

# Simulation of 2D Euler equations on different grids

Jérémy PAWLUS

Université de Strasbourg

August 23, 2023

# Table of Contents

- 1 Introduction
- 2 The numerical scheme
- 3 Implementation, simulation and results
- 4 Conclusion

# Preliminary remarks

- My internship took place at the **University of Strasbourg**, in the *Institut de Recherche Mathématique Avancée (IRMA)* from June 12th to August 4th, 2023 under the supervision of **Dr. Andrea THOMANN**.
- It revolved about the implementation and the simulation of the 2D Euler equations on different grids (Cartesian and structured or unstructured triangular meshes).

## About the research center



Figure 1: *Institut de Recherche Mathématique Avancée (IRMA)*



Figure 2: *Institut National de Recherche en Informatique et en Automatique (Inria)*

# About the Euler equations

The Euler equations in two dimensions are the following system of partial differential equations

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_x)}{\partial x} + \frac{\partial(\rho u_y)}{\partial y} = 0 \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u_x^2 + p)}{\partial x} + \frac{\partial(\rho u_x u_y)}{\partial y} = 0 \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u_x u_y)}{\partial x} + \frac{\partial(\rho u_y^2 + p)}{\partial y} = 0 \\ \frac{\partial E}{\partial t} + \frac{\partial(u_x(E+p))}{\partial x} + \frac{\partial(u_y(E+p))}{\partial y} = 0 \end{array} \right. \quad (1)$$

where  $\rho$  is the **mass density** of the fluid (in  $\text{kg/m}^3$ ),  $u_x$  is the **velocity** of the fluid in the **x-direction** (in  $\text{m/s}$ ),  $u_y$  is the velocity of the fluid in the **y-direction** (in  $\text{m/s}$ ),  $p$  is the pressure of the fluid (in  $\text{Pa}$ ) and  $E$  is the **total energy** of the fluid (in  $\text{J/m}^3$ ).

# Recalling the context

The context of this internship is declined in two levels:

- **General:** implementing and simulating the Euler equations on **different grids**, whose interest concern many fields of application in science and engineering.
- **Scientific:** doing so in order to compare the results obtained on different grids and more particularly the **discrepancies in terms of efficiency and diffusive properties** with respect to grids.

# Recalling the objectives

The objectives of this internship is declined in three levels:

- **Academical:** applying knowledge acquired during this year of study and validating the first year of this Master's degree.
- **Scientific:** rectifying undone work from the previous project, generalizing the code for Cartesian grids based on normal vectors then triangular grids, simulating the Gresho vortex test case and studying the performances of the code (errors, CPU time, etc.).
- **Specific:** learning the language Julia and consolidating skills in general programming (debugging, version control, etc.).

# Rewriting the Euler equations

Let us introduce the conservative variables vector  $q \in \mathbb{R}^4$  given by

$$q(x, t) = \begin{bmatrix} \rho(x, t) \\ \rho(x, t)u_x(x, t) \\ \rho(x, t)u_y(x, t) \\ E(x, t) \end{bmatrix}.$$

Then the Euler equations in two dimension (as in the initial Euler equations (1)) can be written as  $q_t + f(q)_x + g(q)_y = 0$ , where  $f$  and  $g$  are the Euler fluxes.



# The Euler fluxes

Herein, the Euler fluxes  $f$  and  $g$  are given by

$$f(q) = \begin{bmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ u_x(E + p) \end{bmatrix} = \begin{bmatrix} q_2 \\ q_2^2/q_1 + p(q) \\ q_2 q_3/q_1 \\ q_2(q_4 + p(q))/q_1 \end{bmatrix} \quad (2)$$

and

$$g(q) = \begin{bmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ u_y(E + p) \end{bmatrix} = \begin{bmatrix} q_3 \\ q_2 q_3/q_1 \\ q_3^2/q_1 + p(q) \\ q_3(q_4 + p(q))/q_1 \end{bmatrix}. \quad (3)$$

Therein,  $u_x$  denotes the velocity in the  $x$ -direction and  $u_y$  the velocity in the  $y$ -direction. All the remaining physical quantities are the same as in the initial Euler equations (1).

# The sound speed and the eigenvalues

The sound speed  $a$  is then defined as:

$$a = \sqrt{\frac{\gamma p(q)}{\rho}}. \quad (4)$$

For the two-dimension Euler equations, the eigenvalues come from the Jacobian of the fluxes  $f$  and  $g$ .

- 1 For the  $x$ -direction, the eigenvalues of the Jacobian of the flux  $f$  are finally given by:

$$\lambda_1 = u_x - a, \quad \lambda_{2,3} = u_x, \quad \lambda_4 = u_x + a, \quad (5)$$

where  $a$  is the sound speed.

- 2 For the  $y$ -direction, the eigenvalues of the Jacobian of the flux  $g$  are finally given by:

$$\lambda_1 = u_y - a, \quad \lambda_{2,3} = u_y, \quad \lambda_4 = u_y + a. \quad (6)$$

# The Euler flux with respect to any normal vector

If we define  $\vec{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix}$  the normal vector of a given edge, the Euler flux adapted to the direction of  $\vec{n}$  can be written as follows:

$$F_{\vec{n}}(q) = F(q) \cdot \vec{n} = \begin{bmatrix} \rho(u_x n_x + u_y n_y) \\ \rho(u_x^2 n_x + u_x u_y n_y) + p n_x \\ \rho(u_x u_y n_x + u_y^2 n_y) + p n_y \\ (E + p)(u_x n_x + u_y n_y) \end{bmatrix} \quad (7)$$

where  $F(q)$  is a tensor containing the Euler flux functions  $f$  (2) and  $g$  (3), respectively for  $\vec{n} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  and  $\vec{n} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ .

# Schematizing the finite volume method for triangular meshes

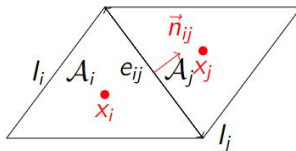


Figure 3: Schematization of the finite volume method for triangular meshes

Herein,  $I_i$  is the cell  $i$ ,  $I_j$  is a neighboring cell of  $I_i$ ,  $e_{ij}$  is the edge between  $I_i$  and  $I_j$ ,  $\vec{n}_{ij}$  (**pointing from  $I_i$  towards  $I_j$** ) is the normal vector of the edge  $e_{ij}$ ,  $A_i$  and  $A_j$  are the areas of  $I_i$  and  $I_j$  respectively, and  $x_i$  and  $x_j$  are the barycenters of  $I_i$  and  $I_j$  respectively.

# Mathematical formulation of the finite volume method for triangular meshes

The finite volume method for triangular meshes updates the state variables vector  $q_i$  on the cell  $I_i$  at time  $t^{n+1}$  from time  $t^n$  with the following formula:

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\mathcal{A}_i} \sum_{j \in \mathcal{V}_i} |e_{ij}| \mathcal{F}_{ij}^n, \quad (8)$$

where  $\mathcal{A}_i$  is the area of the cell  $I_i$ ,  $\mathcal{V}_i$  is the set of **all the cells adjacent to the cell  $I_i$** ,  $|e_{ij}|$  is the length of the edge  $e_{ij}$  and  $\mathcal{F}_{ij}^n$  is the numerical flux at the edge  $e_{ij}$ .

# The Rusanov numerical flux

The expression of the Rusanov numerical flux between two adjacent cells  $I_i$  and  $I_j$  of conservative variables vectors  $q_i$  and  $q_j$  respectively is given by

$$\mathcal{F}_{ij}^{\text{Rusanov}} = \frac{1}{2} \left( F_{\vec{n}_{ij}}(q_i) + F_{\vec{n}_{ij}}(q_j) \right) - \frac{1}{2} \max_{\lambda \in \mathcal{Q}} |\lambda| (q_j - q_i), \quad (9)$$

where  $\mathcal{Q}$  is the set of all the eigenvalues of the Jacobians of the fluxes associated to the conservative variables vectors  $q_i$  and  $q_j$  respectively and given by

$$\mathcal{Q} = \{Sp(f'(q_i)) \cup Sp(g'(q_i)) \cup Sp(f'(q_j)) \cup Sp(g'(q_j))\}.$$

# The HLL numerical flux

In a similar fashion, the expression of the HLL numerical flux between two adjacent cells  $I_i$  and  $I_j$  is given by

$$\mathcal{F}_{ij}^{\text{HLL}} = \begin{cases} F_i & \text{if } S_i > 0, \\ F_{ij}^{\text{HLL}} & \text{if } S_i \leq 0 \leq S_j, \\ F_j & \text{if } S_j < 0. \end{cases} \quad (10)$$

where  $F_{ij}^{\text{HLL}} = \frac{S_j F_{\vec{n}_{ij}}(q_j) - S_i F_{\vec{n}_{ij}}(q_i) + S_i S_j (q_i - q_j)}{S_j - S_i}$ , with  $S_i = u_i \cdot \vec{n}_{ij} - a_i$  and  $S_j = u_j \cdot \vec{n}_{ij} + a_j$  being the signal speeds. Therein,  $u_i$  and  $u_j$ , and  $a_i$  and  $a_j$  are the velocities and the sound speeds of the conservative variables vectors  $q_i$  and  $q_j$ .

# Implementation

- 1 **Generalization of the Cartesian code:** solving two bugs related to my previous project (overdilation in the  $y$ -direction and wrong computation of the total energy with respect to the pressure), then generalizing the code.
- 2 **Implementation of the code for unstructured triangular meshes:** relying on GMSH-generated meshes, creating a **Mesh mutable structure** (class) and adapting the numerical scheme.
- 3 **Implementation for structured triangular meshes:** one personal implementation and another one based on the previous codes for GMSH-generated meshes.

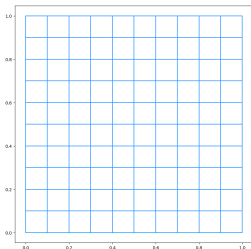


# Simulation

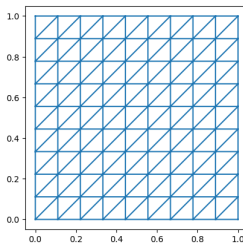
- 1 **Transport problem test case:** simulating the **transport problem** test case on the three grid structures (Cartesian, structured triangular meshes and unstructured triangular meshes) to visualize the diffusive effects with respect to the grid structures and **study the errors and convergence rates**.
- 2 **Gresho vortex test case:** simulating the **Gresho vortex** test case on the three grid structures (Cartesian, structured triangular meshes and unstructured triangular meshes) to study the **diffusive effects with respect to the grid structures**.
- 3 **Performance study:** studying the performances of the code: relying on the transport problem test case to study the errors and the convergence rates, and also to benchmark the **CPU time**.

# Utilized grid structures for the simulation

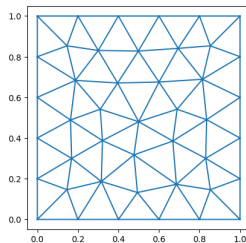
A total of three grid structures (two different sizes for each) were used for the simulation: **Cartesian** grids (2,500 or 10,000 cells), **structured triangular** meshes (2,592 or 10,082 cells) and **unstructured triangular** meshes (2,528 or 10,786 cells).



(a) Cartesian grid



(b) Structured triangular mesh



(c) Unstructured triangular mesh

# Introducing the transport problem test case

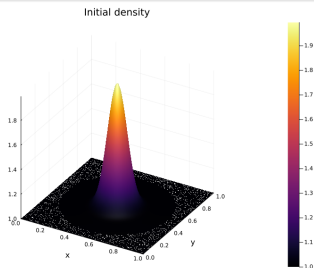
We consider the problem of the transport of density with the following initial data. The physical quantities include

- The pressure  $p = 1$ ,
- The velocity vector  $\mathbf{u} = (1, 1)^T$ ,
- The density

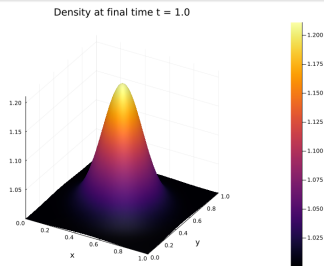
$$\rho = 1 + \exp(-100((x - 0.5)^2 + (y - 0.5)^2)).$$

The computational domain is in that problem  $[0, 1] \times [0, 1]$ . The problem includes periodic boundary conditions in both the  $x$ -and- $y$ -directions. For all the results below, the final time is  $t = 1.0$ .

# Plots of the density of the final solution with respect to the grid structures

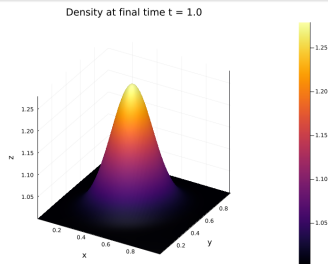


**Figure 5:** Initial density on a 2,500-element Cartesian mesh using the Rusanov flux

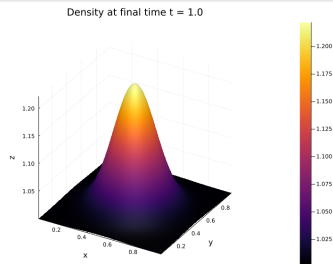


**Figure 6:** Final density on a 2,500-element Cartesian mesh using the Rusanov flux

# Plots of the density of the final solution with respect to the grid structures



**Figure 7:** Final density on a 2,592-element structured triangular mesh using the Rusanov flux



**Figure 8:** Final density on a 2,528-element unstructured triangular mesh (GMSH) using the Rusanov flux

## Plots of the density of the final solution on a Cartesian grid with respect to the mesh size using the HLL flux

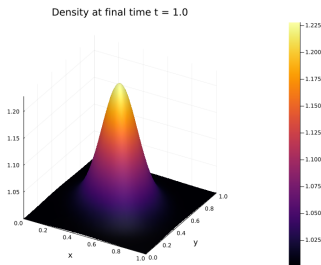


Figure 9: Final density on a 2,500-element mesh

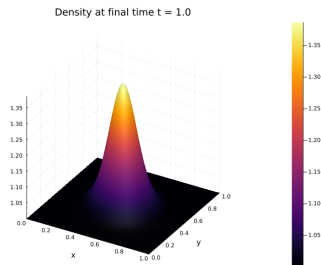


Figure 10: Final density on a 10,000-element mesh

Plots of the density of the final solution on a 10,786-element unstructured triangular mesh with respect to the numerical fluxes

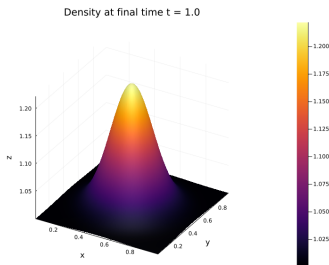


Figure 11: Final density using the Rusanov flux

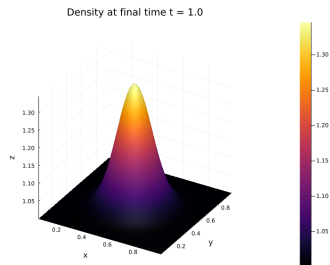


Figure 12: Final density using the HLL flux

## Errors and convergence rates for the Rusanov flux

Mesh Type	Mesh size $h$	Error and Convergence Rate		
		64x64	128x128	256x256
Cartesian	$L^1$ Error	1.438	1.039	0.06881
	$L^2$ Error	0.9543	0.7656	0.5472
	$L^\infty$ Error	3.269	2.776	2.098
	$L^1$ Convergence Rate		0.4557	<b>0.6077</b>
	$L^2$ Convergence Rate		0.3177	<b>0.4847</b>
	$L^\infty$ Convergence Rate		0.2357	<b>0.4041</b>
Structured (triangular)	$L^1$ Error	0.1248	0.08780	0.05598
	$L^2$ Error	0.7656	0.5472	0.03482
	$L^\infty$ Error	0.1739	0.1338	0.09190
	$L^1$ Convergence Rate		0.5076	<b>0.6491</b>
	$L^2$ Convergence Rate		0.3781	<b>0.5417</b>
	$L^\infty$ Convergence Rate		0.3061	<b>0.4765</b>
Unstructured (triangular)	$L^1$ Error	0.1152	0.07836	0.04807
	$L^2$ Error	0.1641	0.1220	0.08028
	$L^\infty$ Error	0.7352	0.5772	0.03992
	$L^1$ Convergence Rate		0.5448	<b>0.7067</b>
	$L^2$ Convergence Rate		0.4199	<b>0.6049</b>
	$L^\infty$ Convergence Rate		0.3421	<b>0.5333</b>

Table 1: Error and Convergence Rate for different mesh types and mesh sizes using the Rusanov flux



## Errors and convergence rates for the HLL flux

Mesh Type	Mesh size $h$	Error and Convergence Rate		
		64x64	128x128	256x256
Cartesian	$L^1$ Error	0.09989	0.06465	0.03811
	$L^2$ Error	0.07342	0.05137	0.03222
	$L^\infty$ Error	3.269	2.776	2.098
	$L^1$ Convergence Rate		0.6302	<b>0.7600</b>
	$L^2$ Convergence Rate		0.5152	<b>0.6731</b>
	$L^\infty$ Convergence Rate		0.4320	<b>0.6011</b>
Structured (triangular)	$L^1$ Error	0.08776	0.05534	0.03213
	$L^2$ Error	0.1334	0.09047	0.05529
	$L^\infty$ Error	0.6086	0.4347	0.2758
	$L^1$ Convergence Rate		0.6551	<b>0.7847</b>
	$L^2$ Convergence Rate		0.5602	<b>0.7104</b>
	$L^\infty$ Convergence Rate		0.4856	<b>0.6563</b>
Unstructured (triangular)	$L^1$ Error	0.08300	0.05176	0.04807
	$L^2$ Error	0.1276	0.08556	0.08028
	$L^\infty$ Error	0.5984	0.4228	0.2644
	$L^1$ Convergence Rate		0.6675	<b>0.8082</b>
	$L^2$ Convergence Rate		0.5650	<b>0.7353</b>
	$L^\infty$ Convergence Rate		0.4914	<b>0.6789</b>

Table 2: Error and Convergence Rate for different mesh types and mesh sizes using the HLL flux

# Introducing the Gresho vortex test case

We consider the problem of the Gresho vortex with the following initial data. The physical quantities are given by

- the initial density  $\rho_0 = 1.0$ ,
- the center of the domain  $[0, 1] \times [0, 1]$   $(x_0, y_0) = (0.5, 0.5)$ , the radius of the vortex  $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$  and the angle  $\phi = \arctan(y - y_0, x - x_0)$ ,
- the velocity in the  $\phi$ -direction
$$u_\phi = \begin{cases} 5.0 \cdot r & \text{if } r < 0.2, \\ 2.0 - 5.0 \cdot r & \text{if } 0.2 \leq r < 0.4, \\ 0.0 & \text{otherwise.} \end{cases}$$
- the velocity in the  $x$ -direction  $u_x = -\sin(\phi) \cdot u_\phi$ ,
- the velocity in the  $y$ -direction  $u_y = \cos(\phi) \cdot u_\phi$ ,
- and the density  $\rho = 1.0$ .

# About the Mach number and the kinetic energy

The **Mach number** is a **dimensionless quantity** (similarly to the Reynolds number that allows to distinguish between laminar and turbulent flows) that is used to measure the speed of an object moving in a fluid environment, relative to the speed of sound in that fluid. It can be written as follows:

$$M = \frac{|u|}{a}, \quad (11)$$

where  $u$  (in m/s) is the velocity of the object and  $a$  is the speed of sound (in m/s).

Then, the kinetic energy is the energy that an object possesses due to its **motion**. It is defined as follows:

$$K = \frac{1}{2} \rho (u_x^2 + u_y^2). \quad (12)$$

Plots of the density of the final solution with respect to the grid structures  
for a Mach number of 0.05 with the HLL flux

The final time is  $t = 0.1$ .

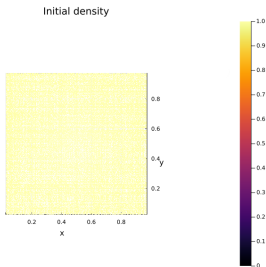


Figure 13: Initial density on a 2,528-element unstructured triangular mesh

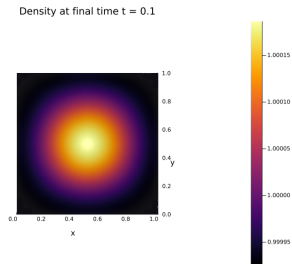


Figure 14: Final density on a 2,500-element Cartesian mesh

Plots of the density of the final solution with respect to the grid structures  
for a Mach number of 0.05 with the HLL flux

The final time is  $t = 0.1$ .

Density at final time  $t = 0.1$

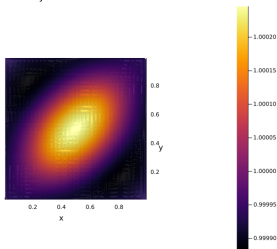


Figure 15: Final density on a 2,592-element structured triangular mesh

Density at final time  $t = 0.1$

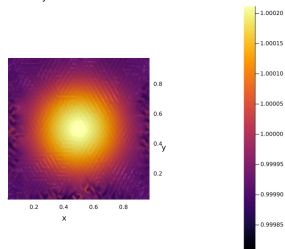
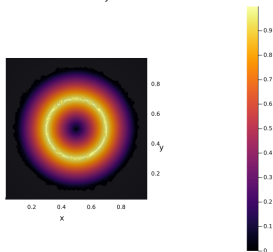


Figure 16: Final density on a 2,528-element unstructured triangular mesh

Plots of the velocity of the final solution with respect to the grid structures  
for a Mach number of 0.05 with the HLL flux

The final time is  $t = 0.1$ .

Initial velocity



Velocity at final time  $t = 0.1$

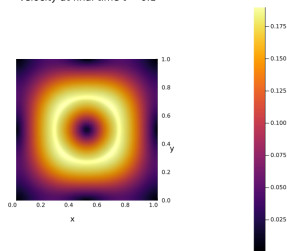


Figure 17: Initial velocity on a 2,528-element unstructured triangular mesh

Figure 18: Final velocity on a 2,500-element Cartesian mesh

Plots of the velocity of the final solution with respect to the grid structures  
for a Mach number of 0.05 with the HLL flux

The final time is  $t = 0.1$ .

Velocity at final time  $t = 0.1$

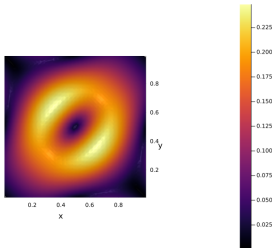


Figure 19: Final velocity on a 2,592-element structured triangular mesh

Velocity at final time  $t = 0.1$

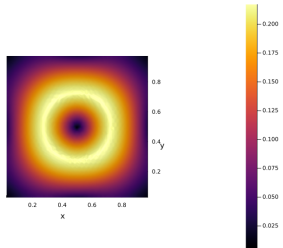


Figure 20: Final velocity on a 2,528-element unstructured triangular mesh

## Plots of the density of the final solution with respect to the Mach number

Here, I use a 10,786-element unstructured triangular mesh (GMSH) and the Rusanov flux. The final time is  $t = 0.1$ .

Density at final time  $t = 0.1$

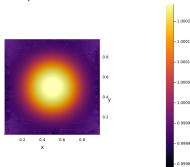


Figure 21: Final density with a Mach number of 0.05

Density at final time  $t = 0.1$

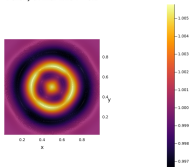


Figure 22: Final density with a Mach number of 0.95

Density at final time  $t = 0.1$

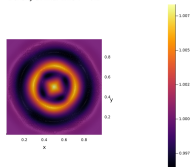


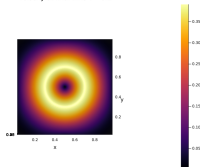
Figure 23: Final density with a Mach number of 1.5



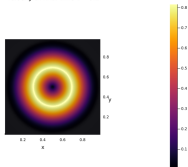
## Plots of the velocity of the final solution with respect to the Mach number

Here, I use the same 10,786-element unstructured triangular mesh (GMSH) and the Rusanov flux. The final time is  $t = 0.1$ .

Velocity at final time  $t = 0.1$



Velocity at final time  $t = 0.1$



Velocity at final time  $t = 0.1$

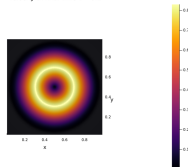


Figure 24: Final velocity with a Mach number of 0.05

Figure 25: Final velocity with a Mach number of 0.95

Figure 26: Final velocity with a Mach number of 1.5

## Decrease of the kinetic energy ratio over time

The plots below show the evolution of the kinetic energy ratio defined by  $\kappa = \frac{\max K}{\max K_0}$  over time. I here use the Rusanov flux. The final time is  $t = 1.0$ .

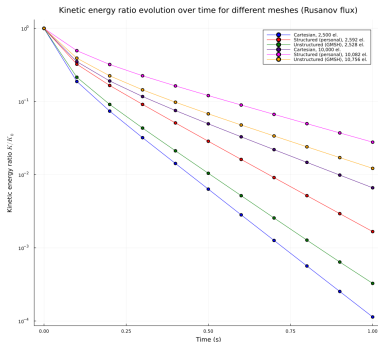


Figure 27: Kinetic energy ratio evolution over time

## Number of iterations for the Gresho vortex test case for meshes approaching a resolution of $50 \times 50$

The final time is  $t = 0.1$  and the Rusanov flux is used.

Mesh Type	Mach Number	Iterations (Rusanov)
Cartesian	0.05	987
	0.95	97
	1.5	85
Structured (pers.)	0.05	2,801
	0.95	286
	1.5	253
Unstructured	0.05	5,884
	0.95	582
	1.5	513

Table 3: Number of iterations

# Benchmark of the CPU time for the Transport problem test case

The final time is  $t = 1.0$  and the HLL flux is used.

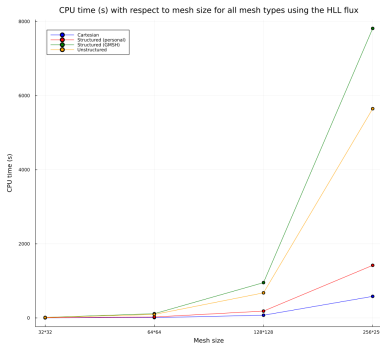


Figure 28: CPU time with respect to the mesh size






# Conclusion

- **Summary: implementation and simulation of 2D Euler equations on different grids** (Cartesian, structured and unstructured triangular), convergence rates approaching 1 for the three grid structures, **less diffusive numerical scheme for triangular grids than for Cartesian grids** (especially for low Mach number Gresho vortex) and lower CPU time for Cartesian grids than for triangular grids.
- **Outlook:** this internship conformed in my desire to pursue in research in applied mathematics.
- **Acknowledgements:** I would like to thank my supervisor, Dr. Andrea THOMANN, for her guidance and her help during this internship.

# Retrospective

- **Success:** implementing and simulating the 2D Euler equations on different grids and obtaining less diffusive numerical schemes for triangular grids than for Cartesian grids.
- **Difficulties:** implementation for structured triangular grids and neighbor-searching algorithm for triangular grids (periodic boundary conditions).
- **Acquired knowledge and skills:** finite volume method extended to triangular grids and general skills in debugging, version control, Julia and GMSH.
- **Axes of improvements:** trying to work more linearly on more sequenced tasks and make more efforts to communicate my personal logic to others.

# References

-  Euler, L., *Principes généraux du mouvement des fluides*, 1757.
-  "Inria & son écosystème". Inria [visited on August 8th 2023]. Available.
-  Toro, E. F., *Riemann solvers and numerical methods for fluid dynamics : A practical introduction*, 2009.
-  Leveque, R. J., *Numerical Methods for Conservation Laws*, 2000.
-  Michel-Dansac, V., "Finite volume methods", *Finite volume schemes for the Euler equations on unstructured meshes*, PhD Thesis, University of Nantes, 2016.