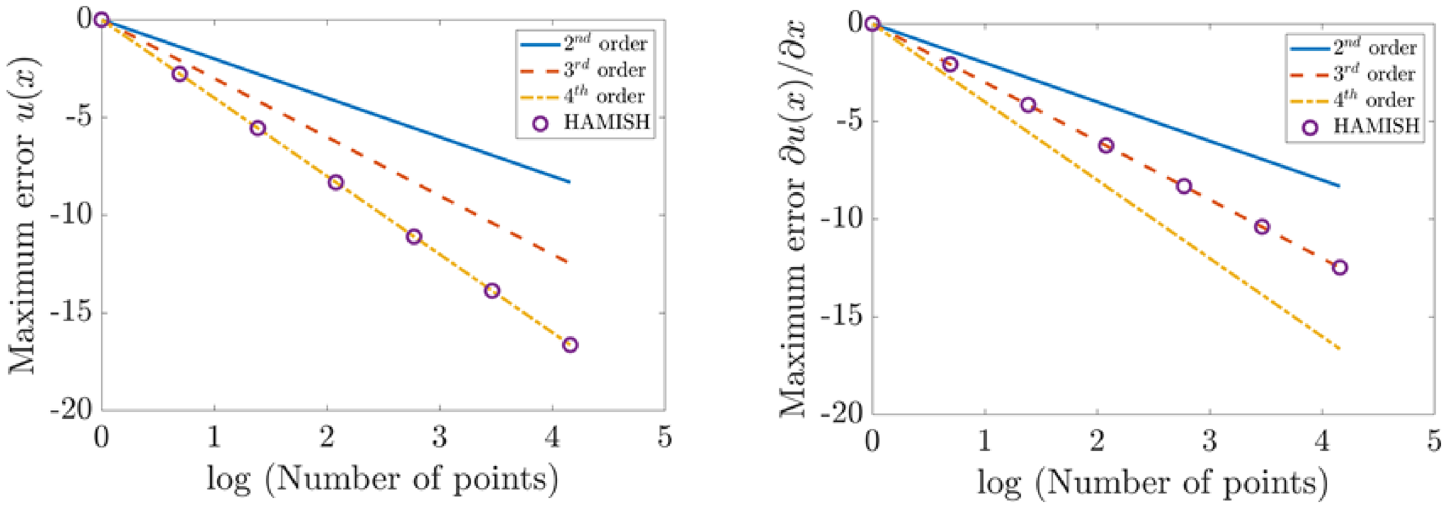


Accuracy

Reconstruction of prescribed function

f(x) = sin*(2πx/L)* and its (numerical) first derivative

Function is 4th order, derivative 3rd order: consistent with RENO scheme



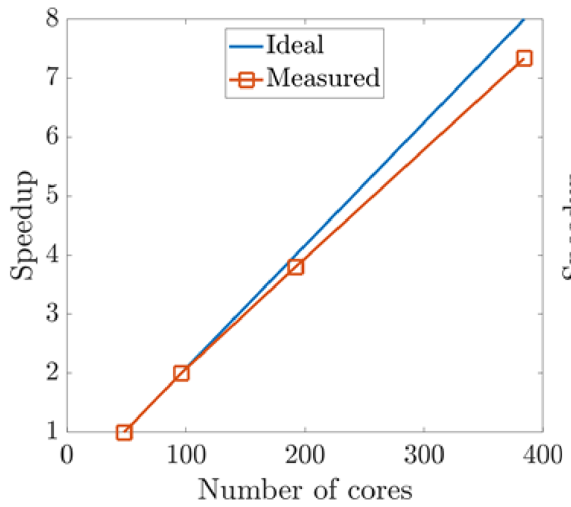
Parallel scaling

3D planar laminar stoichiometric methane-air flame

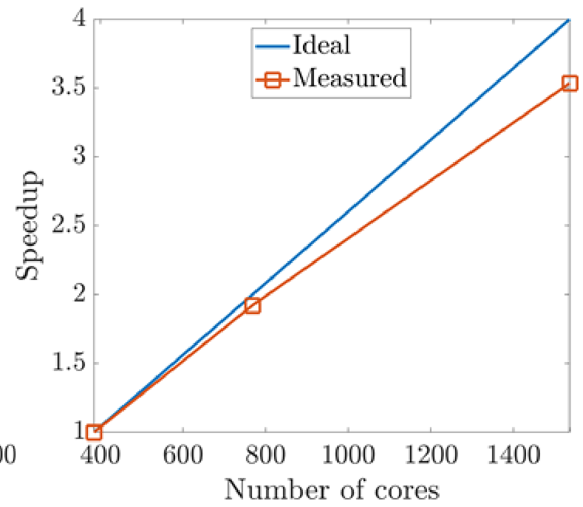
- 25-step chemistry
- cuboidal domain
- no AMR

  mesh 1: 512×128^2

  mesh 2: 1024×256^2
- parallel efficiency 85.6% up to 1536 cores (so far)



mesh 1

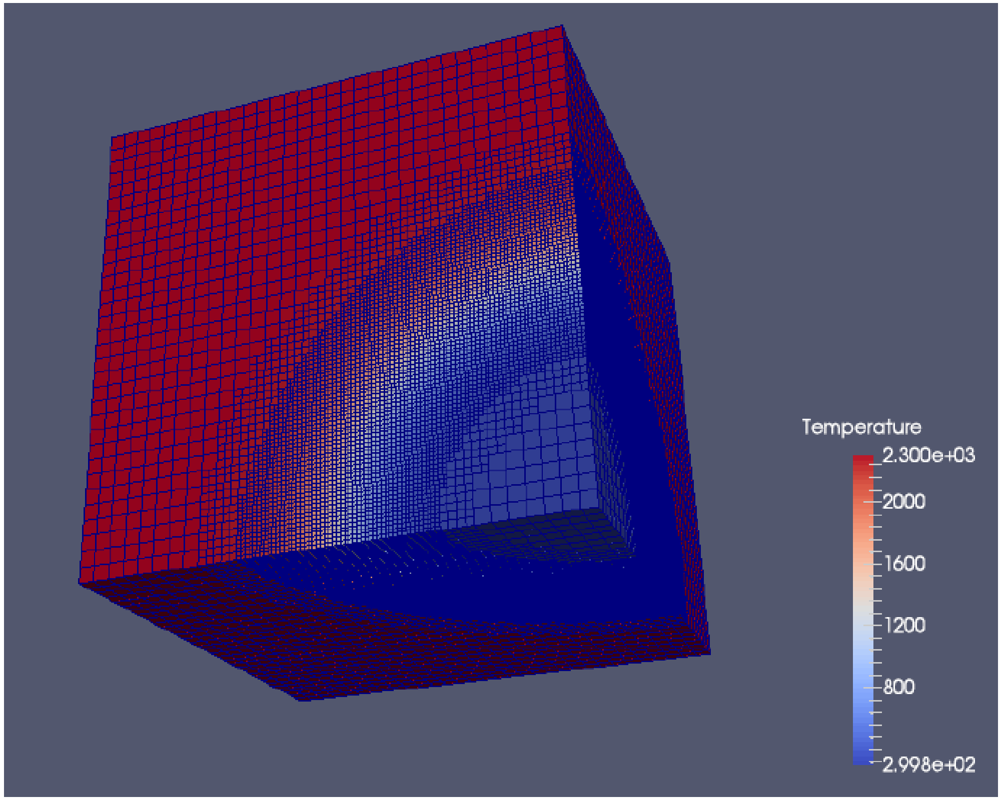


mesh 2

Summary

HAMISH code for reacting flow DNS with AMR

- 4^{th} order accurate in space
- 3^{rd} order adaptive RK in time
- Multi-step chemistry/transport
- AMR with scalars or velocity
- Good range of test cases
- Good parallel efficiency



Next steps

Done: HDF5 parallel I/O

Code tested and running well on ARCHER2

Ongoing: Further test cases: turbulent flame-wall interaction

studies of thermodiffusive instabilities

To come:

Two-phase flow

High-speed flow

Immersed boundaries