rhoEnergyFoam Tutorial Guide

Davide Modesti[1] Sergio Pirozzoli[2]
[1]Aerodynamics Group, Faculty of Aerospace Engineering,
Delft University of Technology, Kluyverweg 2, 2629 HS Delft, The Netherlands

[2]La Sapienza Università di Roma
Dipartimento di Ingegneria Meccanica e Aerospaziale
Via Eudossiana 18, 00184 Roma

e-mail: d.modesti@tudelft.nl

May 8, 2021

1 Introduction

rhoEnergyFoam is compressible flow solver developed in the OpenFOAM® library. The solver is based on a hybrid numerical scheme, in which an energy conserving numerical flux is coupled with AUSM+up dissipative fluxes to enhance numerical stability and allow shock capturing capabilities. A detailed description of the method can be found in Modesti and Pirozzoli (2017).

2 Compiling

rhoEnergyFoam is available for OpenFOAM® library, version 2.1.1, 2.3.x, and v6, thus it can be compiled by copying the source code in

```
cp -r src/ $WM_PROJECT_DIR/applications/solvers/compressible/rhoEnergyFoam/
```

and invoking wmake in the same directory,

```
cd $WM_PROJECT_DIR/applications/solvers/compressible/rhoEnergyFoam/
```

the command,

```
wmake
```

The code has originally been developed for OpenFOAM® 2.1.1 and it has been ported to version 2.3.x by Valerio D'Alessandro, Universitá Politecnica delle Marche.

3 Input files

rhoEnergyFoam maintain the same input files as the standard OpenFOAM® solvers, but few additional input parameters have to be specified in the *controlDict* file, which are reported in Tab. 1. The solver can

Parameter	Description	Type
k_u	AUSM+up coefficient	double
k_p	AUSM+up coefficient	double
epsilon	freestream velocity u_∞^2/δ	double
ducLevelPress	shock sensor threshold	double
ducLevelConv	shock sensor threshol	double
pressArtDiff	activate artificial diffusion on pressure	boolean
convArtDiff	activate artificial diffusion on convection	boolean

Table 1: Input parameters for rhoEnergyFoam.

Modes of operations	convArtDiff	convArtDiff
Mode A	false	false
Mode B	true	false
Mode C	true	true

Table 2: Input parameters for the three modes of operation of the solver, as in Tab. 1 (Modesti and Pirozzoli, 2017). Suggested values of the coefficients in Tab. 1 are $k_p = 0.25$, $k_u = 0.75$, $epsilon = u_\infty^2/\delta$, $ducLevelPress = 0$, $ducLevelConv = 0.05$.

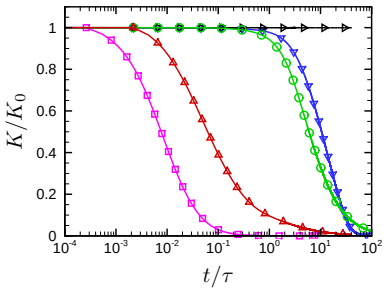

Figure 1: Decaying isotropic turbulence: time evolution of turbulence kinetic energy for *rhoEnergyFoam* in Mode A (right triangles), Mode B (gradient symbols), Mode C with $\theta^* = 0$ (deltas), and for *dnsFOAM* (circles), *rhoCentralFoam* (squares). τ is the eddy turnover time.

be run in three different modes of operation, as described in Modesti and Pirozzoli (2017), setting the input parameters as in Tab. 2. In order to correctly compute the energy conserving part of the numerical flux, the default interpolation scheme in file *fvSchemes* must be *midPoint*, whereas any reconstruction algorithm can be selected for pressure, Mach number, temperature, density and velocity. No time integration scheme need to be specified, as the solver is advanced in time using a low storage Runge-Kutta algorithm.

4 Tutorials

In this Section few tutorials are presented, in different modes of operation, as described in Tab. 1 in Modesti and Pirozzoli (2017).

4.1 Euler turbulence

In this flow tutorial Euler turbulence at initial turbulent Mach number $M_{t0} = 0.1$ is simulated, a detailed description of the flow case can be found in Modesti and Pirozzoli (2017), Section 3.1. The aim of this tutorial is to test the kinetic energy conservation properties of the solver, which is run in Mode A, in which no artificial diffusion is added to the numerical scheme. The mesh can be generate using blockMesh,

```
cd tutorials/hit/

blockMesh
```

Running the solver produces a ASCII file, *diagnostics.dat*, with the turbulence kinetic energy as a function of time. The test case can also be run in Modes B and C to check the amount of numerical dissipation of these modes of operation. Fig. 1 shows the turbulence kinetic energy for the three different modes of operation compared to standard OpenFOAM® solvers.

4.2 Supersonic flow over a forward-facing step

In this tutorial the inviscid supersonic flow over a forward-facing step is simulated at $M_\infty = 3$, a detailed description of the flow case can be found in Modesti and Pirozzoli (2017), Section 3.5. In this case the solver in run in Mode C, with $K_u = 0.25$, $k_p = 0.25$. As a first step the mesh can be generated using blockMesh tool,

```
cd tutorials/forwardStep/
```

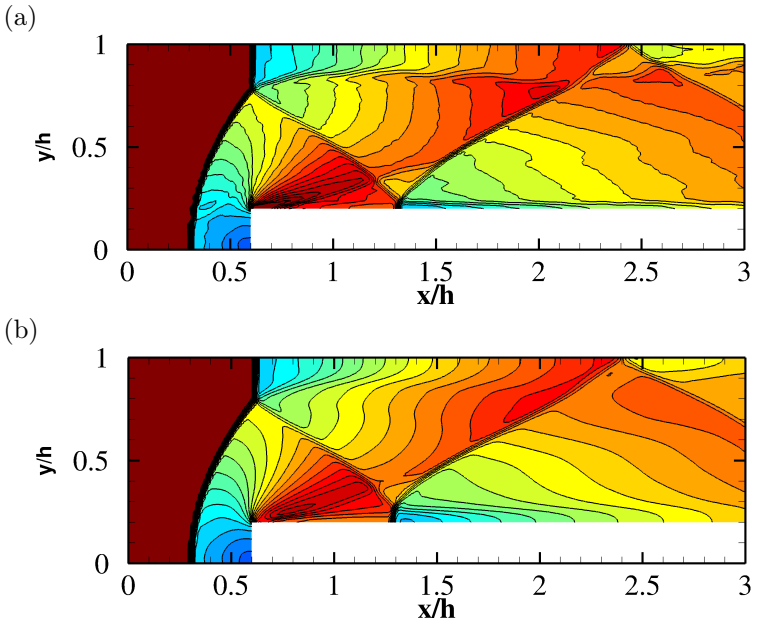

Figure 2: Supersonic flow past forward-facing step at $M_\infty = 3$. 30 Mach number contours are shown in the range $-0.92 \leq M \leq 2.86$ (color scale from blue to red) for *rhoEnergyFoam* (a), *rhoCentralFoam* (b).

`blockMesh`

The Mach number field is reported in Fig. 2, compared to the results obtained using *rhoCentralFoam*.

References

D. Modesti, S. Pirozzoli, A low-dissipative solver for turbulent compressible flows on unstructured meshes, with OpenFOAM implementation, Comput. & Fluids 152 (2017) 14–23.