
**António João Ferreira Reis**

# Validation of NASA Rotor 67 with OpenFOAM's Transonic Density-Based solver

Dissertação para obtenção do Grau de Mestre em Engenharia Mecânica

Orientador: Prof. Doutor José Conde

Júri:

Presidente: Doutor José Fernando de Almeida Dias

Arguente: Doutor Luís Manuel de Carvalho Gato

Vogal: Doutor José Manuel Paixão Conde


**Março 2013**

# Validation of NASA Rotor 67 with OpenFOAM's Transonic Density-Based solver

## Abstract

A transonic density based solver implemented in OpenFOAM is shown to be robust and presents consistent results, when simulating three dimensional viscid flows in a low pressure compressor. Towards the validation of the code above, the NASA Rotor 67 geometry has been tested in different mesh levels, with and without *tip-gap*, with grid independence being suggested.

It is used an odd approach in turbomachinery, to compare experimental with numerical data: BIAS, Root Mean Square Error and Concordance Index compare results in pitch-wise and stream-wise directions, provided a linear interpolation to correct all data.

Several Riemann problems are simulated to test the shock resolution and entropy conditions due to the transonic nature of the Rotor 67 flow.

Finally, a full compressor stage is assembled with the Mixing Plane method and a solution to the multi-row problem is achieved, increasing the reliability of OpenFOAM libraries.

## Resumo

O código transónico de massa específica variável implementado no *software* OpenFOAM mostra-se robusto e apresenta resultados consistentes, quando são simulados escoamentos víscidos tri-dimensionais num compressor de baixa relação de pressão. Com vista a validação do código, testa-se a geometria do NASA Rotor 67 em diferentes níveis de malha, com e sem espaçamento entre a pá e a caixa, ficando sugerida uma independência da mesma.

Segue-se uma abordagem pouco comum nas turbomáquinas, para comparar resultados experimentais com numéricos: os indicadores estatísticos BIAS, Raiz Quadrática Média do Erro e o Índice de Concordância compararam valores nas direções das linhas de corrente e do ângulo de ataque, uma vez que todos os valores tenham sido normalizados com uma interpolação linear.

Foram também resolvidos variados problemas de Riemann para testar a resolução de ondas de choque e satisfação de condições de entropia devido à natureza transónica do escoamento no Rotor 67.

Por último, constrói-se um andar completo de um compressor com o método *Mixing Plane* e obtém-se a solução do escoamento, com a finalidade de acrescentar confiança no uso do software OpenFOAM.

## Acknowledgements

I am thankful to Thy-Engineering for the internship they provided: for the international experience and for the design/simulation knowledge in aerodynamics that end up to motivate this thesis subject.

This thesis must also be dedicated to my parents and siblings which contributed with patience, tolerance, example, motivation and persistence.

I am thankful to my friends and girlfriend for the support during working hours and for all the fun time, necessary to keep the same motivation till the end.

Finally, I am thankful to my supervisor for making decisions easier and pointing interesting paths.

## Agradecimentos

Agradeço à empresa Thy-Engineering pelo estágio que me proporcionou: pela experiência internacional e pelo conhecimento de design/simulação em aerodinâmica que acabou por levar à escolha do tema para esta dissertação.

Esta tese também tem de ser dedicada aos meus pais e irmãos que contríbuiram com paciência, tolerância, exemplo, motivação e persistência.

Agradeço aos meus amigos e à minha namorada pelo apoio nas horas de trabalho e pelas horas de diversão, necessárias para manter a mesma motivação até ao final.

Por fim, agradeço ao meu orientador por tornar mais fáceis variadas decisões e sugerir caminhos pertinentes.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

## Abbreviations

| | |
|---|---|
| $3D$ | Three dimensional |
| $\Pi$ | Total pressure ratio |
| ALE | Arbitrary Lagrangian-Eulerian |
| AP | Axial Position |
| ARS | Approximate Riemann Solver |
| B2B | Blade to Blade |
| BV | Blade Vortices |
| CFD | Computational Fluid Dynamics |
| CHT | Conjugate Heat Transfer |
| CI | Concordance Index |
| CP | Circumferential Position |
| CV | Coriolis Vortices |
| DNS | Direct Numerical Simulation |
| DS | Domain Scaling |
| FNMB | Full Non Matching Boundary |
| FSI | Fluid-Structure Interaction |
| GGI | General Grid Interface |
| GUI | Graphic User Interface |
| HP | High Pressure |

| | |
|---|---|
| IP | Intermediate Pressure |
| IRS | Implicit Residual Smoothing |
| LA | Laser Anemometer |
| LE | Leading Edge |
| LES | Large Eddy Simulation |
| LMSC | $L^p$ Metric Space Condition |
| LP | Low Pressure |
| LTS | Local Time Stepping |
| MP | Mixing Plane |
| MUSCL | Monotone Upstream-Centered Schemes for Conservative Laws |
| NACA | National Advisory Committee for Aeronautics |
| NASA | National Aeronautics and Space Administration |
| NHM | Nonlinear Harmonic Method |
| PS | Pressure Side |
| PV | Passage Vortices |
| r$\times\theta$-M | Radius$\times$Theta-Meridional |
| R67 | Rotor 67 |
| RMSE | Root Mean Square Error |
| RP | Radial Position |
| SS | Suction Side |
| SST | Shear Stress Transport |
| TE | Trailing Edge |
| TVD | Total Variation Diminishing |
| WNBEG | Starting Window Number for Laser Anemometer Survey |

## Greek Symbols

| | | |
|---|---|---|
| $\delta_{ij}$ | Kronecker delta | - |
| $\eta$ | Efficiency | - |
| $\gamma$ | Specific heat ratio | - |
| $\mu$ | Viscosity | $\mathrm{kg\,m^{-1}\,s^{-1}}$ |

| | | |
|---|---|---|
| $\nu$ | Kinematic viscosity | $\mathrm{m^2\,s^{-1}}$ |
| $\Omega$ | Vorticity | $\mathrm{s^{-1}}$ |
| $\omega$ | Specific dissipation rate | $\mathrm{s^{-1}}$ |
| $\phi$ | Adimentional velocity | - |
| $\Pi$ | Pressure ratio | - |
| $\psi$ | Adimentional work | - |
| $\rho$ | Density | $\mathrm{kg\,m^{-3}}$ |
| $\sigma$ | Total shear stress tensor; Effective turbulent quantity | $\mathrm{kg\,m^{-1\,-2}s}$ |
| $\tau$ | Shear stress | $\mathrm{kg\,m^{-1\,-2}s}$ |
| $\tau_t$ | Pseudo time-step | s |
| $\upsilon$ | Volume | $\mathrm{m^3}$ |
| $\varepsilon$ | Turbulent dissipation | $\mathrm{m^2\,s^{-3}}$ |
| $\varphi$ | Arbitrary quantity | - |

## Subcripts

| | |
|---|---|
| $0$ | Total quantities |
| $1$ | Rotor inlet (note the ambiguity with Runge-Kutta coefficients) |
| $2$ | Rotor outlet (note the ambiguity with Runge-Kutta coefficients) |
| $a$ | Adiabatic |
| $c$ | Convective terms |
| $d$ | Dissipative terms |
| $i$ | Cell index (note the ambiguity with the matrix index) |
| $l$, $L$ | Left |
| $r$, $R$ | Right |
| $s$ | Static quantities |
| $t$ | Temporal term |
| $*$ | Star Region |
| $blade$, $rot$ | Blade |
| $diaph$ | Diaphragm |
| $fan$ | Expansion fan |

| | | |
|---|---|---|
| $_{Lam}$ | Laminar | |
| $_{max}$ | Maximum | |
| $_{min}$ | Minimum | |
| $_{r,\theta,z}$ | Cylindrical directions | |
| $_{sw}$ | stream-wise | - |
| $_{Turb}$ | Turbulent | |
| $_{x,y,z}$ | Cartesian directions | |

## Supercripts

| | | |
|---|---|---|
| $^{T}$ | Transpose | |
| $^{n+1}$ | Next time step | |
| $^{n}$ | Current time step | |

## Mathematical Symbols

| | | |
|---|---|---|
| $\Delta$ | Difference operator | - |
| $\frac{\partial}{\partial}$ | Partial derivative operator | - |
| $\nabla$ | Gradient operator | - |
| $\nabla \cdot$ | Divergence operator | - |
| $\nabla \cdot \nabla$ | Laplacian operator | - |
| $\Re_b$ | Blade-to-Blade radius | m |
| $\Re_n$ | Meridional radius | m |
| sup | Supremum operator | - |
| $\mathbf{F}$ | Flux tensor | - |
| $\mathbf{F}^{hllc}$ | Godunov flux | - |
| $\mathbf{R}$ | Residual vector | - |
| $\mathbf{U}$ | Conservative state vector | - |
| $\mathbf{W}$ | Primitive variables | - |
| $\tilde{A}$ | Mass-averaged of an arbitrary quantity | - |
| $\vec{\mathbf{U}}$ | Absolute velocity vector | $\mathrm{m\,s^{-1}}$ |
| $\vec{\mathbf{U}}_{rel}$ | Relative velocity vector | $\mathrm{m\,s^{-1}}$ |
| $\vec{q}$ | Heat flux vector | $\mathrm{kJ\,m^{-2}\,s^{-1}}$ |

| | | |
|---|---|---|
| $A''$ | Arbitrary quantity fluctuation | - |
| $c, \quad a$ | Sound velocity | $\mathrm{m\,s^{-1}}$ |
| $C_{cfl}$ | Courant-Friedrich-Lewy number | - |
| $C_p$ | Specific heat (constant pressure) | $\mathrm{kJ\,kg^{-1}\,K^{-1}}$ |
| $E$ | Energy | kJ |
| $F$ | Decomposed flux tensor in an arbitrary referential | - |
| $h$ | Enthalpy | $\mathrm{kJ\,kg^{-1}}$ |
| $K, \quad k$ | Turbulent kinetic energy | $\mathrm{m^2\,s^{-2}}$ |
| $Ma$ | Mach number | - |
| $O$ | Order | - |
| $P$, $p$ | Pressure | $\mathrm{kg\,m^{-2}}$ |
| $Pr$ | Prandt number | - |
| $Re$ | Reynolds number | - |
| $S$ | Wave speed | $\mathrm{m\,s^{-1}}$ |
| $S_{ij}$ | Mean strain tensor | $\mathrm{s^{-1}}$ |
| $T$ | Temperature | K |
| $t$ | Time | s |
| $u, v, w$ | Velocity components in a Cartesian referential | $\mathrm{m\,s^{-1}}$ |
| $V, \quad U$ | Magnitude of (absolute) velocity | $\mathrm{m\,s^{-1}}$ |
| $V_\theta$ | Tangential velocity | $\mathrm{m\,s^{-1}}$ |
| $W$ | Magnitude of relative velocity | $\mathrm{m\,s^{-1}}$ |
| $x, y, z$ | Cartesian coordinates | m |
| $Y^+$ | Dimensionless distance to wall | - |
| R | Specific gas constant | $\mathrm{kJ\,kg^{-1}K^{-1}}$ |
| r,$\theta$,z | Cylindrical coordinates | - |
| W | Work | $\mathrm{kJ\,s^{-1}}$ |

# CHAPTER 1

# INTRODUCTION

Turbomachinery plays an important role in the industry, in the world's energy supply and mobility. From an engineering perspective, its applications can be divided into their governing physics: compressible and incompressible flows. This work only concerns the former.

Turbomachines ruled by compressible flows are designed to create a great pressure differential. The most sounding example is the gas turbine cycle used in airplanes, automobiles and even boats: air flows through compressors, combustors and turbines. The latter extracts some if not all energy, depending on the applications, i.e. a fighter doesn't need to pass much energy to a shaft while in a Thermoelectric, that is the main purpose.

Although there are considerably distinct applications, the flow structures are similar between all and it becomes important to understand how they work in detail. *Computational Fluid Dynamics* (CFD) and the advance in computing performance in the last decades allowed complex and large simulations to be made in reasonable time, thus much of the pre-project budget is reduced. Furthermore, it has proved to be a powerful research tool when examining complex physics and multi-phenomena interaction, namely on turbomachinery.

In CFD, an important task is to validate numerical results so that these compu-

tational methods can be used with safety. The motivation of this thesis is to add reliability to the steady density based solver implemented by Borm *et. al.* [8] in the *OpenFOAM* libraries: the *transonicMRFDyMFoam*. Moreover, the code used in this thesis is open-source which clearly is an advantage compared to commercial codes.

The main objective of this thesis is to solve a three dimensional (3D) flow of an aircraft fan component, the NASA Rotor 67, and compare it with experimental data. A fan is a *low pressure* (LP) compressor which is part of a turbofan's compressor set (Figure 1.1). It is responsible for generating about 80% of the engine's thrust in common commercial airplanes. Statistical tools and plots will show how well data matches.

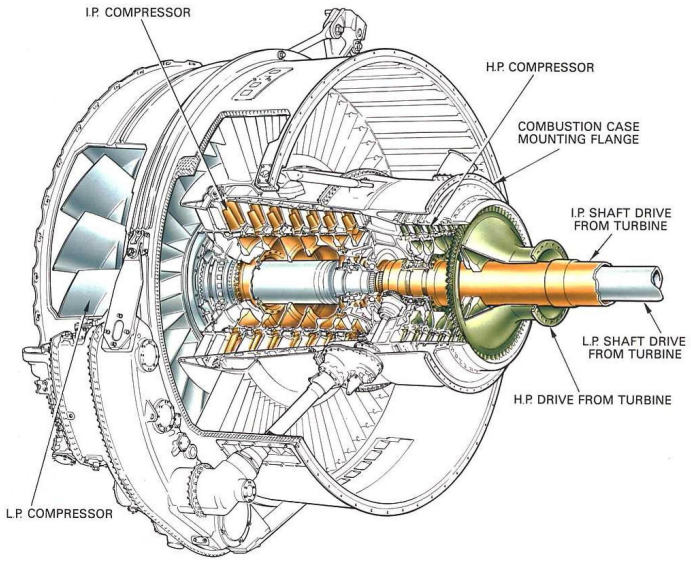



Figure 1.1: Turbofan's compressor set: LP, intermediate pressure (IP), high pressure (HP) compressors (Courtesy Rolls Royce [1])

This thesis will also present some simple tests, called Riemann problems, that show how accurate the hyperbolic system in the Navier-Stokes equations is solved. By last, a numerical feature specially designed to turbomachinery is tested: the mixing plane. It allows steady multi-row problems to be solved which gives strength to OpenFOAM as a reliable alternative to commercial software.

This Chapter will continue by introducing some important concepts and phenom-

ena that are useful when dealing with turbomachinery. Chapter 2 will describe the numerical methods that build the OpenFOAM's transonic density-based solver while Chapter 3 presents the Riemann problems. Chapter 4 is the highlight of the thesis and will present the NASA Rotor 67. Chapter 5 presents the multi-row simulation and Chapter 6 is reserved for final appreciations and conclusions.

## 1.1 Compressible flow

Compressible flows are considered when fluid velocity is above Mach ($Ma$) 0.3, becoming comparable with sound velocity, $Ma = 1$. In air, sound velocity is a function of state, i.e. function of static temperature ($T_s$). Flows in the transonic region, $0.8 < Ma < 1.2$ are a characteristic of turbomachinery thus they are the focus of this thesis. When in presence of such flows, some phenomena may happen: they are shock waves, choked flows and surge flows.

Shock waves [9] start to exhibit when flow enters the transonic region and they are characterized by a thermodynamic state discontinuity, usually velocity decreases with an increase of $P_s$ and $T_s$. They indeed provoke a compression, which is the objective of a compressor, but are mainly a source of losses and noise. Shock waves in wall bounded flows have and additional complexity because there exists an increased interaction with the boundary layer (Figure 1.2).

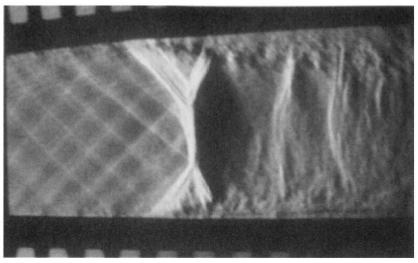


Figure 1.2: Normal shock in a wall bounded flow: interaction between shock and boundary layer can be seen (after Korpela [2])

Choked flows are called when there is a section in a duct where flows reach $Ma = 1$.

It is demonstrated for isentropic flows [10] that this critical section imposes a maximum mass flow. This phenomena obviously must be taken in mind when projecting a duct, or more precisely a turbo-machine, because depending on their geometry , mass flow will always be limited.

Surge flows exhibit when flow incidence in compressor blades starts to be misaligned with their metallic angles: separation produces losses and instability. This phenomena is closely related to stall: it usually happens with low mass flow and airflow can simply stop, entering a surge cycle (Figure 1.3).

Another compressible flow effect is that air compressibility can no longer be neglected meaning that density ($\rho$) of air will change. For further details see White [10] or Anderson [9].

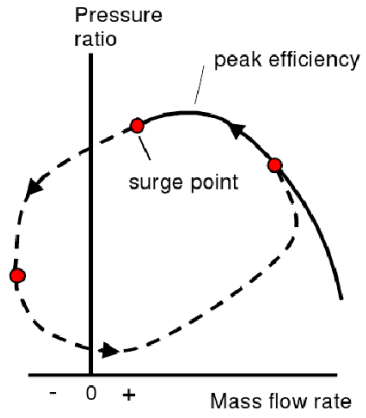



Figure 1.3: Surge cycle: strong separation makes airflow stop (stall) and high pressure air downstream rushes back through compressor. When back-flow stops, airflow is reestablished and the cycle repeats itself (Courtesy of Thy-Engineering; Private communication)

## 1.2 Turbomachinery

It is important to introduce static and stagnation quantities. Static quantities define a thermodynamic state. Taking the line of thought of Anderson Jr. [9]: static quantities

is what you feel if you were moving with the flow. Total quantities is what you would feel if you could grab the flow and *adiabatically* take the velocity to zero. In other words, total quantities take into account kinetic energy and it is usual to define the following [2]: total enthalpy (Eq. 1.1), total pressure (Eq. 1.2) and total temperature (Eq. 1.3).

$$h_0 = h + \frac{V^2}{2} \tag{1.1}$$

with $h = C_p T_s$ being the static enthalpy, $C_p$ the specific heat at constant pressure and $V$ the magnitude of absolute velocity.

$$P_0 = P_s \times (1 + \frac{\gamma - 1}{2} Ma^2)^{\frac{\gamma}{\gamma - 1}} \tag{1.2}$$

$$T_0 = T_s \times (1 + \frac{\gamma - 1}{2} Ma^2) \tag{1.3}$$

with $Ma = \frac{V}{c}$, $c = \sqrt{\gamma R T_s}$ is the speed of sound,$\gamma$ the specific heat ratio and $R$ the specific gas constant.

It follows that overall aerodynamic parameters [6] can be defined and will be used in the next chapters. They are adiabatic efficiency (Eq. 1.4) and total pressure ratio (Eq. 1.5),

$$\eta_a = \frac{\left(\frac{P_{02}}{P_{01}}\right)^{\frac{(\gamma-1)}{\gamma}} - 1}{\frac{T_{02}}{T_{01}} - 1} \tag{1.4}$$

$$\Pi = \frac{P_{02}}{P_{01}}, \tag{1.5}$$

where subscripts $_1$ and $_2$ denote inlet and outlet of the rotor, respectively.

### 1.2.1 Compressor flow

Compressor flow is characterized by a deceleration of relative velocities. It can be seen in Figure 1.4 the Euler velocity triangles of a rotor and a stator. The work done by the blades [2] is related to the change in tangential velocity ($V_\theta$), according to Eq. 1.6.

$$\mathrm{W} = U_{blade} \left(V_{\theta 2} - V_{\theta 1}\right), \tag{1.6}$$

with $U_{blade}$ being the blade velocity.

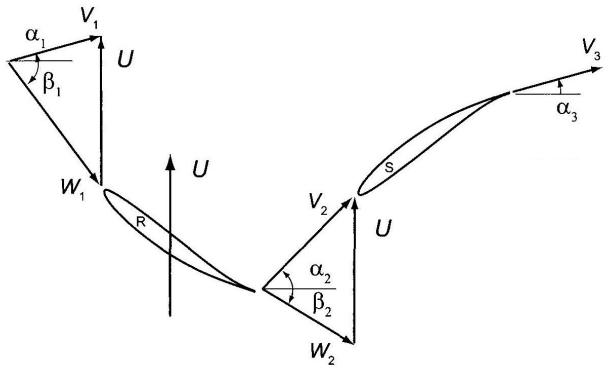



Figure 1.4: Euler triangle velocities in a compressor stage; exceptional notation for blade velocity $U$ (after Korpela [2])

Compressors are designed for the flow to achieve a pressure ratio at a specific mass flow. It is called the design point and it is thought to be the point where the compressor will work more often. Usually, there will be more points where the compressor needs to work and it will be limited by the choked mass flow and surge or stall conditions. Therefore, every compressor has a map closely related to Figure 1.5: $\psi$ and $\phi$ denote dimensionless work and velocity, respectively. It can be advanced that a map for NASA Rotor 67 will be made in Chapter 4.

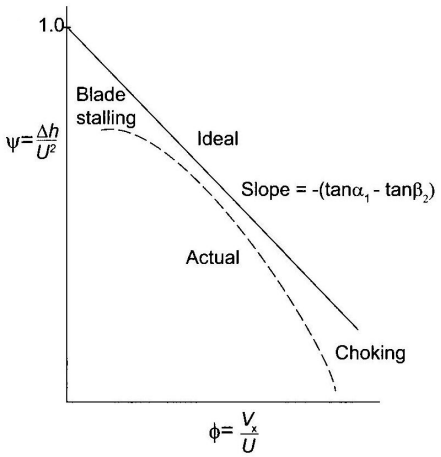



Figure 1.5: Example of a compressor map (after Korpela [2])

It is important to remark some phenomena in compressor flows that will gain bigger relevance in next sections. There is a clearance between the blade and the casing called *Tip-Gap*: flow that goes through that gap is called *tip leakage*. Figure 1.6 shows a flow structure provoked by the tip-gap: a tip vortex. It is created because the suction and pressure side don't have a surface to separate them in the gap and so they struggle to get mixed. In Chapter 4 this situation will be simulated and illustrated.

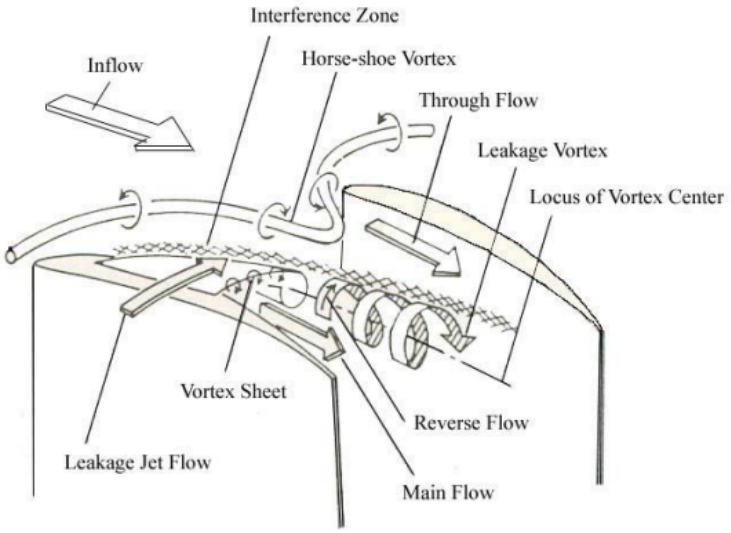



Figure 1.6: Tip-gap flow structure (after Tang [3])

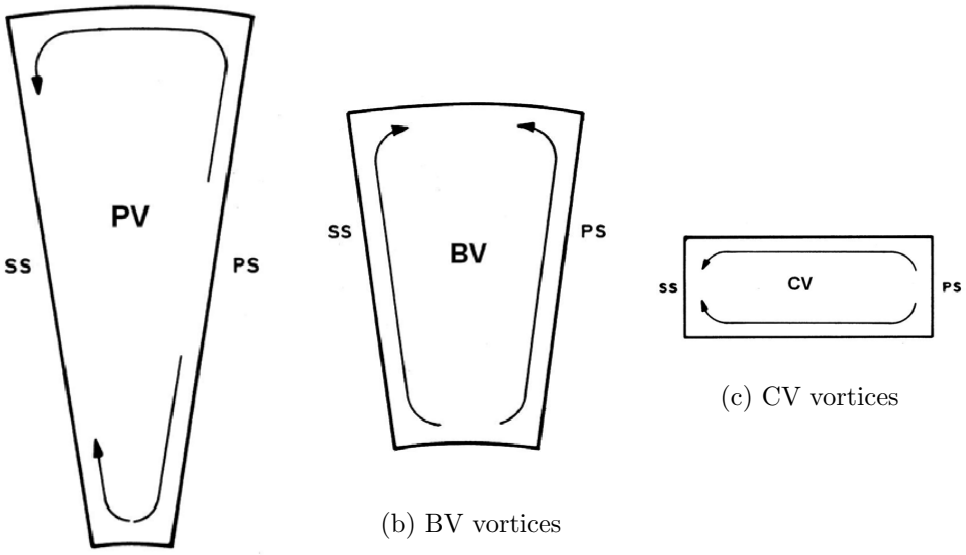
Under certain conditions there is also separation induced by passage shock/boundary layer interaction which was seen in Figure 1.2. By last, secondary flows may appear too and will be described next.

### 1.2.2 Secondary flows

Secondary flows are seen as the difference between the $3D$ inviscid solution and the real viscous flow happening in turbomachinery flows. Van den Braembussche [4] pointed out recently that this flows redistribute low energy fluid, due to blockage, through the stream-wise vorticity influencing the inviscid core velocity and pressure.

$$\frac{\partial}{\partial s}\left[\frac{\Omega_{sw}}{W}\right] = \frac{2}{W}\left[\frac{1}{\Re_n}\frac{\partial W}{\partial b} + \frac{1}{\Re_b}\frac{\partial W}{\partial n} + \frac{2\Omega}{W}\frac{\partial W}{\partial z}\right] \tag{1.7}$$

The first two right hand side terms of Eq. 1.7 express the generation of vorticity due to flow turning in meridional and *blade-to-blade* (B2B) planes, respectively. The last term represents generation of vorticity due to Coriolis forces. Thus, second and third term generate passage vortices (PV), first term generates vortices along blade vortices (BV) and again third term also generates Coriolis vortices (CV). $\Re_n$ and $\Re_b$ stand for curvature radius in meridional and B2B plane, respectively. $\Omega_s$ is the stream-wise vorticity and $s$ denotes streamline. Figure 1.7 presents a scheme of these vortices. Further details in [4] [11].




(a) PV vortices

(b) BV vortices

(c) CV vortices

Figure 1.7: Vortices's schemes (after Landmann [4])

# CHAPTER 2

# NUMERICAL METHODS

This chapter will introduce the numerical methods used by Borm *et. al.* [8] to implement the transonic density-based libraries [12] in OpenFOAM. These libraries have mainly two solvers, the steady *transonicMRFDyMFoam* and the unsteady *transonicUnsteadyMRFDyMFoam*. It is also available an *arbitrary Lagrangian–Eulerian* (ALE) [13] method for moving meshes like *Fluid-Structure Interaction* (FSI) or rotating components and a *Conjugate Heat Transfer* (CHT) formulation.

In the main solvers, primitive variables $[\rho, U, p]^T$ are reconstructed by a Monotone Upstream-Centered Schemes for Conservative Laws (MUSCL) [14]. Then, they are used as an input to an upwind *approximate Riemann solver* (ARS) which will return the inviscid numerical fluxes. These were first used in a Godunov scheme [5] to solve the Euler equations.

The viscid terms are build with central difference discretization and are added as diffusive fluxes to inviscid ones, thus a new Godunov-like scheme is assembled. This new scheme is nothing more than a discretized form of the conservative Navier-Stokes equations (N-S). It becomes a set of algebraic equations which are solved by a direct inversion method [15]. Finally, time discretization can reach $3^{rd}$ order if unsteady calculation is done and *Local Time-Stepping* is implemented as an accelerating convergence technique. The following sections will provide further details.

## 2.1 Navier-Stokes equations

The complete form of the N-S presented in the differential form [16] is read

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{U}) = \nabla \cdot \mathbf{F}_d(\mathbf{U}, \nabla \mathbf{U}) \tag{2.1}$$

where $\mathbf{U}$ is a conservative state vector

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{2.2}$$

$\mathbf{F}_c = (F_c^x, F_c^y, F_c^z)$ is the convective flux tensor

$$F_c^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, F_c^y = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, F_c^z = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix} \tag{2.3}$$

$\mathbf{F}_d = (F_d^x, F_d^y, F_d^z)$ is the dissipative flux tensor

$$\begin{aligned} F_d^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x \end{pmatrix}, F_d^y = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y \end{pmatrix}, \\ F_d^z = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z \end{pmatrix} \end{aligned} \tag{2.4}$$

and $\tau_{ij}$ the dissipative stress tensor

$$\tau_{ij} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix}, \tag{2.5}$$

considered symmetric and a linear function of velocity gradients due to Newtonian fluid assumptions [15].

If viscosity and thermal conduction is neglected one arrives to the Euler equations, which are solved first by Godunov for compressible flows [5].

When dealing with problems that admit discontinuities (i.e. shocks) it becomes necessary to solve the N-S equations in a conservative form. Non-conservative numerical schemes will fail in providing the right shock speed if we are in presence of a Riemann problem [5]. As Zanoti and Manca [17] remember, Hou & Le Floch [18] proved that non-conservative schemes do not converge if a shock wave is part of the solution while Lax & Wendroff [19] showed that conservative schemes, if convergent, they converge to the weak solution of the problem.

## 2.2 RANS equations in a rotating referential

In order to solve turbulence in a practical way, the transonic density-based libraries are implemented under the N-S equations with mean flow quantities: the so called called Reynolds-Average Navier-Stokes (RANS) equations. In fact, for compressible flows, there is a variation of RANS equations called Favre-average Navier-Stokes. If the first uses a time average of the quantities, its variant uses a mass-average because density indeed changes in compressible flows. Hereinafter this work will use the Favre equations but the reference to the Reynolds-Average Navier-Stokes will remain the same.

Flow quantities of RANS are defined by

$$A = \tilde{A} + A'' \tag{2.6}$$

where the double prime is the fluctuation and the tilde is the mass-average quantity and has the form

$$\tilde{A} = \frac{\overline{\rho A}}{\bar{\rho}} \tag{2.7}$$

with

$$\overline{\rho A} \equiv \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} \rho A dt. \tag{2.8}$$

Eq. 2.1 expanded with Eq. 2.6 gives place to the RANS equations, see William [20] for the full expansion. The following pseudo heat fluxes and stresses appear (also called

Reynolds Stress)

$$(\tau_{ij})_{Turb} = -\overline{\rho u_i'' u_j''} \tag{2.9}$$

$$(q_j)_{Turb} = -C_p \overline{T'' \rho u_j''}. \tag{2.10}$$

The so called closure of the RANS equations is made by expressing the Reynolds stress (Eq. 2.9) with the mean quantities. For most well known turbulence models (i.e. Spalart Allmaras [21], $k-\omega$ [22], $k-\omega$ SST [23]) this is done with a linear relation between Reynolds stress and the mean strain tensor $S_{ij}$

$$S_{ij} = \frac{1}{2}\Big[\frac{\partial \tilde{U}_i}{\partial x_j} + \frac{\partial \tilde{U}_j}{\partial x_i}\Big] \tag{2.11}$$

that is read

$$(\tau_{ij})_{Turb} = -2\mu_{Turb}\Big(S_{ij} - \frac{1}{3}\frac{\partial \tilde{U}_k}{\partial x_k}\delta_{ij}\Big) + \frac{2}{3}\tilde{\rho}\tilde{K}\delta_{ij}. \tag{2.12}$$

Eq. 2.12 is analogous to the linear constitutive equation for Newtonian flows

$$(\tau_{ij})_{Lam} = -2\mu_{Lam}\Big(S_{ij} - \frac{1}{3}\frac{\partial \tilde{U}_k}{\partial x_k}\delta_{ij}\Big), \tag{2.13}$$

being $(\tau_{ij})_{Lam}$ the laminar stress tensor: this analogy is called Boussinesq hypothesis [24]. Note the last term in Eq. 2.12 is added to correct the trace of Reynolds stress [25].

Eq. 2.9 and 2.12 are very popular and are worth some comments. They have been the base for development of the most used turbulence models in industry but fail in physical meaning. Even for the simplest flow, concepts such as fluctuating velocities don't seem to have solid experimental support neither have an explicit derivation from equations. Schmitt [24] reviewed in the detail the work of Boussinesq, pioneer of mean quantities in NS equations, and pointed its major weakness: a bad extrapolation of molecule kinetic energy theory to turbulent flows is made concerning the mean free path of molecules.

Once pointed out the derivation and limitation of RANS, and being this thesis about compressible flow in turbomachines, it makes sense to present those equations in a rotating referential. This relative frame of reference is the only one where steady

state assumptions are valid. Gönç [26] and Borm [8] write them in vectorial form, where quantities are already in their mean value

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{\mathbf{U}}_{rel}) &= 0 \\ \frac{\partial \rho \vec{\mathbf{U}}}{\partial t} + \nabla \cdot (\rho \vec{\mathbf{U}}_{rel} \otimes \vec{\mathbf{U}}) + \nabla p &= -\rho(\vec{w} \times \vec{\mathbf{U}}) + \nabla \cdot \sigma \\ \frac{\partial \rho E}{\partial t} + \nabla \cdot ((\rho E + p)\vec{\mathbf{U}}_{rel} + p\vec{\mathbf{U}}_{rot}) &= \nabla \cdot (\sigma \cdot \vec{\mathbf{U}}) - \nabla \cdot \vec{q} + \nabla \cdot (\mu + \beta^* \mu_T) \nabla K \end{aligned} \tag{2.14}$$

$$\sigma = (\mu_{Lam} + \mu_{Turb}) \Big( \nabla \vec{\mathbf{U}} + (\nabla \vec{\mathbf{U}})^T - \frac{2}{3} (\nabla \cdot \vec{\mathbf{U}}) \delta_{ij} \Big) - \frac{2}{3} \rho K \delta_{ij}. \tag{2.15}$$

Note that Eq. 2.15 is the total shear stress tensor and it is the sum of Eq. 2.12 and 2.13. It is usually called effective turbulent quantity [16].

## 2.3 Hyperbolic system solution

The transonic density-based solvers implemented in OpenFOAM by Borm *et. al.* [8] use an ARS to calculate the inter-cell's flux based on characteristic's speed. Then, a Godunov scheme solves the hyperbolic system.

### 2.3.1 Godunov scheme

The Godunov scheme can be written in the conservative form [5]

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x} \left( \mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2} \right). \tag{2.16}$$

To understand how the Godunov scheme is assembled, one must consider the following conservative form of the 1D Euler equations

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \tag{2.17}$$

with the conservative state vector and the flux

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (\rho E + p) u \end{pmatrix}. \tag{2.18}$$

One can divide the spacial domain in $N$ computing cells $I_i = \left[x_{i-1/2}, x_{i+1/2}\right]$ of length $\Delta x = x_{i+1/2} - x_{i-1/2}$ with a "height" of $\Delta t = \left[t^n - t^{n+1}\right]$ (Figure 2.1). Integrating Eq. 2.17 on cell $I_i$, first in space and then in time, it results the following

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}\left(x, t^{n+1}\right) dx = \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}\left(x, t^{n}\right) dx + \int_{t}^{t+1} \mathbf{F}\left(\mathbf{U}\left(x_{i-1/2}, t\right)\right) - \int_{t}^{t+1} \mathbf{F}\left(\mathbf{U}\left(x_{i+1/2}, t\right)\right). \tag{2.19}$$

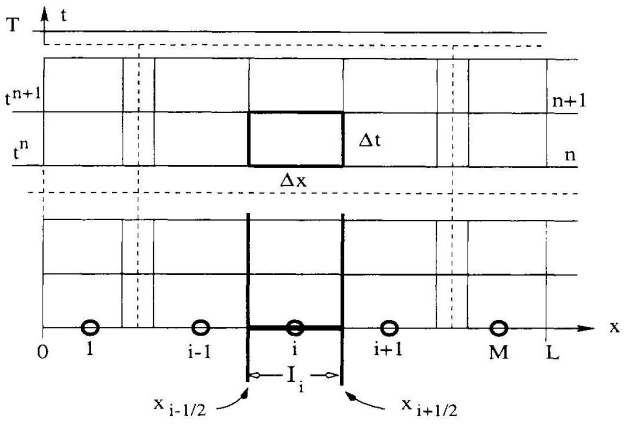



Figure 2.1: Computational mesh (after Toro [5])

At this point two concepts are introduced. Eq. 2.20 makes some information being lost and the scheme to be first order. This is called the piecewise constant data [5] and it is the first concept. It is convenient to note that in order to make this averaging, waves can't interact (Figure 2.2). The second (Eq. 2.21) is the Godunov flux or inter-cell numerical flux. Rewriting Eq. 2.20 and Eq. 2.21 into Eq. 2.19 one arrives to the Godunov scheme.

$$\mathbf{U}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}\left(x, t^n\right) dx \tag{2.20}$$

$$\mathbf{F}_{i\pm 1/2} = \mathbf{F}\left[\mathbf{U}\left(x_{i\pm 1/2}, t\right)\right] \tag{2.21}$$

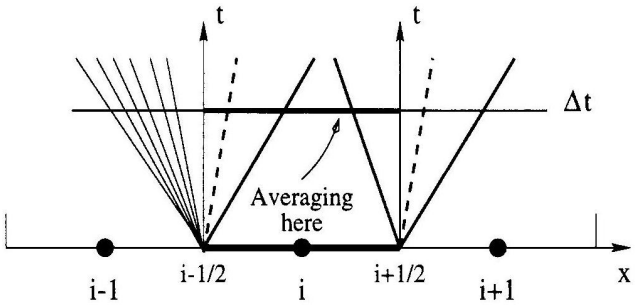



Figure 2.2: Godunov averaging of local solutions to the Riemann problem within cell $I_i$ at fixed time $\Delta t$ (after Toro [5])

Other note about this scheme is that in order to prevent the interaction of waves described above, the time step $\Delta t$ must satisfy the following condition,

$$\Delta t \leq \frac{C_{cfl}\Delta x}{S^n_{max}} \tag{2.22}$$

with $S^n_{max}$ being the fastest wave speed in the domain at time $t^n$. This is the so called *Courant-Friedrich-Lewy (CFL) condition* [27] and $C_{cfl}$ is the CFL number . For the 3D Euler system, results follow similarly.

### 2.3.2 Solution of the Riemann problem in the Euler equations

Considering the following initial valued problem with discontinuous data, the Riemann problem [5]

$$\begin{array}{lcl} \textit{Hyperbolic system} & : & \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \\ \textit{Initial condition} & : & \mathbf{U}(x,0) = \mathbf{U}^{(0)}(x), \\ \textit{Boundary condition} & : & \mathbf{U}(0,t) = \mathbf{U}_l(t),\ \ \mathbf{U}(L,t) = U_r(t), \end{array} \tag{2.23}$$

in a domain $x_l < x < x_r$ using the explicit conservative formula of Eq. 2.16. The Godunov flux at interface $x_{i+1/2}$ is defined by

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}\left(\mathbf{U}_{i+\frac{1}{2}}(0)\right), \tag{2.24}$$

being $\mathbf{U}_{i+\frac{1}{2}}(0)$ the exact similarity solution $\mathbf{U}_{i+\frac{1}{2}}(x/t)$ of the Riemann problem. Figure 2.3 shows all the possible wave patterns resulting from the Euler equations and Table

2.1 shows all the possible state solutions $\mathbf{W}_{i+\frac{1}{2}}(0)$. Note that they are computed as primitive variables and the star region is the space between the fastest and slowest speed waves (Figures 2.3 and 2.4).

| Sub-case | Case (a) positive speed $u_*$ | Case (b) negative speed $u_*$ |
|---|---|---|
| 1 | $\mathbf{W}_L$ | $\mathbf{W}_R$ |
| 2 | $\mathbf{W}_{*L}$ | $\mathbf{W}_{*R}$ |
| 3 | $\mathbf{W}_L$ | $\mathbf{W}_R$ |
| 4 | $\mathbf{W}_{*L}$ | $\mathbf{W}_{*R}$ |
| 5 | $\mathbf{W}_{Lfan}$ | $\mathbf{W}_{Rfan}$ |

Table 2.1: Value of $\mathbf{W}_{i+\frac{1}{2}}(0)$ required to evaluate the inter-cell flux (after Toro [5])

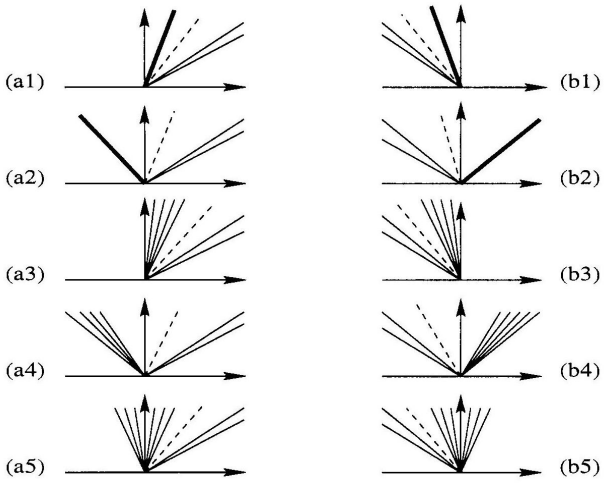



Figure 2.3: Wave pattern possibilities; a) positive particle velocity in star region b) negative particle velocity in star region; Thick, dashed and multi lines represent shock, contact and rarefaction waves (after Toro [5])

For example, considering the situation in which $u_*$ is positive, to calculate $\mathbf{W}_{i+\frac{1}{2}}(0)$ one must identify the character of the left wave. If it is a shock, one calculates the state $\mathbf{W}_{*L}$ between the contact and shock wave through shock relations [5]. Than the speed $S_L$ of the left shock is computed. If $S_L \geq 0$ it is said supersonic and

$$\mathbf{W}_{i+\frac{1}{2}}(0) = \mathbf{W}_L. \tag{2.25}$$

If $S_L \leq 0$ it is said subsonic and

$$\mathbf{W}_{i+\frac{1}{2}}(0) = \mathbf{W}_{*L}. \tag{2.26}$$

It follows that the inter-cell flux is calculated by Eq. 2.27 and the remaining possibilities presented in Figure 2.3 are solved in a analogous way.

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}\left(\mathbf{W}_{i+\frac{1}{2}}(0)\right) \tag{2.27}$$

This method is the exact[1] solution of the Riemann problem applied to the Euler equations but it is not efficient due to large number of iterations. Next section will present an approximate Riemann solver called Harten Lax Van Leer Contact (HLLC) which is more practical and it is used in all simulations.

### 2.3.3 The HLLC approximate Riemann solver

Harten, Lax and Van Leer (HLL) [28] introduced the HLL approximate Riemann solver which consisted in defining a new wave structure. Because they considered just two waves [5], the solver becomes unacceptable when solving the Euler equations: the contact wave is missing and has an important physical meaning.

Toro, Spruce and Spears [29] introduced the HLLC approximate Riemann solver that takes the HLL and recovers the contact wave. The star region is composed by two states and the structure of the solution is presented in Figure 2.4.

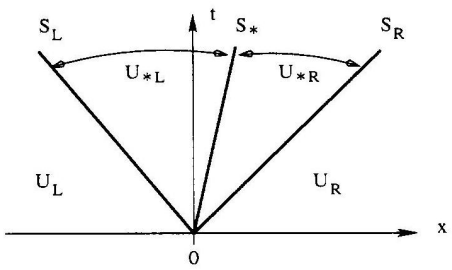



Figure 2.4: Structure of the HLLC approximate solution (after Toro [5])

Then, the HLLC solver is given by

[1] It doesn't have an analytical solution, the user specifies the desired accuracy

$$\tilde{\mathbf{U}}(x,t) = \begin{cases} \mathbf{U}_L & if \quad \frac{x}{t} \leq S_L \\ \mathbf{U}_{*L} & if \quad S_L \leq \frac{x}{t} \leq S^* \\ \mathbf{U}_{*R} & if \quad S^* \leq \frac{x}{t} \leq S_R \\ \mathbf{U}_R & if \quad \frac{x}{t} \geq S_R \end{cases} \tag{2.28}$$

By applying the Rankine-Hugoniot conditions [30] and relating the states in the star region [5], the Godunov flux to be applied into the Godunov scheme can be written as

$$\mathbf{F}_{i+\frac{1}{2}}^{hllc} = \begin{cases} \mathbf{F}_L & if \quad \frac{x}{t} \leq S_L \\ \mathbf{F}_{*L} = \mathbf{F}_L + S_L(\mathbf{U}_{*L} - \mathbf{U}_L) & if \quad S_L \leq x/t \leq S^* \\ \mathbf{F}_{*R} = \mathbf{F}_R + S_L(\mathbf{U}_{*R} - \mathbf{U}_R) & if \quad S^* \leq x/t \leq S_R \\ \mathbf{F}_R & if \quad \frac{x}{t} \geq S_R \end{cases} \tag{2.29}$$

It only remains to calculate the wave speeds which can be done directly from data

$$S_L = u_L - a_L, S_R = u_R + a_R \tag{2.30}$$

and

$$S_L = min\{u_L - a_L, u_R - a_R\}, \quad S_R = max\{u_L + a_L, u_R + a_R\}. \tag{2.31}$$

where $u$ denotes the particle speed and $a$ the sound speed. For different approaches in calculating wave speeds or more details about the Riemann problem, see Toro [5].

## 2.4 Turbulence modeling

It is widely accepted that the 3D N-S are the correct representation of turbulent flow [31]. Every length scales are modeled in a RANS approach whether in Large Eddy Simulation (LES) only the smallest scales [32] are modeled. In Direct Numerical Simulations (DNS) every scale is calculated which leads to the need of massive computational resources. Landmann [16] rescues Kolmogorov and its micro-scale theory, pointing out that one needs $O(Re^{3/4})$ to resolve all turbulent length scales per space dimension. Due to the fact that turbulence is a three dimensional phenomena, one would need $O(Re^{9/4})$ grid cells.

OpenFOAM has a wide range of turbulence models available and these transonic density-based solvers can use every RANS based model, i.e. one that returns a turbulent

viscosity $\mu_T$. Borm [8] points out there might be some conflict in rotor-stator problems when using a turbulence model that use vorticity terms because velocity jumps in the interface. The *Shear stress transport* model was used and doesn't have this problem: it is described below.

### 2.4.1 Shear stress transport (SST)

The SST model, introduced by Menter [23], is a two equation model that takes the approach of the $k-\omega$ [22] at the inner region of the boundary layer, using the $k-\varepsilon$ in the outer region and free-stream. It modifies the definition of eddy viscosity by taking into account the transport of turbulent shear stress. This modification proved to be very important for accuracy in adverse pressure gradients [33] which is a requisite in aerodynamic turbulence models [34]. The $k-\omega$ formulation makes it robust and a good choice for the logarithmic and viscous sublayer regions. For the outer region and free-stream, $k-\omega$ predictions are very sensible to the free-stream specific dissipation rate $\omega$, therefore SST switches to $k-\varepsilon$ model which hasn't this behavior. In SST transport equations one can read the (1) Lagrangian derivative, (2) production term, (3) destruction term and (4) the dissipation term

$$k: \quad \underbrace{\frac{D\rho k}{Dt}}_{1} = \underbrace{\tau_{ij}\frac{\partial u_i}{\partial u_j}}_{2} - \underbrace{\beta^*\rho\omega k}_{3} + \underbrace{\frac{\partial}{\partial x_j}\Big[(\mu_{Lam}+\sigma_k\mu_{Turb})\frac{\partial k}{\partial x_j}\Big]}_{4} \tag{2.32}$$

$$\omega: \quad \underbrace{\frac{D\rho\omega}{Dt}}_{1} = \underbrace{\frac{\gamma}{\nu_T}\tau_{ij}\frac{\partial u_i}{\partial u_j}}_{2} - \underbrace{\beta^*\rho\omega^2}_{3} + \underbrace{\frac{\partial}{\partial x_j}\Big[(\mu_{Lam}+\sigma_\omega\mu_{Turb})\frac{\partial k}{\partial x_j}\Big] + 2\rho(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\omega x_j}}_{4} \tag{2.33}$$

where constants[2] $C_{k\omega}$, $C_{k\varepsilon}$ correspond to $k-\omega$ model, $k-\varepsilon$, respectively

$$C_{k\omega}\begin{cases} \sigma_{k1}=0.85 \quad \sigma_{\omega 1}=0.5 \qquad \beta_1=0.0750 \\ \beta^*=0.09 \quad k=0.41 \quad \gamma_1=\dfrac{\beta_1}{\beta^*}-{}^{\sigma_{\omega 1}k^2}/_{\sqrt{\beta^*}} \end{cases}$$

[2]Turbulence model constants are not indexed on List of Symbols

$$C_{k\varepsilon}\begin{cases} \sigma_{k2}=1 & \sigma_{\omega 2}=0.856 & \beta_2=0.0828 \\ \beta^*=0.09 & k=0.41 & \gamma_2=\dfrac{\beta_2}{\beta^*}-{}^{\sigma_{\omega 2}k^2}/_{\sqrt{\beta^*}} \end{cases}.$$

The constants $C_{k\omega SST}$ will result of a combination between $C_{k\omega}$ and $C_{k\varepsilon}$ that reads

$$C_{k\omega SST}=F_1C_{k\omega}+(1-F_1)C_{k\varepsilon} \tag{2.34}$$

and the model closes with the following relations

$$F_1=tanh\left\{\left\{\left[max\left(\frac{\sqrt{k}}{\beta^*\omega d},\frac{500\nu}{d^2\omega}\right),\frac{4\sigma_{\omega 2}k}{CD_{k\omega}d^2}\right]\right\}^4\right\}, \tag{2.35a}$$

$$F_2=tanh\left[\left[max\left(\frac{2\sqrt{k}}{\beta^*\omega d},\frac{500\nu}{d^2\omega}\right)\right]^2\right], \tag{2.35b}$$

$$CD_{k\omega}=max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i},10^{-10}\right), \tag{2.35c}$$

$$\nu_{Turb}=\frac{a_1k}{max(a_1\omega,\Omega F_2)}, \tag{2.35d}$$

being $\nu_{Turb}$ the kinematic turbulent viscosity.

The derivation of Eq. 2.32 and 2.33 comes from a transformation of the $k-\varepsilon$ model to a $k-\omega$ formulation with the following relation

$$\omega=\frac{\varepsilon}{\beta^*k}. \tag{2.36}$$

and then a combination of both $k-\varepsilon$ and $k-\omega$ models with a function $F_1$ that acts like a switch in the inner and outer regions of the boundary layer. Full details are given by Davidson [35].

There are variations of the model with different approaches [36] [37].

## 2.5 Space discretization

The discretization method implemented in these transonic density-based libraries is known as method of lines [38]: it allows the selection of numerical methods with different accuracy for time and space.

The interpolation formulae of the inviscid terms is done with Van Leer's MUSCL [14] which defines them as second-order accurate in space.

Taking the Godunov integral average for a specific quantity $\varphi_i^n$ in cell $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, the piece-wise linear reconstruction of MUSCL is given by

$$\varphi_i(x) = \varphi_i^n + \frac{(x - x_i)}{\Delta x}\Delta\varphi, \quad x \in [0, \Delta x] \tag{2.37}$$

where $\dfrac{\Delta\varphi}{\Delta x}$ is a chosen slope of $\varphi_i(x)$ in cell $I_i$. Figure 2.5 shows the average and its linear reconstruction.

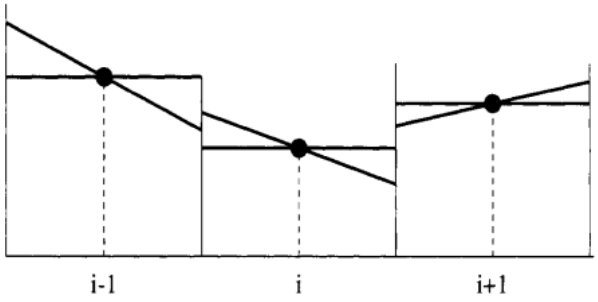



Figure 2.5: Piece-wise linear reconstruction for three successive cells (after Toro [5])

It is also of great importance the extreme points of $\varphi_i(x)$, they will define the right $\varphi_{i+1/2}$ and left $\varphi_{i-1/2}$ states introduced in Section 2.3.2 and are given by

$$\varphi_{i-1/2} = \varphi_i^n - \frac{1}{2}\Delta\varphi, \quad \varphi_{i+1/2} = \varphi_i^n + \frac{1}{2}\Delta\varphi. \tag{2.38}$$

The slope is read

$$\Delta\varphi = \frac{1}{2}(1+\omega)\Delta\varphi_{i-1/2} + \frac{1}{2}(1-\omega)\Delta\varphi_{i+1/2} \tag{2.39}$$

with

$$\Delta\varphi_{i-1/2} \equiv \varphi_i^n - \varphi_{i-1}^n, \quad \Delta\varphi_{i+1/2} \equiv \varphi_{i+1}^n - \varphi_i^n. \tag{2.40}$$

and $\omega$ is a free parameter in the real interval $[-1, 1]$. Setting $\omega = 0$ leads to a second-order accurate upwind-biased linear interpolation. The MUSCL scheme needs to add more features when in presence of strong gradients. They are called *limiter functions* or *limiters* and are build to reduce the solution wiggles near discontinuities.

In all simulations the Venkatakrishnan [39] limiter is used. It is second order accurate $\omega = 0$ and reduces the reconstructed gradient by the following factor

$$\Psi = \begin{cases} \frac{1}{\Delta_2}\left[\frac{\left(\Delta_{1,max}^2 + \epsilon^2\right)\Delta_2 + 2\Delta_2^2\Delta_{1,max}}{\Delta_{1,max}^2 + 2\Delta_2^2 + \Delta_{1,max}\Delta_2 + \epsilon^2}\right] & if \quad \Delta_2 > 0 \\ \frac{1}{\Delta_2}\left[\frac{\left(\Delta_{1,min}^2 + \epsilon^2\right)\Delta_2 + 2\Delta_2^2\Delta_{1,min}}{\Delta_{1,min}^2 + 2\Delta_2^2 + \Delta_{1,min}\Delta_2 + \epsilon^2}\right] & if \quad \Delta_2 < 0 \\ 1 & if \quad \Delta_2 = 0 \end{cases} \tag{2.41}$$

where

$$\Delta_{1,max} = \phi_{max} - \phi_i \Delta_{1,min} = \phi_{min} - \phi_i. \tag{2.42}$$

In Eq. 2.42, $\phi_{max}$ and $\phi_{min}$ are the maximum and minimum values of all neighbor nodes and the owner itself. The parameter $\epsilon^2$ is meant to control the amount of limiting and it was found that it should be proportional to a local scale length, i.e.

$$\epsilon^2 = (K\Delta h)^3 \tag{2.43}$$

It was found [39] that setting $K = 5$ provided the best convergence properties with the best shock resolution thus this value is used in all simulations.

Other limiters are implemented such as Barth-Jespersen [40] and Minmod [41]. The upgrade of MUSCL with limiter schemes is also called Total Variation Diminishing (TVD) schemes.

The viscid terms are build with central difference discretization which is second order. Therefore, this solver is second order in space although lower order can be used.

## 2.6 Time discretization

### 2.6.1 Steady solver

For each control volume, Eq. 2.1 can be written in a semi discrete form

$$\frac{\partial \upsilon \mathbf{U}}{\partial t} = -\mathbf{R}(\mathbf{U}) \tag{2.44}$$

with $\upsilon$ for volume and the residual vector $\mathbf{R}$ being the full spacial discretization and a non-linear function of the conservative state vector. By integrating Eq. 2.44 in

time, one can reach a steady state $\mathbf{R}(\mathbf{U}) = 0$ or an unsteady solution. Although both solutions are available in the density-based libraries, just the former will be presented in detail.

It is important to note that time accuracy in steady cases is irrelevant. The accuracy of the solution only depends on spacial discretization making the advantage of method of lines clearer. From the other side, in unsteady flows, the aim is to get the highest temporal accuracy possible not forgetting stability and computational costs issues.

Runge-Kutta [42] time stepping is available and will used in all simulations. It is given by

$$\begin{aligned}
\mathbf{U}^0 &= \mathbf{U}^n \\
\mathbf{U}^1 &= \mathbf{U}^0 - \alpha_1 \frac{\Delta t}{\Delta x}\mathbf{R}(\mathbf{U}^0) \\
\mathbf{U}^2 &= \mathbf{U}^0 - \alpha_2 \frac{\Delta t}{\Delta x}\mathbf{R}(\mathbf{U}^1) \\
\mathbf{U}^3 &= \mathbf{U}^0 - \alpha_3 \frac{\Delta t}{\Delta x}\mathbf{R}(\mathbf{U}^2) \\
\mathbf{U}^4 &= \mathbf{U}^0 - \alpha_4 \frac{\Delta t}{\Delta x}\mathbf{R}(\mathbf{U}^3) \\
\mathbf{U}^{n+1} &= \mathbf{U}^4
\end{aligned} \tag{2.45}$$

and the following coefficients were used in all simulations

$$\begin{aligned}
\alpha_1 &= 0.11 \\
\alpha_2 &= 0.2766 \\
\alpha_3 &= 0.5 \\
\alpha_4 &= 1.0
\end{aligned} \quad . \tag{2.46}$$

By reducing the residual of equation 2.44 towards machine zero, a steady state will be reached. Convergence measures will be introduced in Section 2.8.

### 2.6.2 Unsteady solver

The unsteady solver will only be used in Chapter 3 to solve the Riemann problems.

The approach to calculate an unsteady problem used in this solver called *dual time-stepping* (DTS) and it a is common practice. Explicit Runge-Kutta and convergence acceleration techniques are not time accurate and so, Jameson [43] introduces a pseudo-time $\tau$ solver that will reformulate Eq. 2.44 the following way

$$\frac{\partial \upsilon \mathbf{U}}{\partial \tau_t} = \frac{\partial \upsilon \mathbf{U}}{\partial t} + \mathbf{R}(\mathbf{U}) = \mathbf{R}^*(\mathbf{U}). \tag{2.47}$$

$\mathbf{R}^*(\mathbf{U})$ is a new residual and the physical time derivative $\frac{\partial}{\partial t}$ is approximated by a second order accurate three-point backward difference [44]. Hence,

$$\frac{\partial \upsilon \mathbf{U}}{\partial \tau_t} = \frac{3\upsilon^{n+1}\mathbf{U}^{n+1} - 4\upsilon^n \mathbf{U}^n + \upsilon^{n-1}\mathbf{U}^{n-1}}{2\Delta t} + \mathbf{R}(\mathbf{U}^{n+1}) = \mathbf{R}^*(\mathbf{U}^{n+1}), \tag{2.48}$$

with the steady state techniques mentioned above being used to reduce the residual $\mathbf{R}^*(\mathbf{U})$. When the pseudo-time solver converges $\frac{\partial \Omega \mathbf{U}}{\partial \tau_t} \approx 0$, the physical time step advances.

## 2.7 Accelerating convergence techniques

Here is described the only accelerating convergence technique implemented in the density-based libraries: the *Local Time Stepping* (LTS).

The LTS consists in using the maximum allowable time step taking the dissipative and convective contributions. Hence, according to Blazek [44] they can be given by

$$\Delta t_d = max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\left(\frac{\mu_{Lam}}{Pr_{Lam}} + \frac{\mu_{Turb}}{Pr_{Turb}}\right)\frac{(\Delta x)^2}{\upsilon}, \quad \Delta t_c = (|U| + c)\,\Delta x \tag{2.49}$$

and Andrea [45] proposes the following final time step

$$\Delta t = C_{cfl}\left(\frac{\Delta t_d \Delta t_c}{\Delta t_d + \Delta t_c}\right). \tag{2.50}$$

Other accelerating convergence techniques are *Implicit residual smoothing* (IRS) and *Multigrid*. IRS is a technique to diminish the dependence of time steps from CFL number and consequently raise the maximum time step. Applying IRS in the explicit multistage scheme mentioned in section 2.6 was introduced by Jameson [46] and consists in replacing the residual for an average of neighbor's residuals.

Multigrid is a powerful technique and the literature about the subject is vast [47] [45] [44]. Basically, it creates coarse mesh levels with an agglomeration technique [48] that speed up the computation of residuals. This way, much of the work is done in a coarse mesh and then can be extrapolated to the fine mesh. Hence, convergence is reached with significant less CPU time [49].

## 2.8 Convergence in metric spaces $L^2$ and $L^\infty$

Metric spaces are a useful tool to see how much converged a solution is. In fact, it is the metric on $\mathbb{R}$ that defines the concept convergence of a sequence [50].

A metric space is a set $X$ of elements whose nature is left unspecified, with a distance function $d$ defined on it [50]. It is convenient to note that the pair $(X, d)$ is a metric space if $d$ satisfies some conditions.

The $L^\infty(X, d)$ metric space is the set $X$ of all bounded sequences with the metric

$$d(x, y) = \sup_{j \in \mathbf{N}} |\xi_j - \eta_j| \tag{2.51}$$

where $\xi_j$ and $\eta_j$ are sequences of the set. Here it is enough to say, without the proof [50], that $L^\infty$ is a complete metric space (*Banarch space*). A *Cauchy sequence* is a sequence $\{x_n\}_{n=1}^{\infty}$ such that

$$\forall \epsilon > 0 \quad \exists N_\epsilon \Rightarrow d(x_n, x_m) < \epsilon \quad \forall n, m < N_\epsilon. \tag{2.52}$$

Again, without the proof, it follows that all Cauchy sequences converge in complete spaces.

The $L^p$ metric space is a set of sequences $x = (\xi_j) = (\xi_1, \xi_2, \cdots)$ such that $|\xi_1|^p + |\xi_2|^p + \cdots$[3] converges. The metric reads

$$d(x, y) = \left( \sum_{j=1}^{\infty} |\xi_j - \eta_i|^p \right)^{1/p}. \tag{2.53}$$

Setting $p = 2$ one arrives to the well known *Hilbert sequence space* $L^2$ [51].

Applying these concepts to CFD and more precisely to residuals, one can make the following analogy. Every cell's residual can be seen as a sequence and the whole mesh is the set of sequences: it is true that at each step they have a value. If a steady state is to be reached, LMSC is true and we can measure the *steadiness* with Eq. 2.53. Moreover, LMSC is a Cauchy sequence thus we are sure it will converge in $L^\infty$.

In a practical way, by plotting the metrics of both $L^2$ and $L^\infty$, some results follow naturally and can be mathematically proven:

[3] $L^p$ metric space condition (LMSC)

- if the problem is steady, the residual can definitely be interpreted as a sequence in $L^2$ and its metric will converge to $d(x, y) = 0$;
- the metric $d(x, y)$ in $L^2$ gives a qualitative measure of *steadiness*;
- if the residual can be interpreted as a sequence in $L^2$, LMSC will converge in $L^\infty$;
- the solution is said to be converged if and only if both the $L^\infty$ and $L^2$ metrics reduces its order to an arbitrarily value $\epsilon$;
- convergence in $L^2$ implies convergence in $L^\infty$ but the inverse need not to be true.

## 2.9 Comments on OpenFOAM

OpenFOAM is an open-source software which uses an unstructured mesh connection by nature, thus all numerical schemes must be implemented accordingly. Being unstructured makes it very versatile when dealing with cell shapes [52] which allows higher mesh quality in complex geometries and easier local refinement.

Its creator Weller and chief architect Jasak planed it with an *object oriented programming* technique [53] that allows the user to extend functionalities as he pleases in a consistent manner, thus being highly customizable. OpenFOAM is available under a GNU General Public License and it is free.

Parallel computation is handled via a domain decomposition method [54] and is often available in all solvers. The steady solver used in this thesis supports parallelization except when using the mixing plane[4] and when performing unsteady simulations. Finally, both of these solvers run with OpenFOAM's version 1.6-ext.

[4] Difficulties and solutions will be given in Chapter 5

# CHAPTER 3

# RIEMANN PROBLEM VALIDATION

This chapter checks the performance of the ARS used in this thesis and, consequently, the ability of Borm's code implementation to catch shock, rarefaction and contact waves. Moreover, second order extensions will be compared to first order and the performance of the limiters will be accessed.

Recalling Section 2.3, the Riemann problem in the Euler equations is a discontinuous initial valued problem with a left $U_L = [\rho, U, p]^T$ and right state $U_R$ whose solution is approximated with the propagation of three waves through the medium. This approximation is done by an ARS and the one being used is the HLLC. Physically, this situation is illustrated as two states separated by a diaphragm: at $t = 0$ the diaphragm is broken. Solutions are naturally evaluated at $t > 0$.

The computational domain used for all the following tests is a one dimensional mesh defined such that $x \in [0, 1]$. It is discretized in 100 cells and the notation $x_{diaph}$ denotes the position of the diaphragm. All tests are compared with the exact solution of Riemann-problem provided by Toro and all data is non-dimensional.

## 3.1 Sod's shock tube

The so called shock tube problem was heavily studied by Sod [55] and consists in a right shock wave, a right contact wave and a left sonic rarefaction wave. Its initial data

is composed by $U_L = [1, 0, 1]^T$ , $U_R = [0.125, 0, 0.1]^T$.

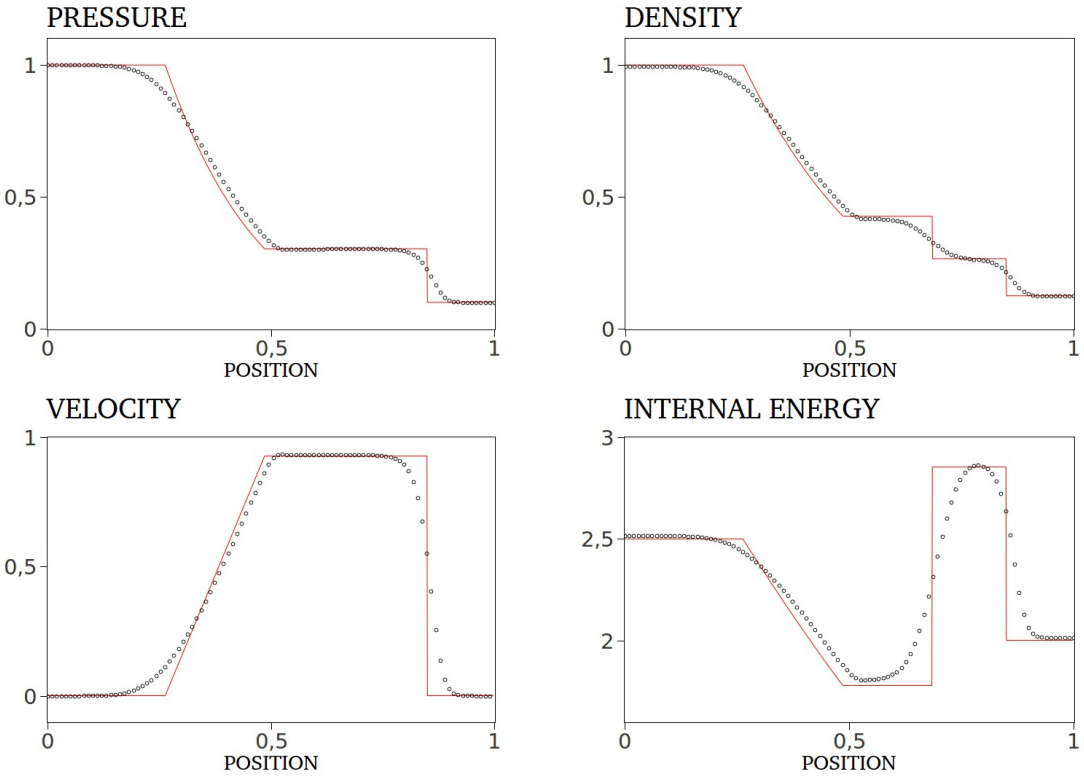



Figure 3.1: First order HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 0.2$ and $x_{diaph} = 0.5$.

*Sod's shock tube*

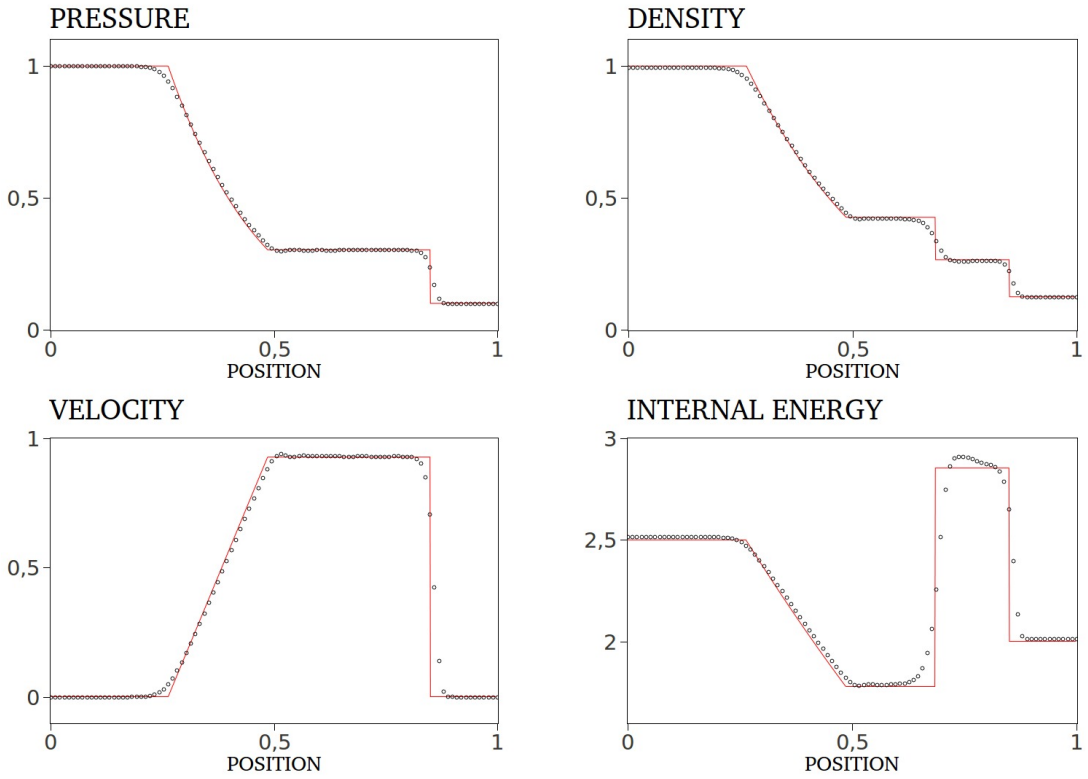



Figure 3.2: Second order HLLC solver with Van Albada slope limiter. The numerical (icon) and exact (line) solutions are shown for $t = 0.2$ and $x_{diaph} = 0.5$.

Results for first and second order simulations are plotted in Figures 3.1 and 3.2, respectively. Across shock waves it is usual to have discontinuities in all variables while in contact waves, velocity and pressure are continuous. Regarding internal energy across contact waves, continuity isn't observed which is natural as this type of wave separate two different states. This test is meant to evaluate the entropy condition across shock established in the last half of 19th century by Rankine and Hugoniot [56] [30] [57].

## 3.2 123 problem

This Riemann problem consists of two rarefaction waves and a trivial stationary contact discontinuity. The region between the two waves is characterized by very low pressure, close to vacuum, and this test is usually done to evaluate the solver's ability to deal with low density flows. The initial data is composed by $U_L = [1, -2, 0.4]^T, U_R = [1, 2, 0.4]^T$ with results for first and second order presented in Figures 3.3 and 3.4, respectively.

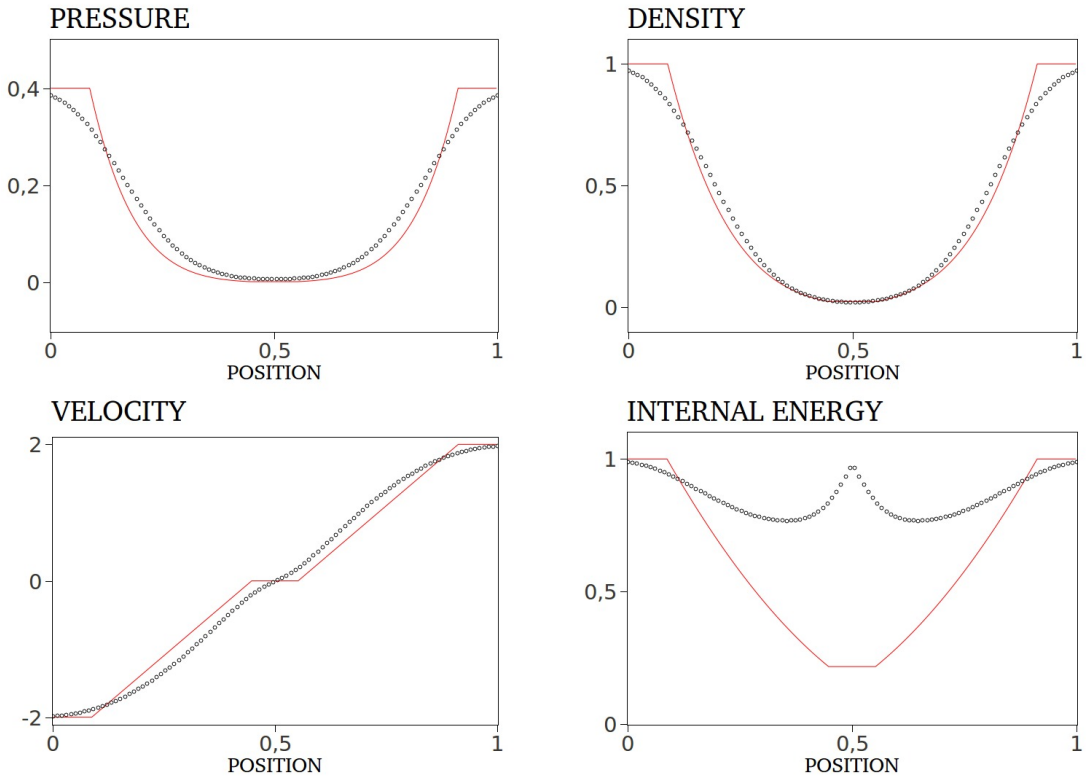



Figure 3.3: First order HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 0.15$ and $x_0 = 0.5$.

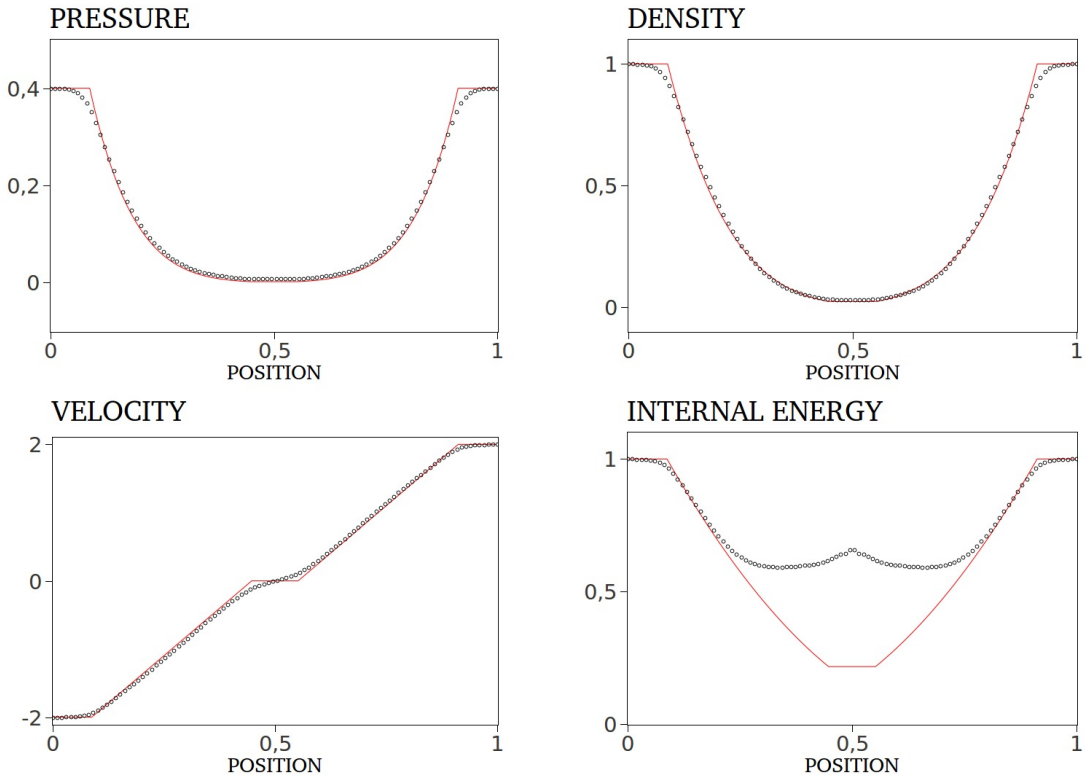



Figure 3.4: Second order HLLC solver with Van Albada slope limiter. The numerical (icon) and exact (line) solutions are shown for $t = 0.15$ and $x_{diaph} = 0.5$.

It can be noted that near zero pressure and density, the HLLC behaves quite wrong as already mentioned by Toro. This is coherent with the fact that Godunov-like solvers fail in this kind of test. Nevertheless, it's remarkable the improvement achieved with second order extensions and results are in agreement with Toro's.

## 3.3 Collision of 2 shocks

The wave system presented in Figures 3.5 and 3.6 has resulted from a collision of two shocks. Three discontinuities arise propagating to the right: two shocks and a contact

wave being the right shock the fastest signal. Left shock is slow and it is defined with two grid points as can be seen in pressure plot. This is coherent with the fact that Godunov-like methods resolve perfectly slow moving shocks. The contact wave is heavily smeared where the fast shock is smeared in five or six grid points. This is also a test to see the robustness and accuracy of the solver as different discontinuities are present. The initial data is composed by $U_L = [5.99924, 19.5975, 460.894]^T$ , $U_R = [5.99242, -6.19633, 46.0950]^T$.

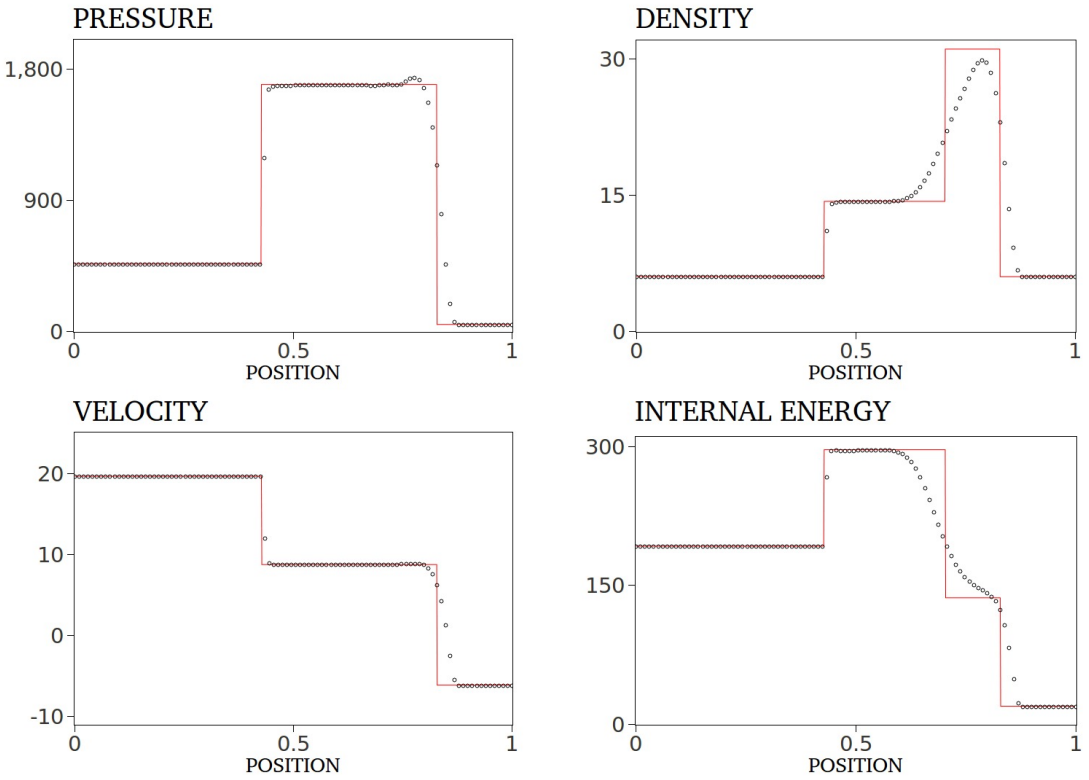



Figure 3.5: First order HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 0.035$ and $x_{diaph} = 0.4$.

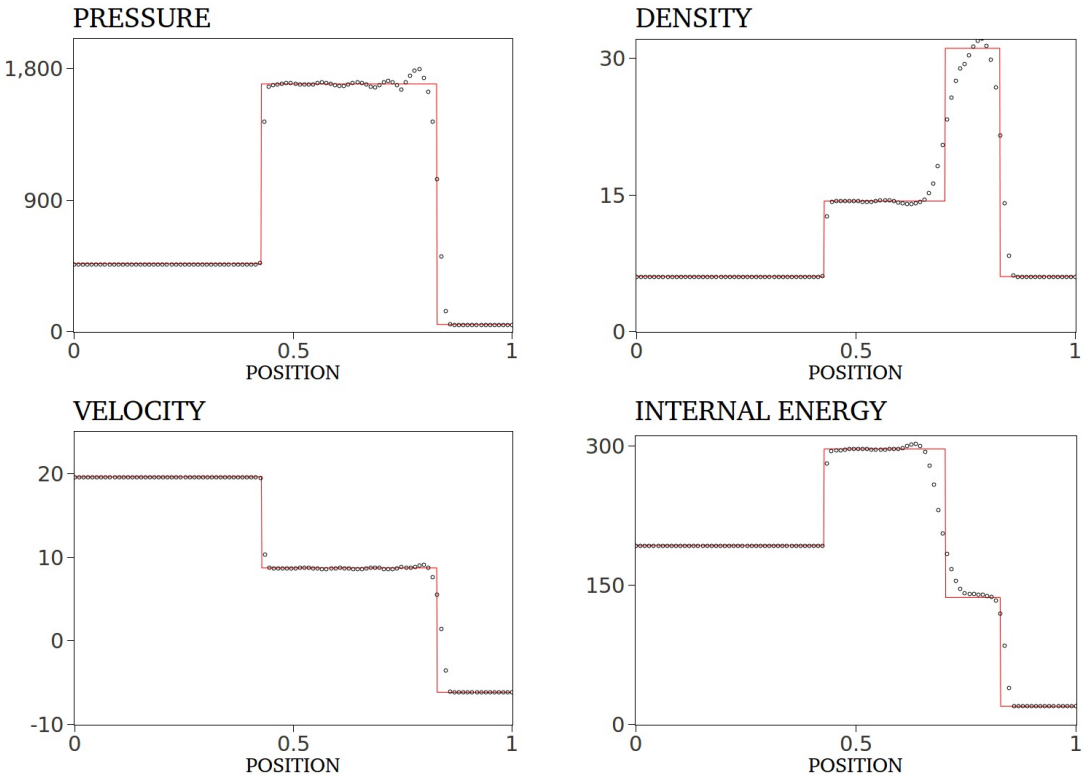



Figure 3.6: Second order HLLC solver with Van Albada slope limiter. The numerical (icon) and exact (line) solutions are shown for $t = 0.035$ and $x_{diaph} = 0.4$.

## 3.4 Stationary contact

The main reason to perform this test is to access the ability to catch slowly moving contact discontinuities. Stationary contact test initial data reads $U_L = [1, -19.5975, 1000]^T$ , $U_R = [1, -19.5975, 0.01]^T$ and the exact solution is composed by a left rarefaction wave, a right traveling shock wave and a stationary contact discontinuity. Figures 3.7 and 3.8 show the good accuracy, almost exact, when dealing with stationary discontinuities.

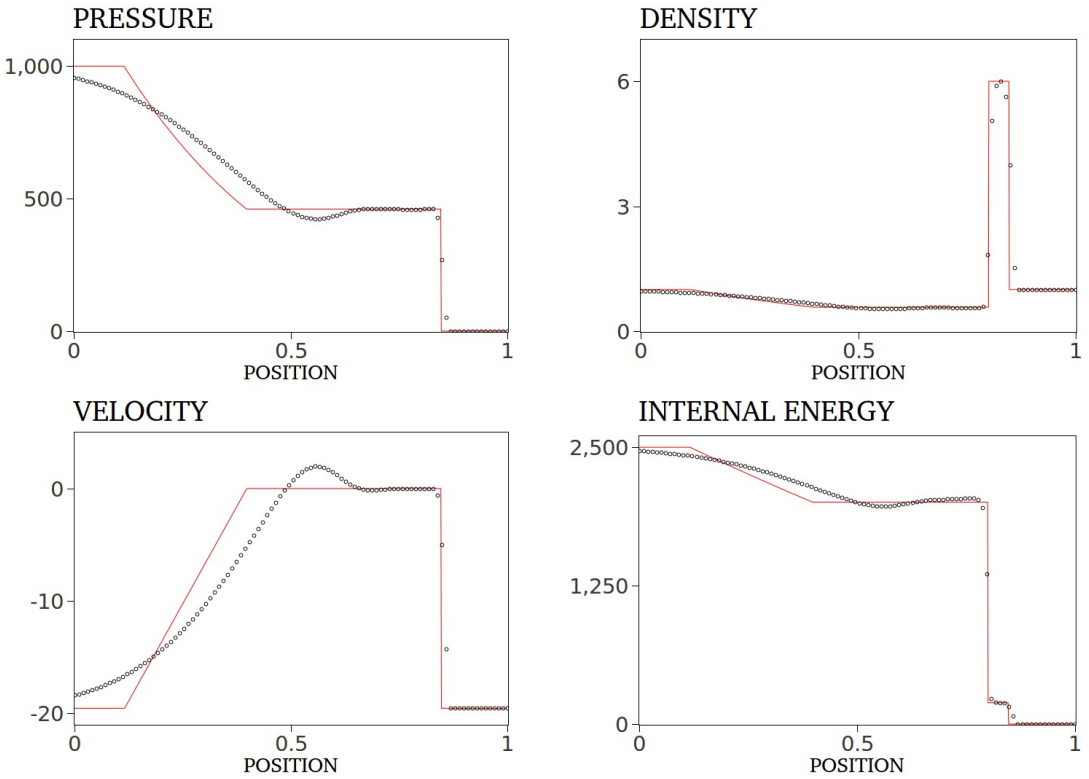



Figure 3.7: First order HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 0.012$ and $x_{diaph} = 0.8$.

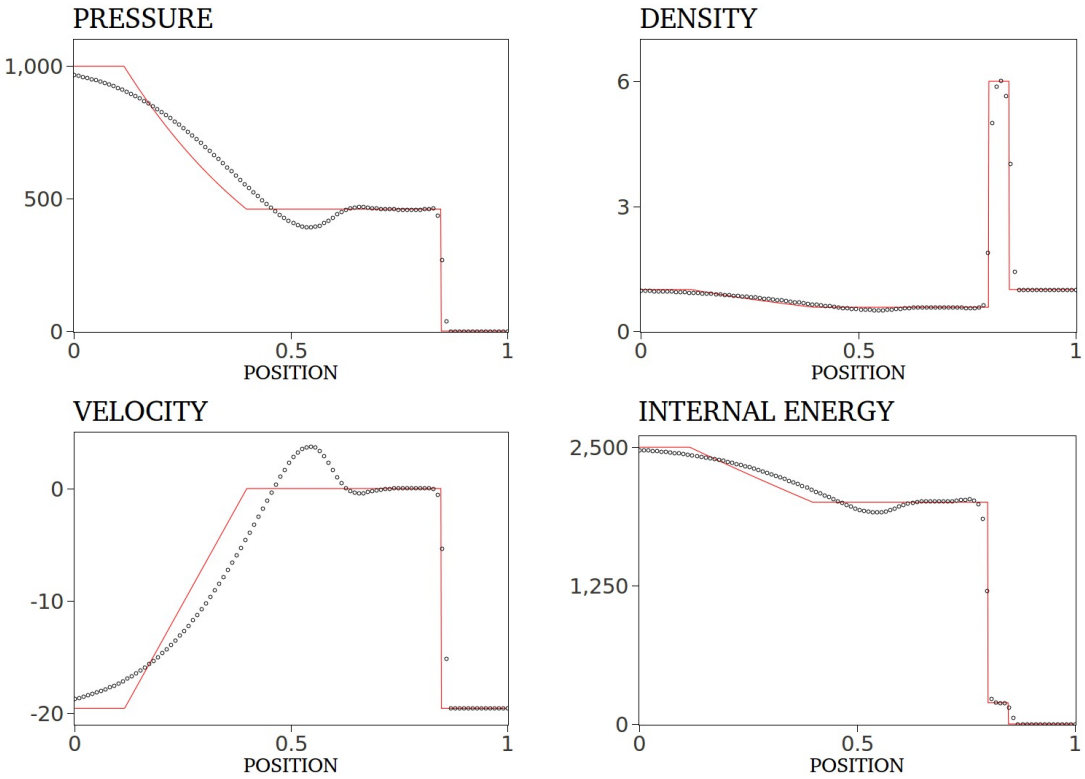



Figure 3.8: Second order HLLC solver with Van Albada slope limiter. The numerical (icon) and exact (line) solutions are shown for $t = 0.012$ and $x_{diaph} = 0.8$.

## 3.5 Extra tests

Two more tests are done in this chapter, Test 6 and Test 7. These tests access the performance for contacts, shear waves and vortices. For Test 6, the initial data reads $U_L = [1.4, 0, 1]^T, U_R = [1, 0, 1]^T$ and the wave system is composed by an isolated stationary contact wave. For Test 7 the initial data is $U_L = [1.4, 0.1, 1]^T, U_R = [1, 0.1, 1]^T$ and the solution consists in a isolated slowly moving contact.

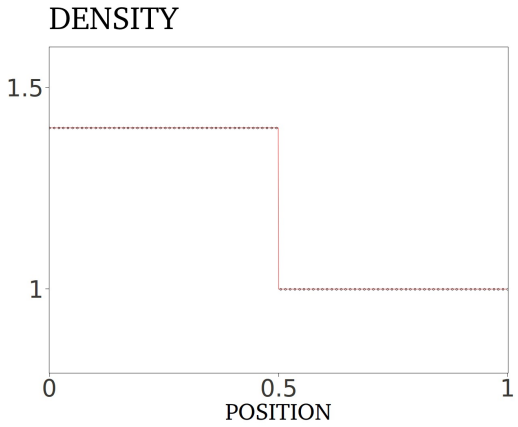



Figure 3.9: First and second order HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 2$ and $x_{diaph} = 0.5$.

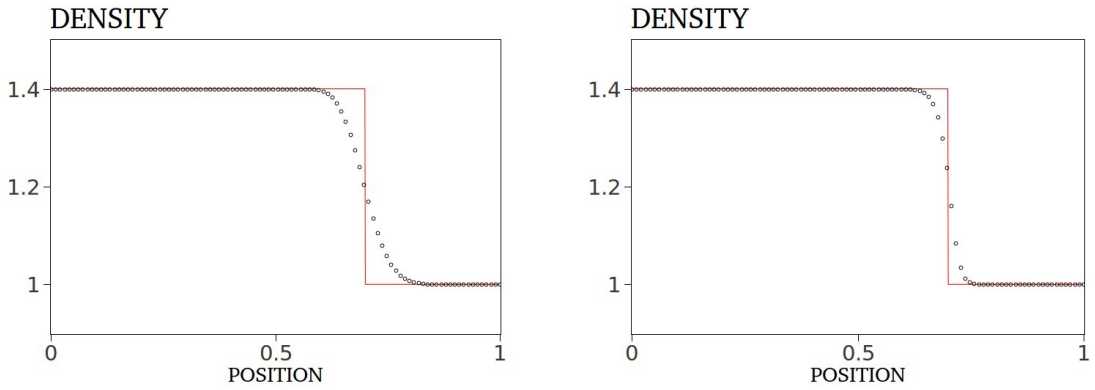



Figure 3.10: First (left) and second order (right) HLLC solver. The numerical (icon) and exact (line) solutions are shown for $t = 2$ and $x_{diaph} = 0.5$.

Regarding Test 7 plotted in Figure 3.10, second order discretization presents some improvements and results are also in agreement with Toro. Toro [5] points out that these tests have the same results when extending to multi-dimensional problems.

## 3.6 Remarks

This chapter presented the performance of the approximate Riemann solver HLLC used to evaluate the convective term of the Navier-Stokes equations. Results are good in the generality: they achieve perfect accuracy in slowly moving discontinuities and, from the other side, highly smeared results are produced in fast moving contact discontinuities. It is important to note that all results are in agreement with Toro, which means the algorithm is well implemented. Second order extensions help reducing the necessary grid points to define shock discontinuity thus accuracy being gained, however, even the implementation of slope limiters isn't enough to eliminate the wiggles in presence of strong shock. Moreover, because all the above shocks satisfy the entropy condition and the HLLC does a good approximation to the exact solution, it follows that it can be used with safety in transonic flows. More details about the simulation's parameters are given in Appendix B.

# CHAPTER 4

# NASA ROTOR 67

A widespread test case is the NASA Rotor 67 (R67) [6]. NASA stands for National Aeronautics and Space Administration. It is used since the late 80s to test computational algorithms, specially those which include viscous terms. The R67 is a great test case to see the robustness of the density-based code implemented by Borm: the flow is transonic with shock, vortex are created at the *tip-gap* and at the trailing edge (TE).

Figure 4.1 presents the fan R67: it is a low pressure compressor, made with 22 blades. The low pressure compressor has the total of two stages and it is designed for short-haul aircraft. It is characterized by a low aspect ratio geometry and designed to achieve a tip speed of 429 $\mathrm{m\,s^{-1}}$.

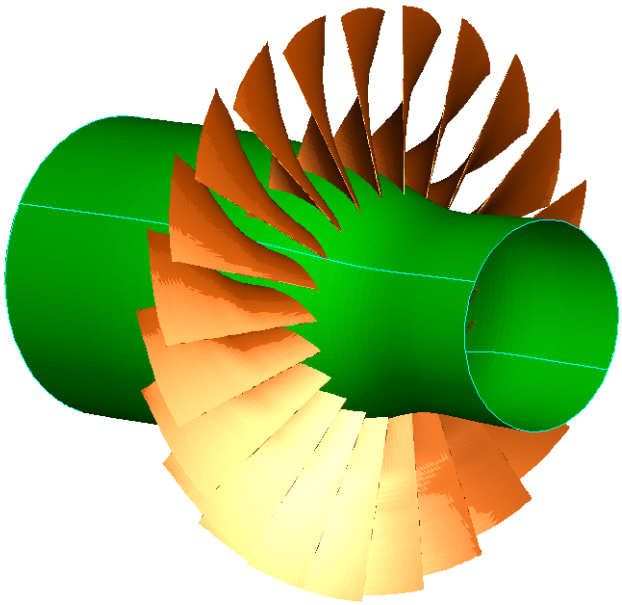


Figure 4.1: Fan rotor

## 4.1 Data

### 4.1.1 Experimental Data

Experimental data is available near stall and near peak efficiency. A laser anemometer (LA) technique was used to collect the tangential and axial velocities. A 1100 windows cover 360 degrees of the rotor providing a single passage to be measured in 50 positions along $\theta$. 30 axial positions crossed with 9 surface revolutions gives a total of $50 \times 30 \times 9$ measuring positions in a rotor passage where results are often presented in *radius*$\times$*theta-meridional* (r$\theta$-M) planes, also called B2B. Figure 4.2 illustrates how the positions of measurements are defined.

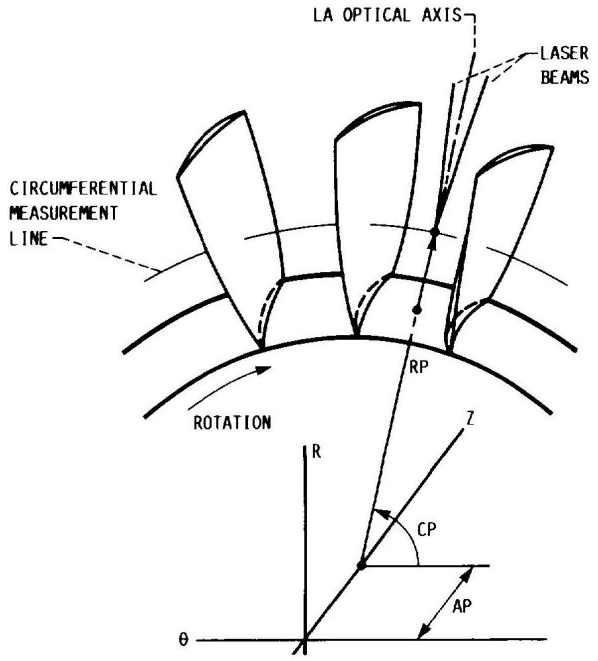



Figure 4.2: Measurement coordinates: radial position (RP), axial position (AP), circumferential position (CP) (after NASA TP2879 [6])

It is important to mention how experimental plots were generated because it will influence how numerical data should be manipulated. There exists two kinds of plots: stream-wise and pitch-wise. As the name implies, stream-wise will follow a streamline (Figure 4.3) and pitch-wise will follow the pitch, each one always in constant % span. Obviously, a stream-wise plot will be defined by constant pitch percentage and pitch-wise plot will be defined by constant chord percentage position. Moreover, when the pitch is defined, consequently so it is the streamline. Thus, 50 windows per passage define a range for the pitch and Figure 4.4 shows how these windows are numbered. Therefore, the starting window number (WNBEG) for LA survey is available for all plots.

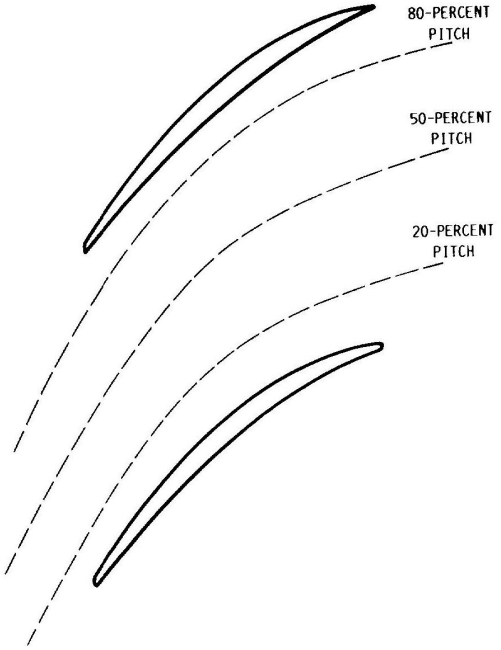



Figure 4.3: Streamlines (after NASA TP2879 [6])

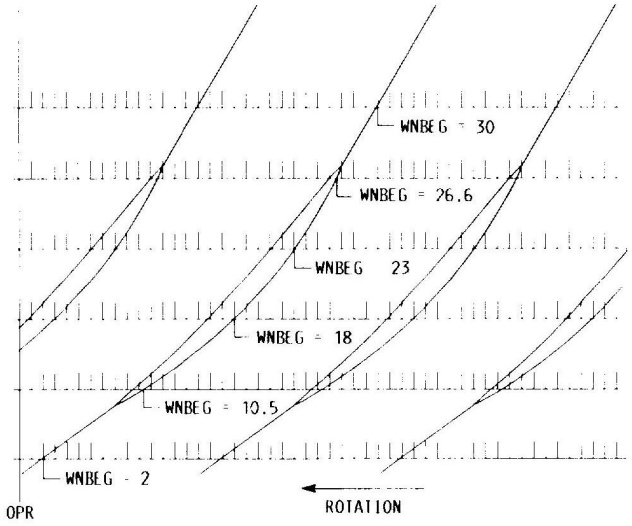



Figure 4.4: Window numbering (after NASA TP2879 [6])

Besides the blade surveys on velocity, radial measurements were taken up and downstream of the rotor. These included $P_0$ and $T_0$ that were used to calculate the overall rotor aerodynamic performance. All data is corrected to National Advisory Commit-

tee for Aeronautics (NACA) standard day conditions at sea level: $T_0 = 288.15\text{K}$ and $P_0 = 101325\text{Pa}$.

### 4.1.2 Numerical Data

Extracting numerical data from a rotor is not straightforward, at least with open-source software. A Cartesian referential isn't appropriate to understand rotating geometries and solutions, while the open source visualizer software *Paraview* doesn't have the necessary tools in the *Graphic User Interface* (GUI) to perform B2B cuts. Moreover, as experimental data is only available in such formats, there is no other way than implementing this tools in order to be able to compare results. Fortunately, Borm [59] developed some python scripts that do the task and work together with *Paraview*.

Despite the effort of Borm, there is no tool that extracts streamlines along span. Therefore, an utility called *Plot on Interception Curves* from *Paraview* is used. We make the span cut being intercepted by a sphere with sufficient radius to approximate a streamline with constant pitch percentage. Figure 4.5 illustrates one attempt considered successful and similar was done for all plots.

To extract pitch-wise lines the task is much easier: taking into account that chord percentage we (and experimental data) mean axial chord, coordinates of leading edge (LE) and TE were taken and extraction follows naturally. If streamlines are difficult and error prone to extract, but very easy to plot, pitch lines are very easy to extract and difficult to plot: the reason is relating the WNBEG with the numerical R$\times\theta$-M plane.

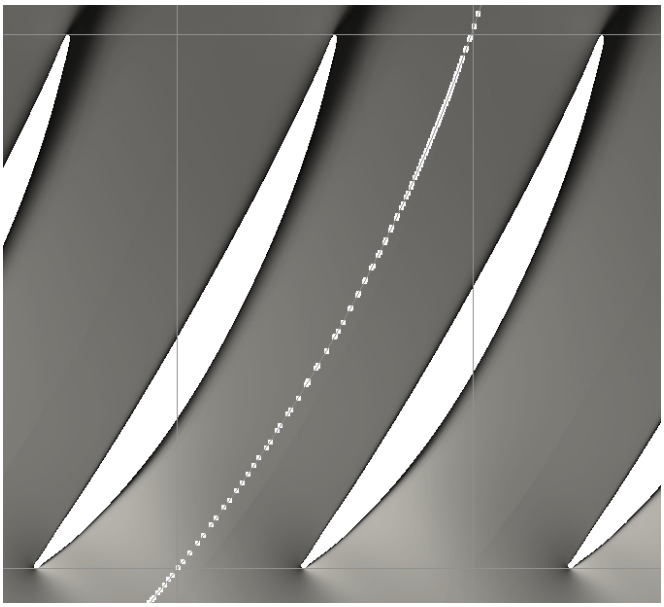


Figure 4.5: Streamline extraction in *Paraview*

As only one passage is simulated, it comes necessary to identify where all the relevant WNBEG are, just in one numerical passage. The solution the author found was to associate the passage pitch range at constant chord percentage to 0.285599 radians, which is equivalent to $^1/_{22} \times 360$ degrees, and then getting the $\Delta$WNBEG between a known point(i.e. blade surface point) and the unknown starting position.

Concerning the overall aerodynamic performance, OpenFOAM® provides a *functionObject* utility called *patchAverage* that acts like the radial surveys up and downstream of the rotor by making an area average or mass-average of variables. In this case, an area average is made because density wasn't measured in the experimental facility. In a related way, another function object called *patchMassFlow* is used to know whether there is conservation of mass between the inlet and outlet of the rotor[1]

## 4.2 Mesh

The rotor and casing geometries are available in [6]. Due to time limitations, a full hexahedral mesh was generated with a commercial mesher.

[1] Useful to test convergence, see Section 4.4.5

*Mesh*

First, a commercial design program was used to introduce blade's surfaces and camber coordinates with angles *beta*: it is the angle of blade's surfaces/camber from meridional direction. The blade surface and camber line were defined with 35 points in the r×$\theta$-M plane and thickness was set symmetric to camber line. All blade sections were linearly stacked at the LE due to limited input parameters when defining the blade position. One can see from Figure 4.6 that stacking isn't linear. Then, the geometry was imported to the commercial mesher which was used to build a mesh of only one passage. Meshes were created so that the fine level had about $8 \times 10^5$ cells, which is a common practice [60].

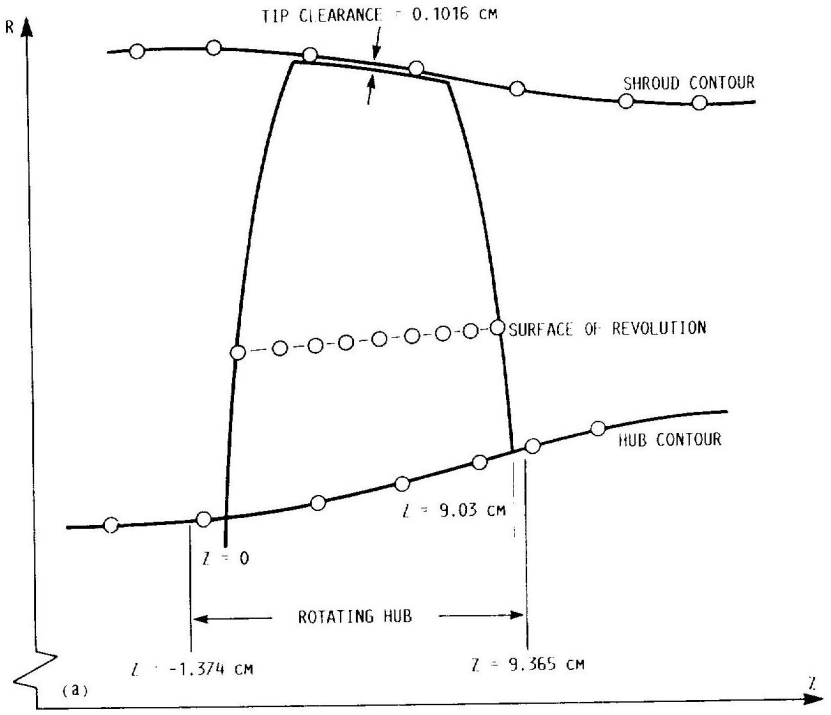


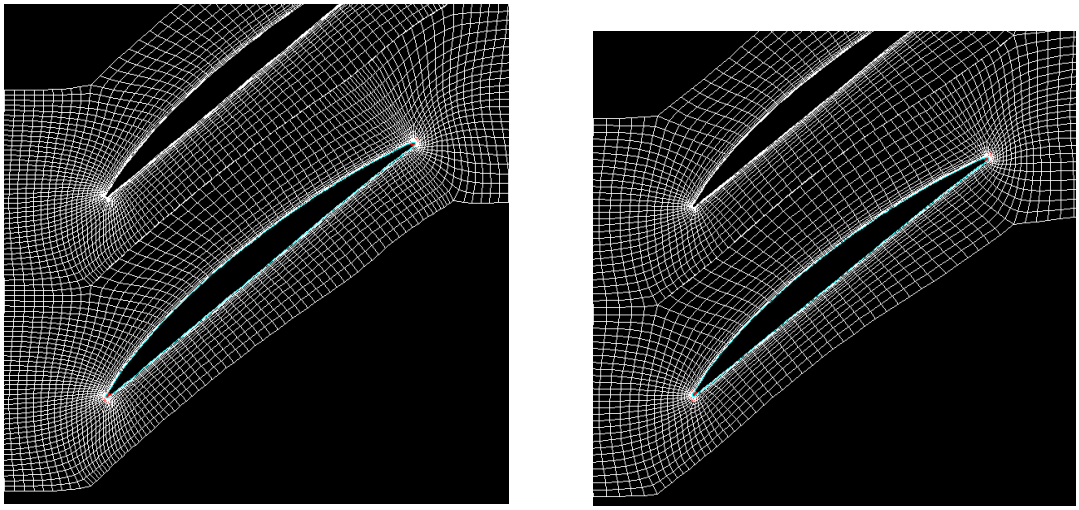

Figure 4.6: RZ plane (after NASA TP2879 [6])

Three mesh levels were created: fine intermediate and coarse. They were converted to OpenFOAM's format with the utility *fluent3DMeshToFoam*. They differ in the cell number by a factor of approximately 2 and tip gap is only meshed in the intermediate level. Tip gap can be seen in Figure 4.6.

Attention was specially given to mesh quality, namely aspect ratio and non-orthogonality which must stay under 1000 and 70, respectively. These values change from code to code and Appendix A shows the quality reports of all the meshes used in this chapter. It is worth mention that those reports were produced with an utility called *checkMesh* from OpenFOAM libraries. Table 4.1 resumes all the meshes used and Figures 4.7, 4.8 and 4.9 present some mesh details.

| Mesh level | ID | Number of cells |
|---|---|---|
| Coarse | R67_C | 272400 |
| Intermediate | R67_I | 417600 |
| Tip Gap | R67_GAP | 484640 |
| Fine | R67_F | 793600 |

Table 4.1: Mesh cell number

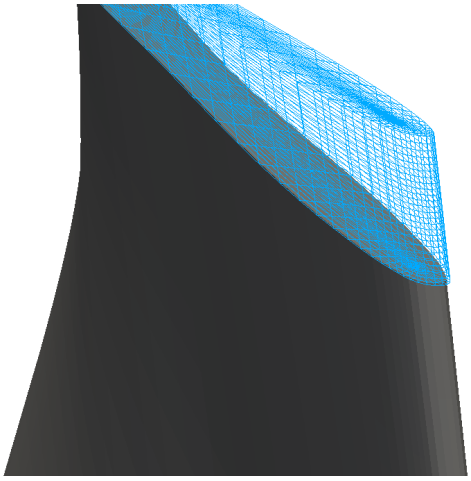


(a) Fine mesh

(b) Coarse mesh

Figure 4.7: Blade-to-Blade mesh detail with repetition

(a) Fan blade (tip gap in blue)



(b) Tip gap detail

Figure 4.8: Mesh details

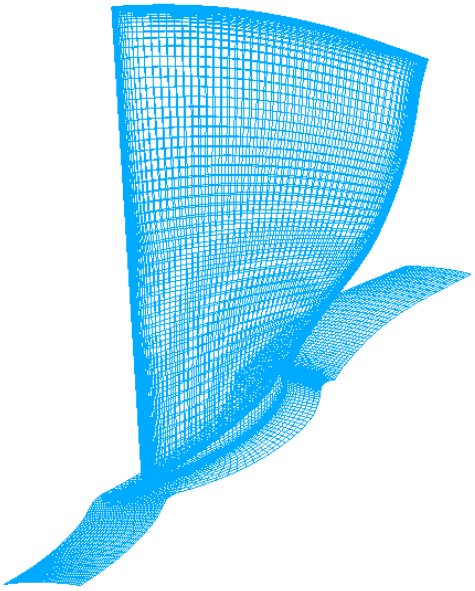


Figure 4.9: 3D mesh (fine)

## 4.3 Numerical setup

### 4.3.1 Solver

The *transonicMRFDyMFoam* solver from the *densityBasedTurbo* library was used in all simulations.

The dictionary named *MRFZones* which denotes *Multiple Rotating Frames* must be specified to the desired blade speed and the direction of rotation is specified with positive or negative value. In all simulations, blade speed was 16042 *rpm* and the value introduced was $-1680\,\mathrm{rad\,s^{-1}}$. This dictionary not only sets the blade velocity but it is where the relative frame of reference is defined in the mesh. The relative frame is thus defined with a set of cells taken from the utility *cellSet* or *regionCellSets*.

Sutherland's law is used in the *transportProperties* dictionary and links the absolute temperature to the dynamic viscosity, which is set to $1.8 \times 10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-1}}$.

As in this chapter we only deal with steady state simulations, the *dynamicMesh* dictionary is set to *staticFvMesh*: because we are in the relative frame, there will be no mesh motion. Additional details about steady state solutions and this particular solver were already given in Chapter 2.

### 4.3.2 Turbulence modeling

It was defined in Chapter 2 that we were following a RANS approach so *turbulenceProperties* dictionary will be set to *RASModel*. Specification of the turbulence model comes with the *RASProperties* dictionary and it is set to *kOmegaSST*. All mesh's $Y^+$ is approximately 1 so that we are solving the boundary layer from somewhere between the viscous (sublayer) and buffer layer (Figure 4.10) above.

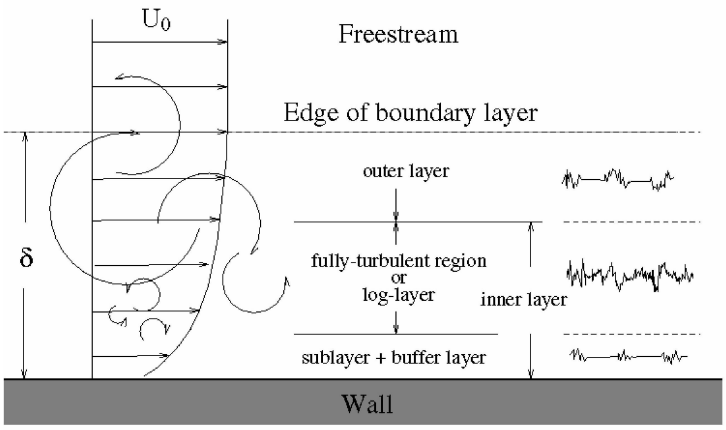



Figure 4.10: Boundary layer regions (after Bakker [7])

### 4.3.3 Boundary Conditions

Standard day conditions are imposed at inlet so that all solution data is ready to be compared with experimental results without further manipulation. To close the problem, a back-pressure will be specified and changed so that a compressor map can be built. All walls are modeled as adiabatic and with no slip conditions. There is a back-flow control that switches the pressure at outlet to "zeroGradient" if indeed exists back-flow. Table 4.2 resumes the inlet boundary conditions used in all simulations.

| inlet boundary conditions |
|---|
| $P_0 = 101325$ Pa |
| $T_0 = 288.15$ K |
| $\mu_{Turb} = 0.0001$ Pa s$^{-1}$ |

Table 4.2: Boundary conditions

### 4.3.4 Mesh connection

In this thesis only one passage is meshed. One passage simulations are common practice as the full annulus simulations most of the time carry huge computational costs. If one considers rotors with 20 to 50 blades, each passage made of $8 \times 10^5$ cells, a full meshed rotor can easily reach 40 million cells becoming intractable in modern computers. In

addition, a General Grid Interface (GGI)[2] [61] is implemented in cyclic boundaries. GGI interfaces behave as boundary conditions and are used when a connection in the mesh is needed but cell faces don't match: an interpolation is done. Because the number of cells in the pressure side is usually different than the suction side, *cyclicGGI* is used as periodic boundary. Thus, the *boundary* file must read

```
periodic
{
    type            cyclicGgi;
    nFaces          4578;
    startFace       465742;
    shadowPatch     periodic_0;
    zone            periodic_faces;
    bridgeOverlap   false;
    rotationAxis    (0 0 1);
    rotationAngle   -16.363636364;
    separationOffset (0 0 0);
}
```

where *periodic* is the name of the boundary condition and it refers to the repetition of the rotor. *type* specifies the kind of boundary condition whether *nFaces* and *startFaces* refer to which faces the boundary is assigned to. *shadowPatch* is the name of the boundary which will make a pair for the repetition, therefore a similar entry must be introduced for *periodic_0*. *zone* defines a global position for the boundary and it is needed for parallel computation. *bridgeOverlap* is a parameter that checks if every face in the specified boundary is covered in the correspondent pair: *false* means it won't allow any mismatch and it is the only situation in this thesis. The last three entries follow naturally from their names.

Other mesh configurations might exist such that there is no GGI in the repetition boundary, i.e. it is a simple cyclic boundary. If that is the case, and a tip-gap is being reproduced, a GGI is always needed and it is usual to implement it in the tip-gap. Nevertheless, whether reproducing or not the tip-gap, GGI will always bring better

[2] In commercial codes is called Full Non Matching Boundary (FNMB)

mesh quality thus it is always used in this thesis.

## 4.4 Results

Overall aerodynamic data will be presented in tabular form while relatives Mach number and flow angle at peak efficiency will be presented in plots. All numerical data will be presented with the correspondent experimental data and both refers to constant[3] blade speed. This section is meant to evaluate the sensibility of the solution with mesh size and mesh type. Turbulence parameters are left constant as well as all boundary conditions except back pressure: it will change to build a compressor map.

It is worth mention that near stall conditions aren't presented because there was difficulty in achieving convergence probably due to the unsteady effects characteristic of stall.

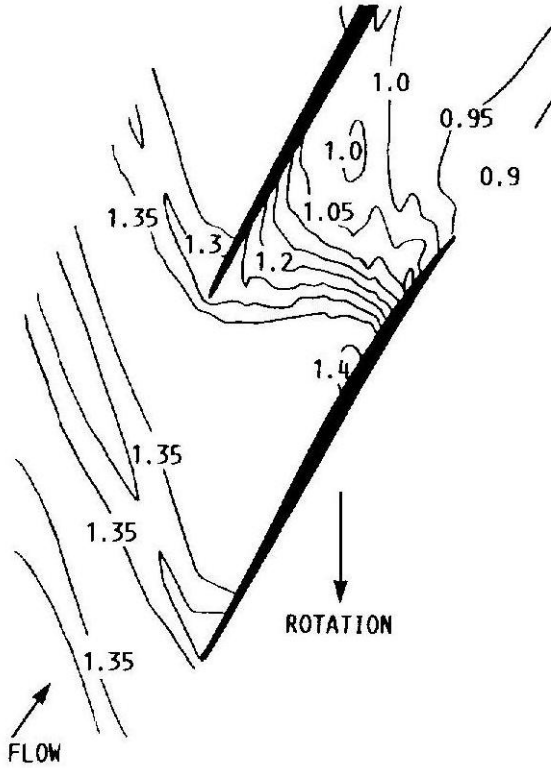
The following section is going to present plots in 10%, 30% and 70% span from the shroud. Most of the plots will present experimental data with error bars. It should be highlighted that although the apparatus of R67 has its uncertainty, characteristic of the measuring tools, error bars will be used to present many measurements in the same position. In fact, those with the largest error bars correspond to the presence of unsteady phenomena and numerical results are, a priori, expected to fail. Moreover, pitch-wise plots are expected to give poorer results as LE and TE are not exactly reproduced as it was mentioned in section 4.2. Finally, all simulations whose results are presented here had a value of 115000 Pa as back-pressure, to better compare the different meshes.

### 4.4.1 Relative flow angle and relative Mach number

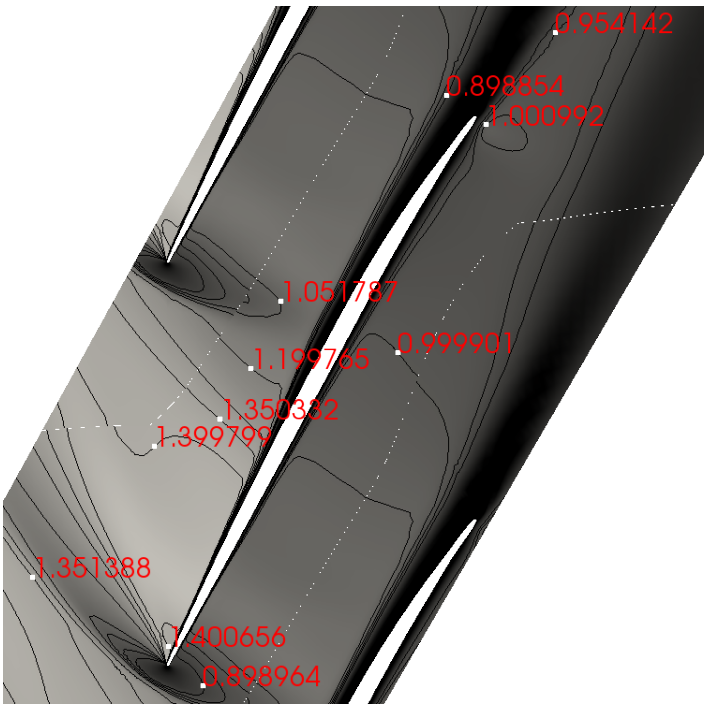
As was mentioned earlier, we expect to see a deceleration of relative velocities and a decrease of relative flow angle along axial chord: it will prove that our problem is well defined. Figures 4.11, 4.19 and 4.27 present the rotor's Mach number structure at 10%, 30% and 70% span. Stream-wise and pitch-wise distributions of relative Mach

[3] LA experimental readings with blade speed drift bigger than 0.4% are excluded.

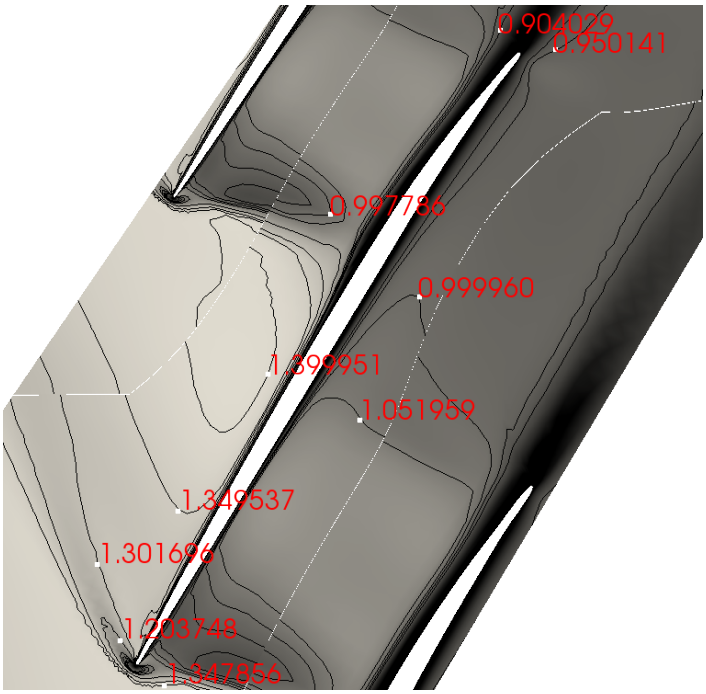
number and relative flow angle are presented in Figures 4.12-4.18 for 10% span, Figures 4.20-4.26 for 30% span and Figures 4.28-4.34 for 70% span.




(a) Experimental




(b) Numerical (R67_F)




(c) Numerical (R67_GAP)

Figure 4.11: Mach number plot at 10% span from shroud near peak efficiency

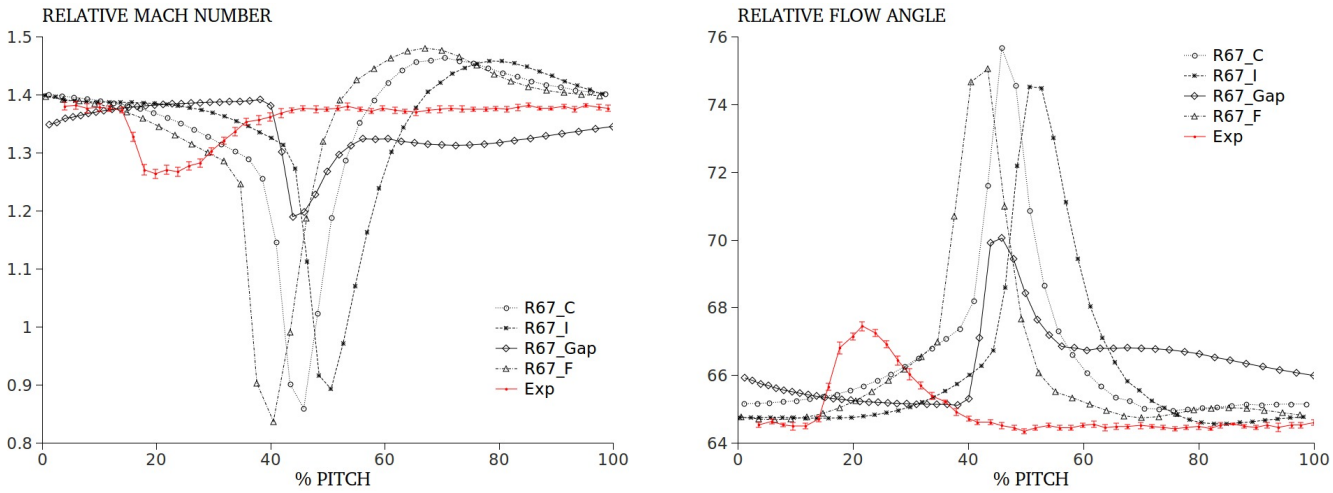



Figure 4.12: Plot of $-5.4\%$ chord at 10% span

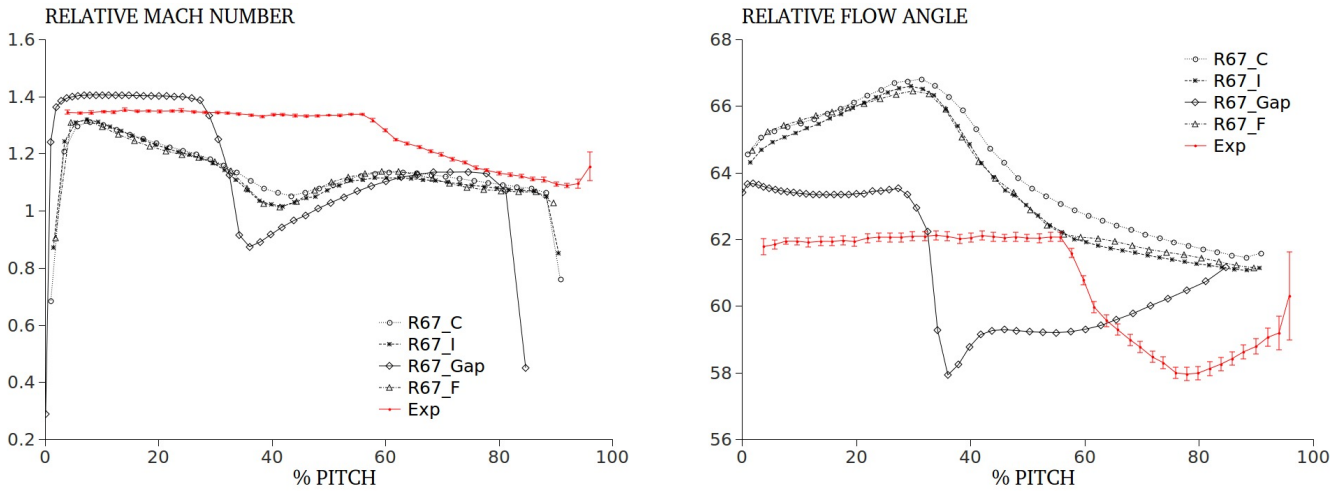



Figure 4.13: Plot of 30% chord at 10% span

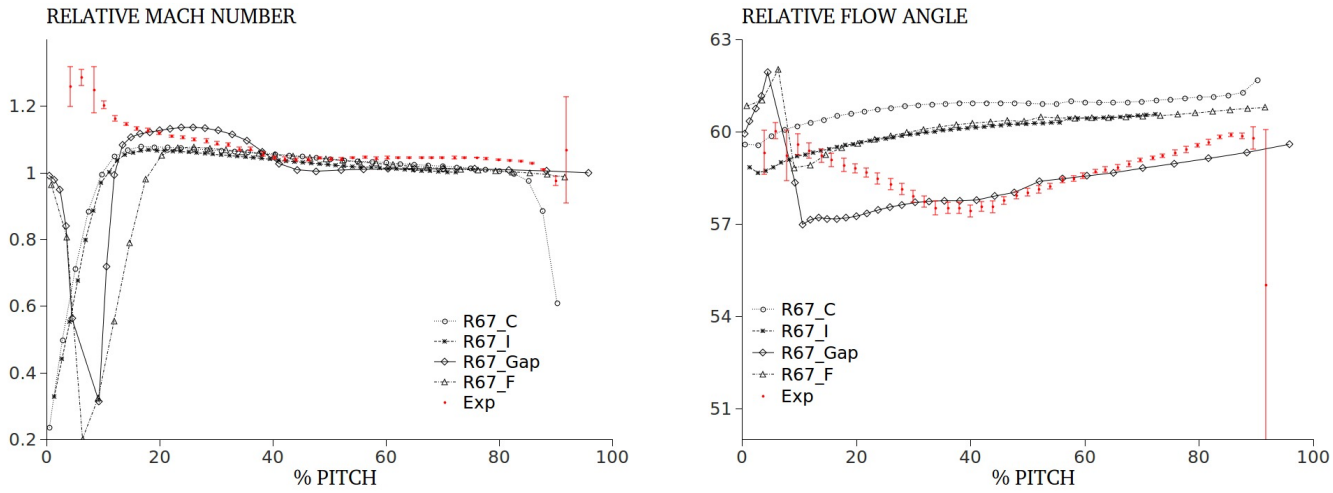



Figure 4.14: Plot of 70% chord at 10% span

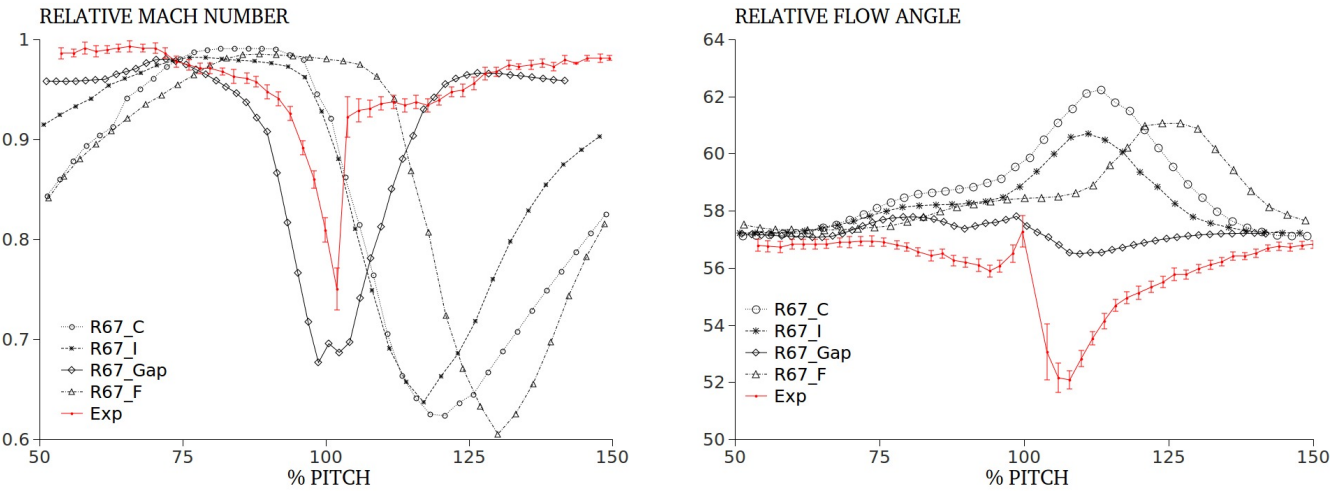



Figure 4.15: Plot of 124% chord at 10% span

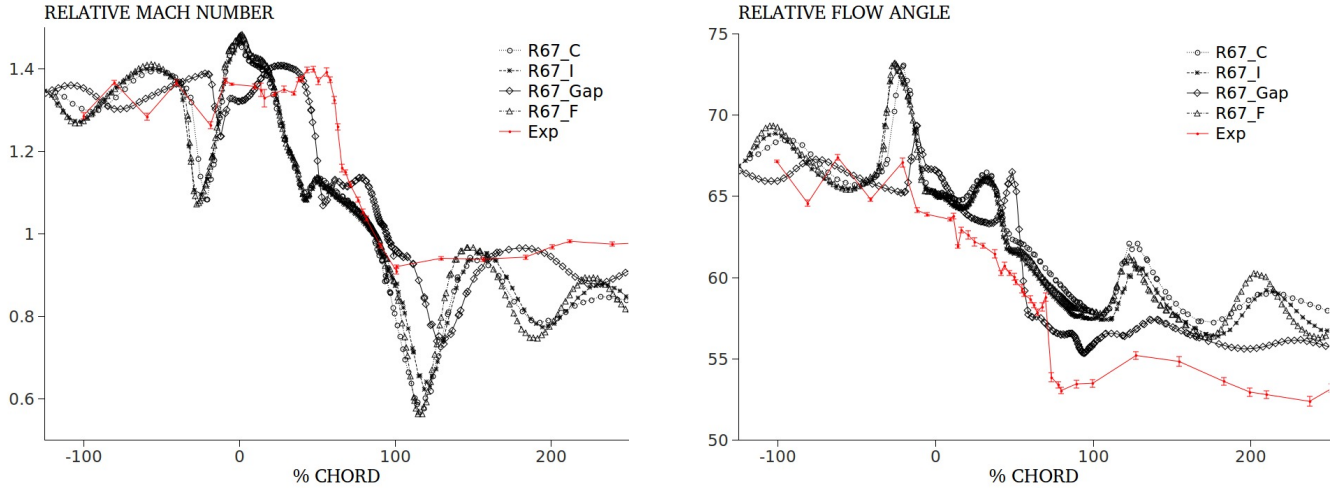



Figure 4.16: Plot of 20% pitch at 10% span

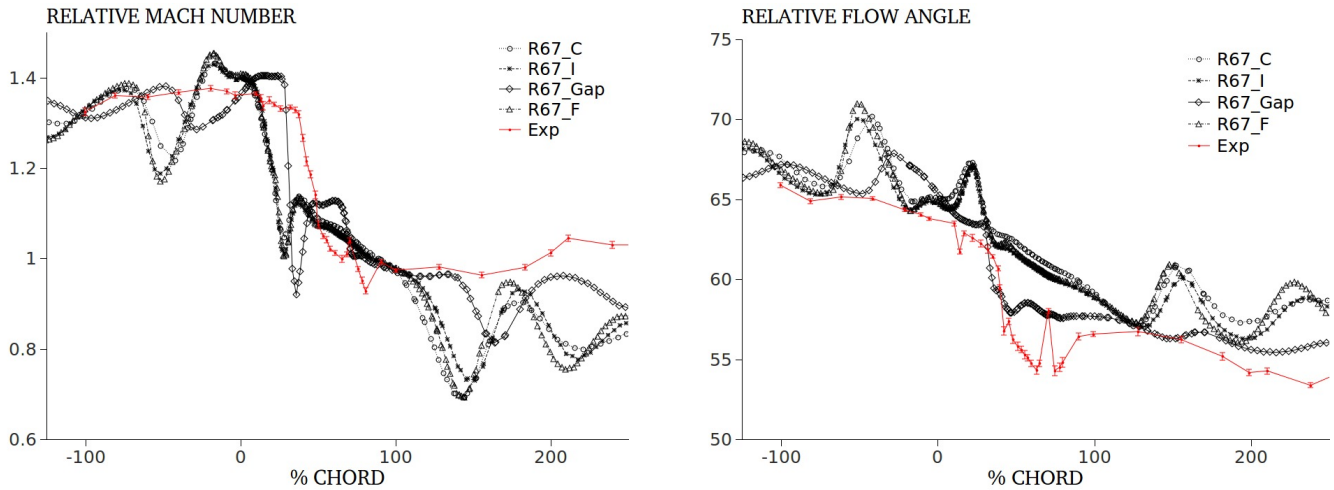



Figure 4.17: Plot of 50% pitch at 10% span

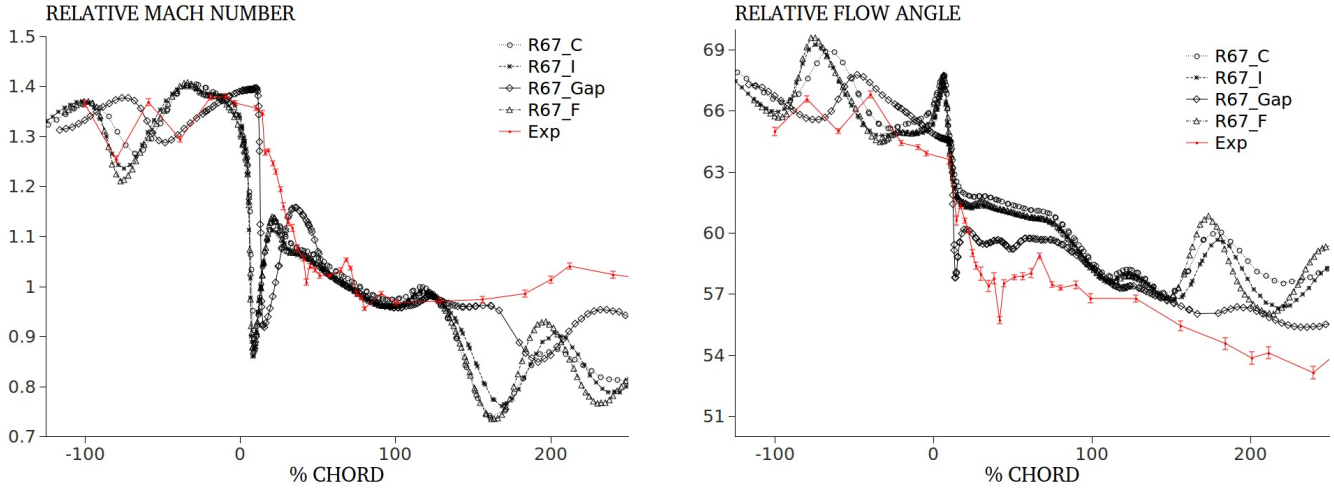
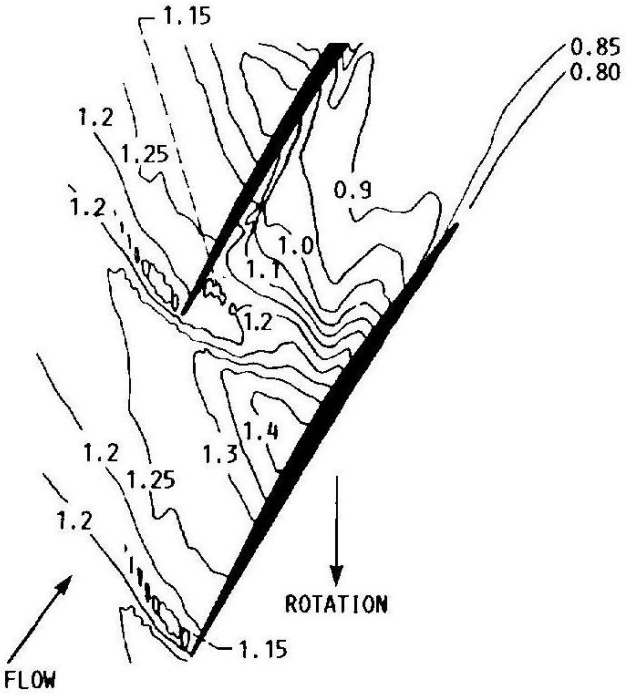
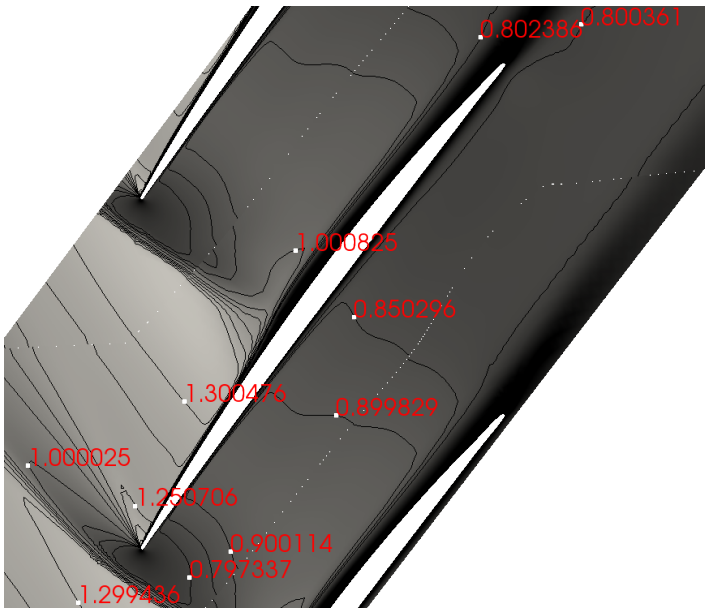


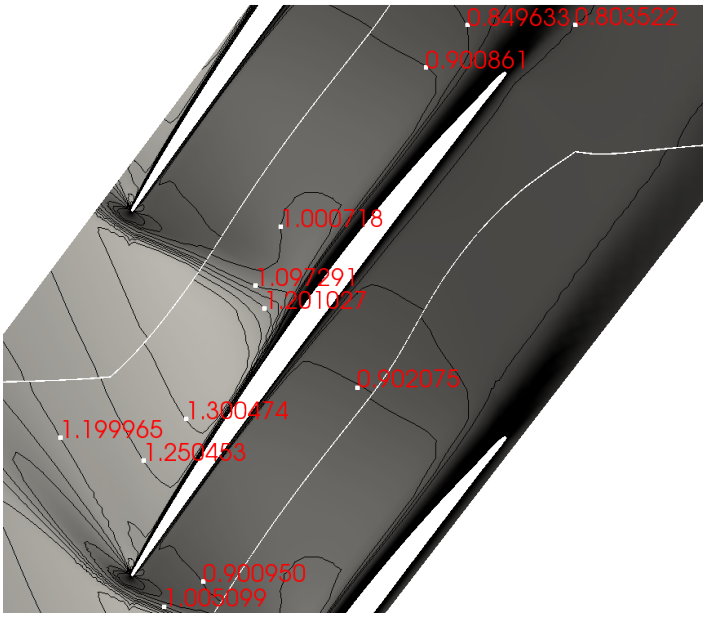

Figure 4.18: Plot of 80% pitch at 10% span




(a) Experimental




(b) Numerical (R67_F)




(c) Numerical (R67_GAP)

Figure 4.19: Mach number plot at 30% span from shroud near peak efficiency

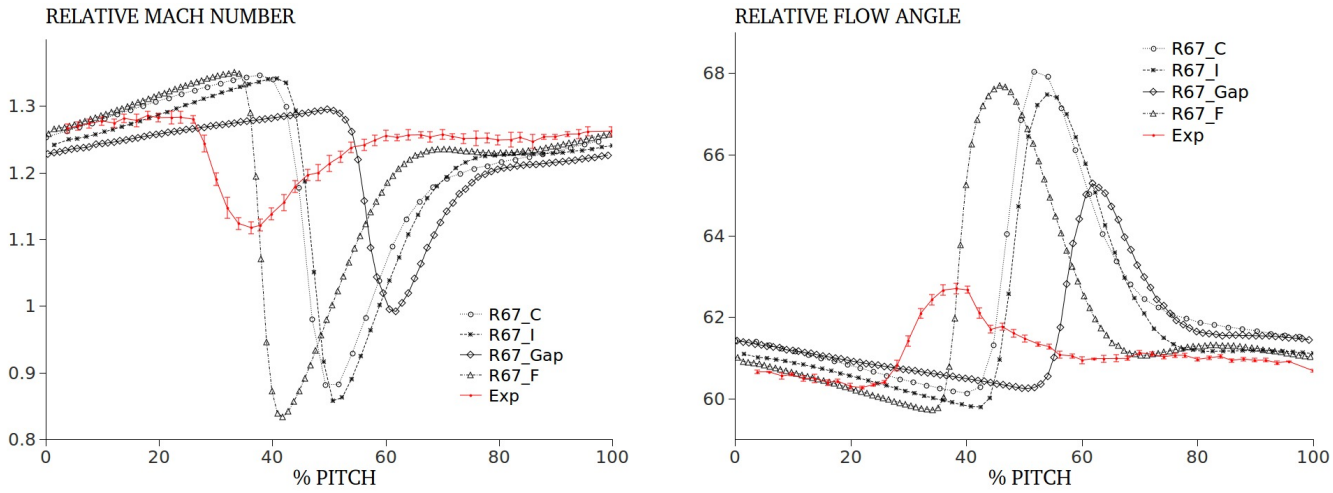



Figure 4.20: Plot of −10% chord at 30% span

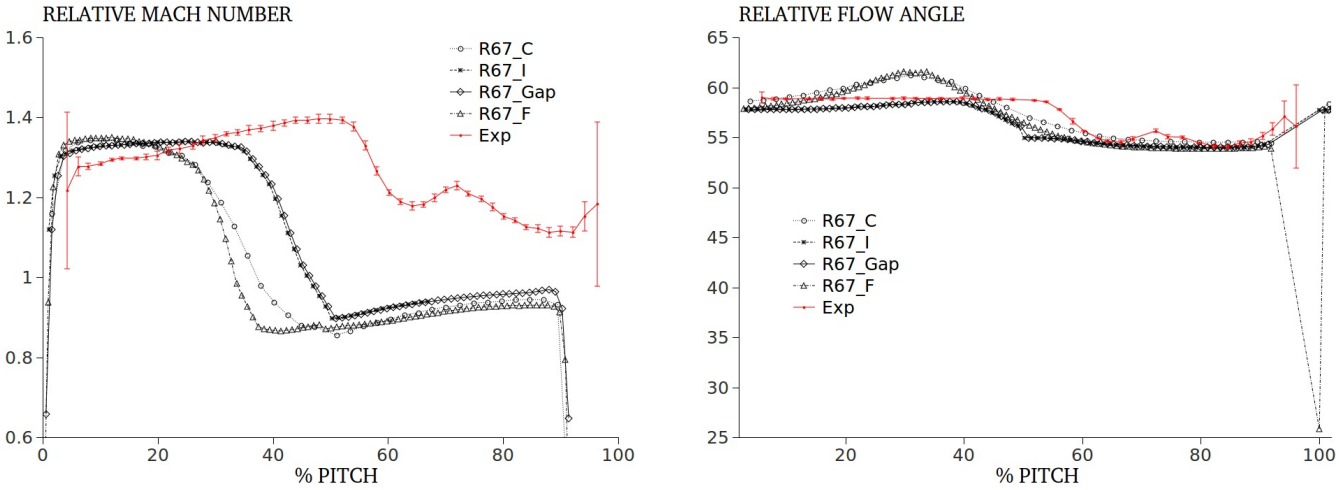



Figure 4.21: Plot of 23% chord at 30% span

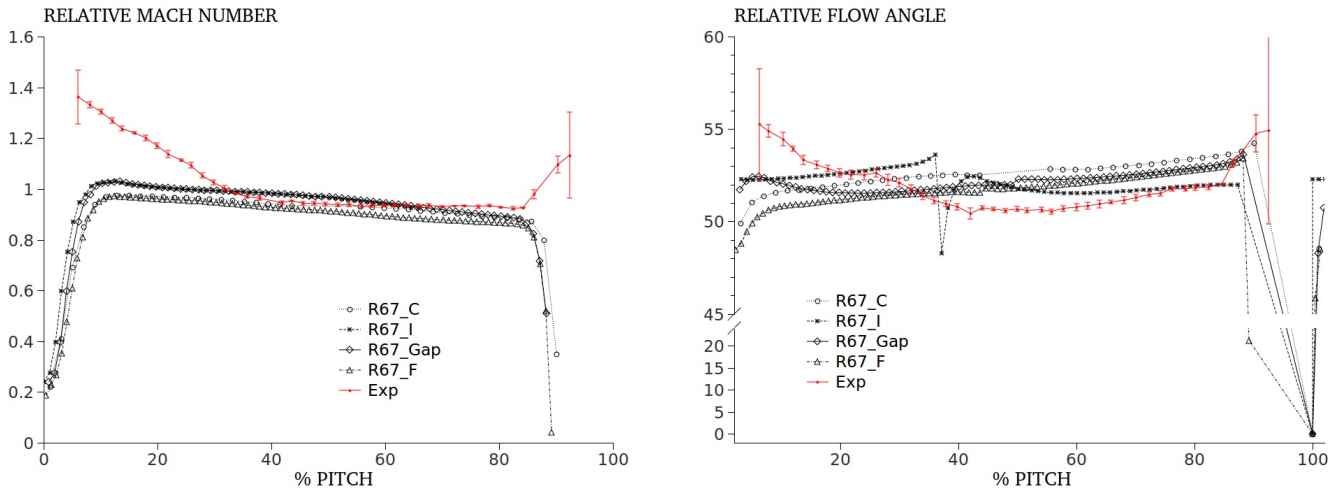



Figure 4.22: Plot of 50% chord at 30% span

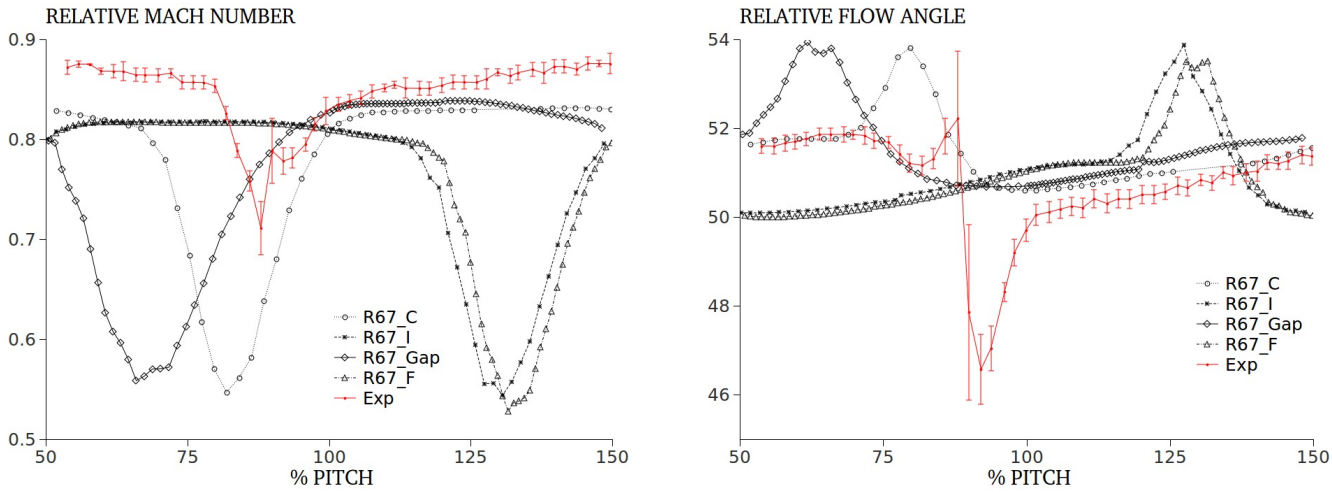



Figure 4.23: Plot of 124% chord at 30% span

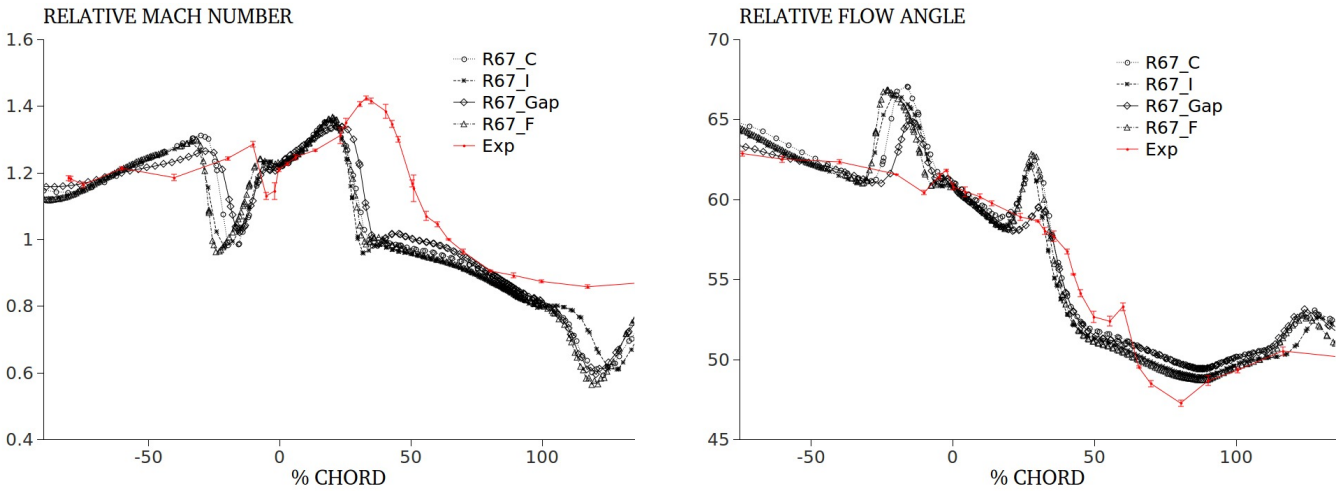



Figure 4.24: Plot of 20% pitch at 30% span

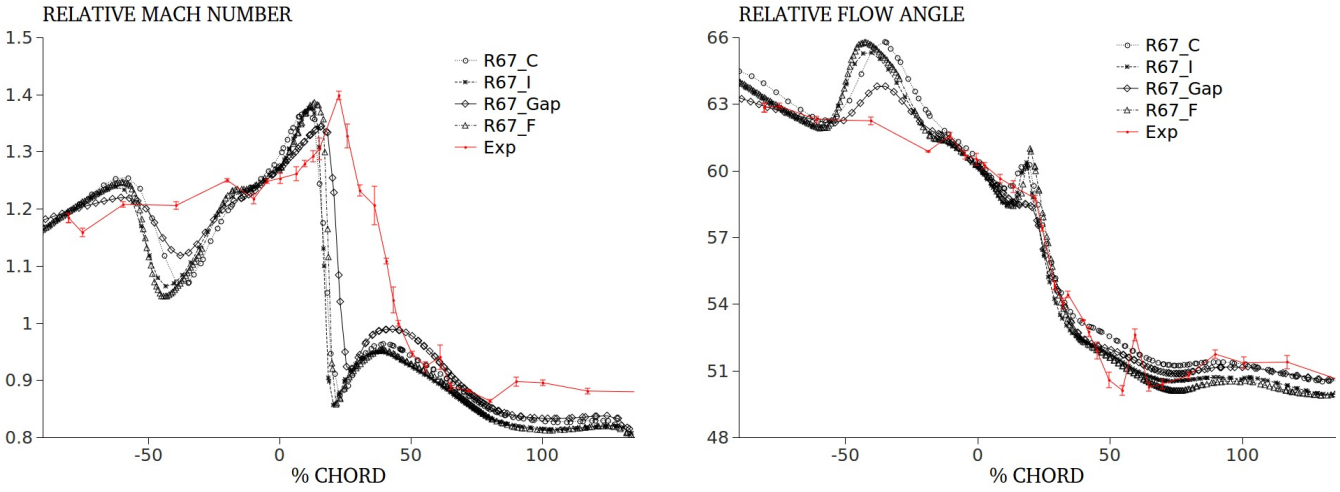



Figure 4.25: Plot of 50% pitch at 30% span

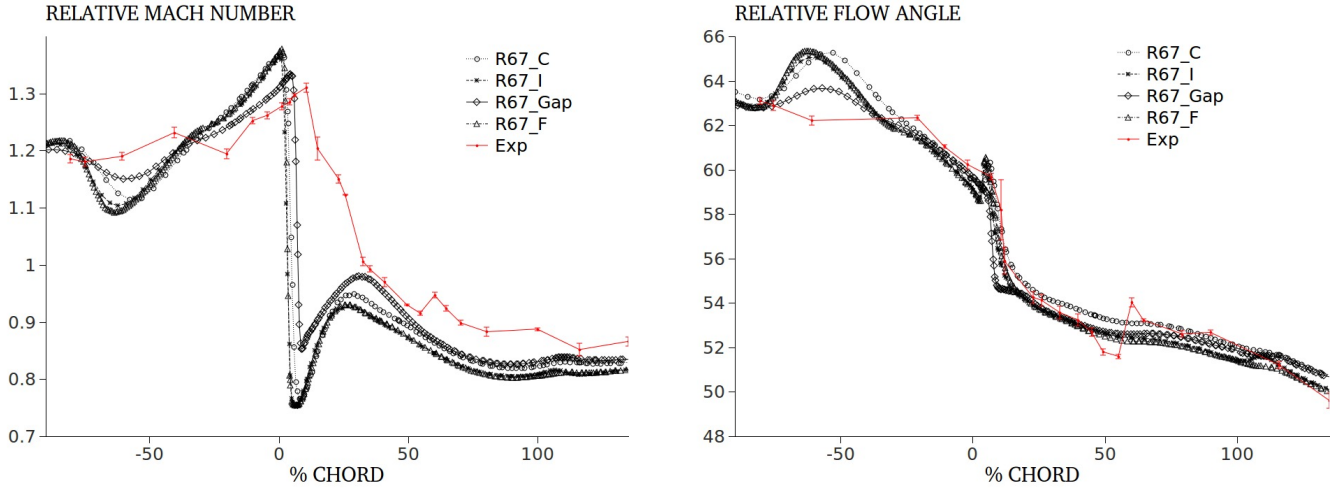
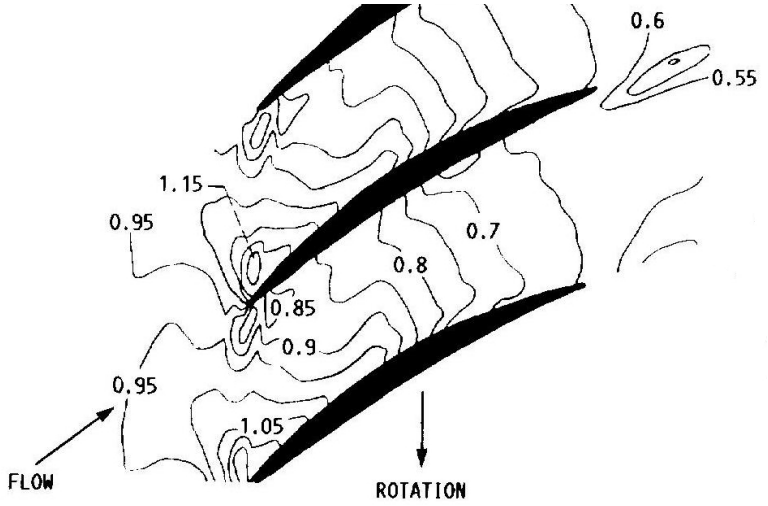
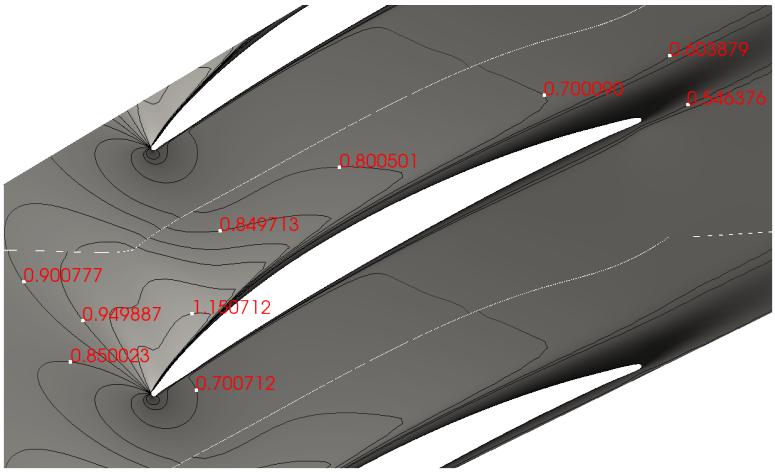


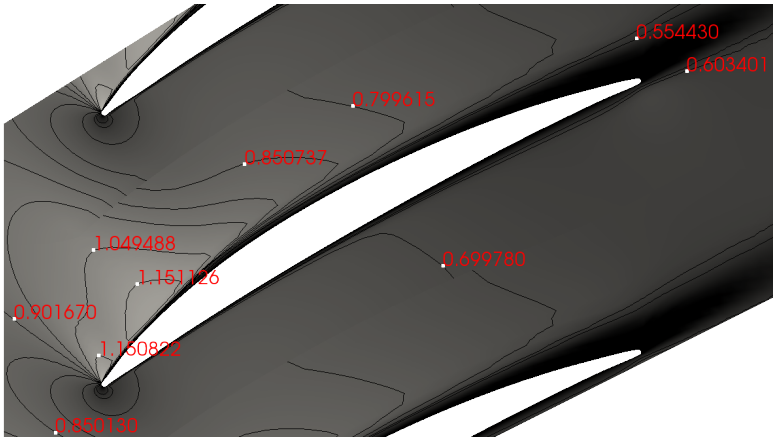

Figure 4.26: Plot of 80% pitch at 30% span




(a) Experimental




(b) Numerical (R67_F)




(c) Numerical (R67_GAP)

Figure 4.27: Mach number plot at 70% span from shroud near peak efficiency

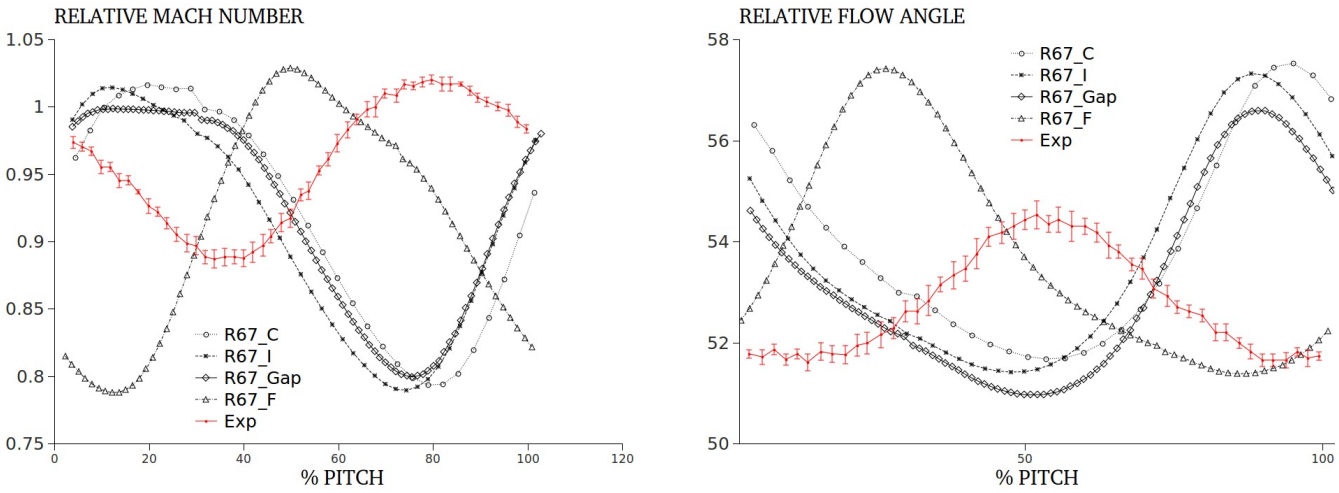



Figure 4.28: Plot of −10% chord at 70% span

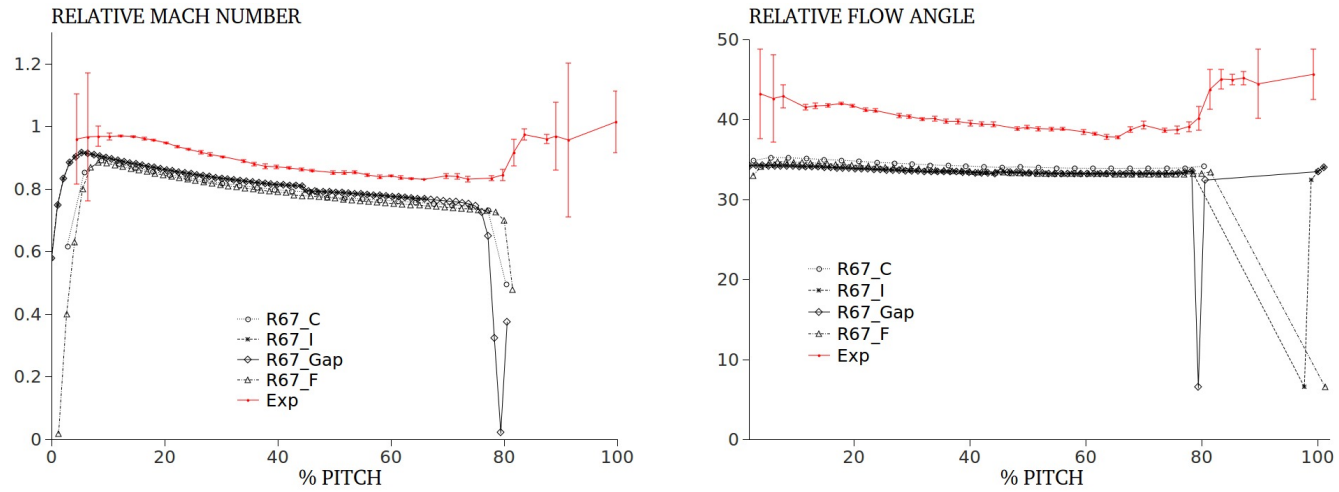



Figure 4.29: Plot of 30% chord at 70% span

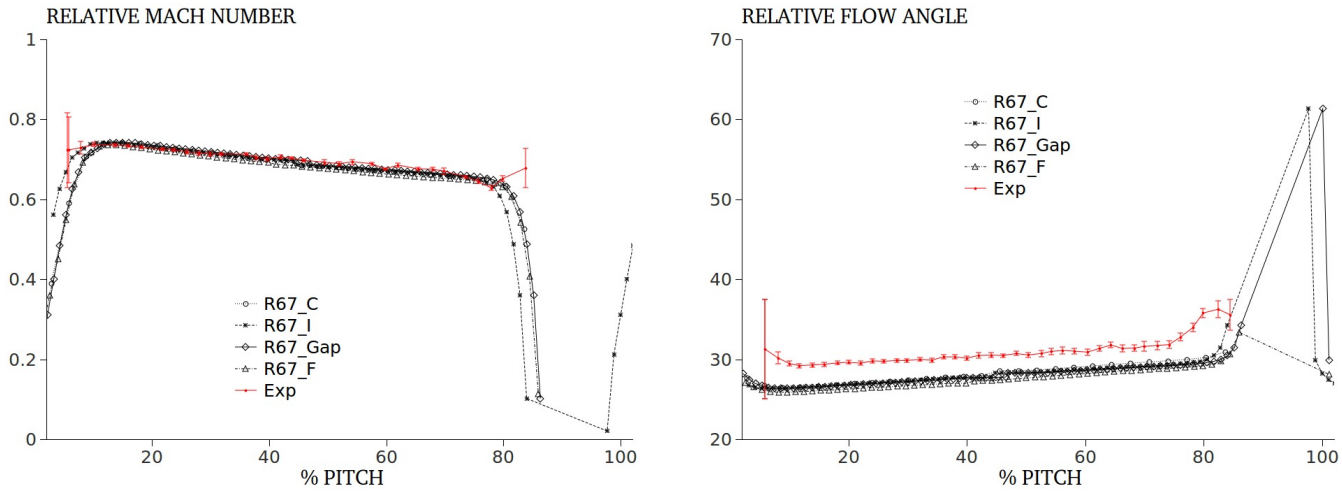



Figure 4.30: Plot of 70% chord at 70% span

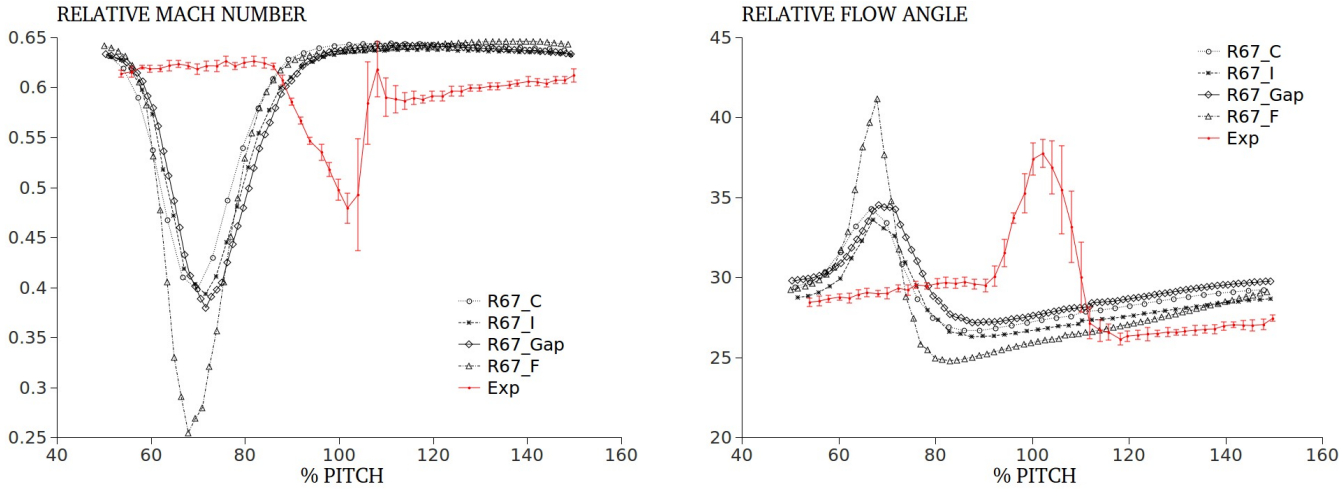



Figure 4.31: Plot of 121% chord at 70% span

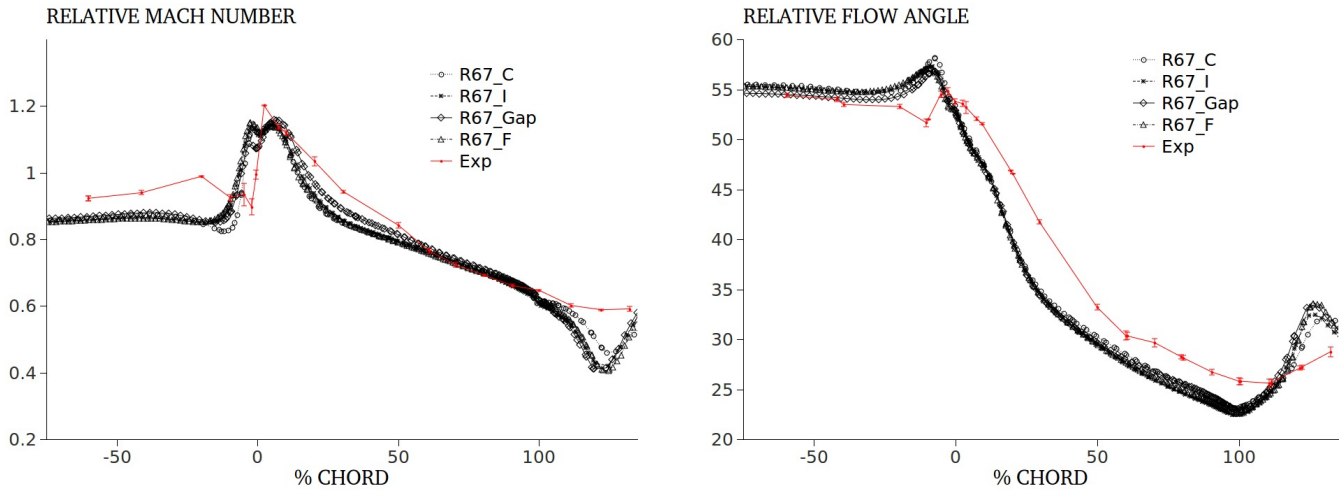



Figure 4.32: Plot of 20% pitch at 70% span

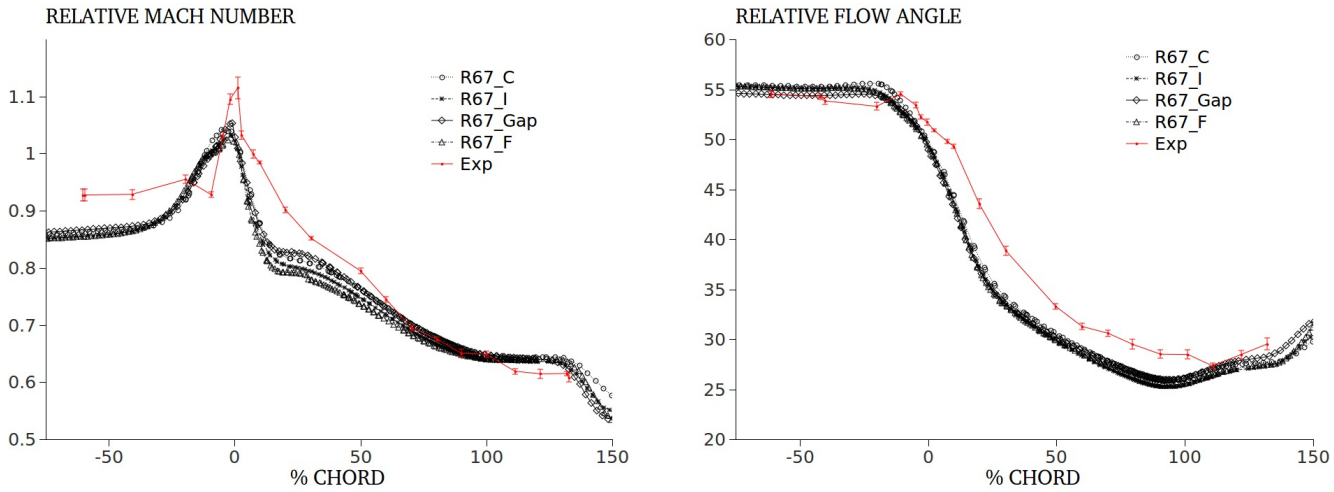



Figure 4.33: Plot of 50% pitch at 70% span

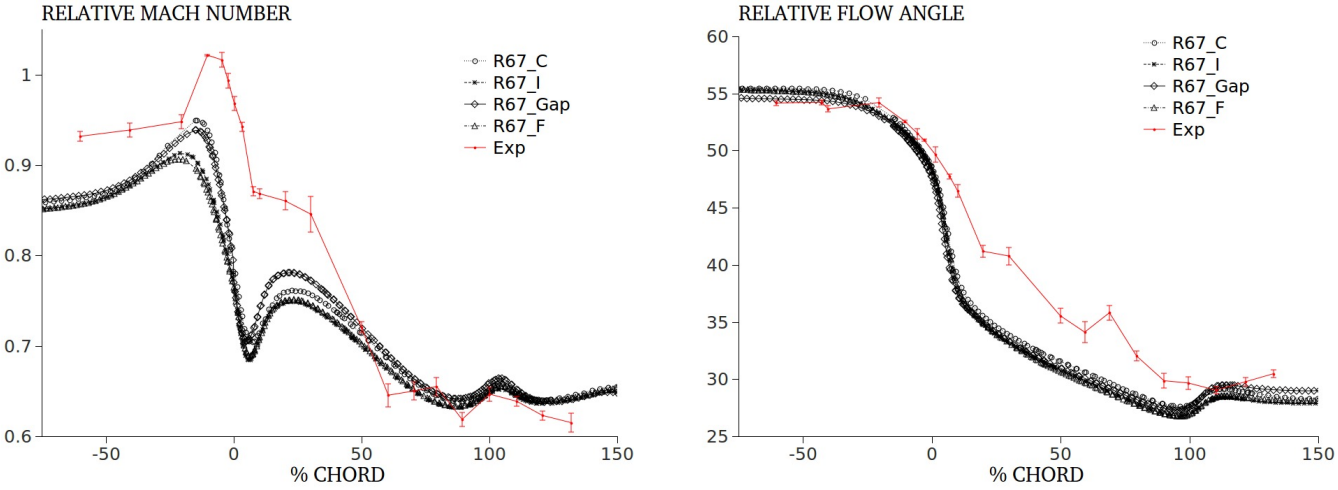



Figure 4.34: Plot of 80% pitch at 70% span

Some trends pop up when inspecting the rotor's Mach number structure: at 10% span, it is clear that the R67_GAP is in better agreement with experimental data than the R67_F. Although the number of cells between those meshes increases by a factor of approximately 2, the coarser is more accurate. At 30% and 70% span the differences don't seem to be significant and other tools to quantify these are needed: stream-wise and pitch-wise plots are studied with statistical tools in the Section 4.4.3.

As mentioned earlier, pitch-wise graphs are more likely to give bad results due to the linear stacking of blade's sections. Indeed, results appear to be associated with some translation vector which could be related to the distance between real and meshed LE and TE. It happens specially at "out-of-chord" positions because pressure distribution around LE and in the blade wake are most peculiar. Nevertheless, again with R67_GAP in Figure 4.15, it has a remarkable match despite what has been said, reinforcing the idea of the serious flow dependence on the tip-gap.

Finally, to strengthen that major errors in pitch-wise data are due to the linear stacking assumption, note the generality of stream-wise plots before 0% and after 100% chord and particularly Figures 4.18, 4.25: they are plotting a Mach distribution (wiggles) created by the LE and TE that doesn't exist in experimental data. If these results just occurred at the TE, it could be argued that the wake wasn't being resolved properly by turbulence models. Yet, flow arrives clean at the LE, where dissipative terms don't dominate. Thus, the geometry being different is a plausible justification for this local behavior.

### 4.4.2 Overall aerodynamic performance

Table 4.3 resumes the experimental and numerical overall aerodynamic performance while Figure 4.35 presents the experimental and numerical compressor maps[4]. Similarly to the previous section, only near peak efficiency results are presented.

[4]Mass flow is reduced to choked flow to better compare results.

| | **Near Peak Efficiency** | | | | |
|---|---|---|---|---|---|
| | **Experimental** | **Numerical** | | | |
| | | R67_C | R67_I | R67_GAP | R67_F |
| **Mass flow** ($\dot{m}/\dot{m}_{choked}$) | 0.988 | 0.951 | 0.953 | 0.973 | 0.946 |
| $\Pi$ (-) | 1.642 | 1.581 | 1.581 | 1.576 | 1.578 |
| **Adiabatic efficiency** (-) | 0.93 | 0.831 | 0.808 | 0.862 | 0.814 |

Table 4.3: Overall aerodynamic performance (numerical)

| | **Near Peak Efficiency** | | | |
|---|---|---|---|---|
| | **Numerical** | | | |
| | R67_C | R67_I | R67_GAP | R67_F |
| **Mass flow** (%) | 3.75 | 3.54 | 1.52 | 4.25 |
| $\Pi$ (%) | 3.71 | 3.71 | 4.02 | 3.90 |
| **Adiabatic efficiency** (%) | 10.65 | 13.11 | 7.31 | 12.47 |

Table 4.4: Relative error

From Table 4.4 it is clear the superior match of R67_GAP to experimental data. From the other meshes, it is not easy to deduce any conclusion or trend. This follows essentially because this parameters, as the name implies, are an overall measure and by default gather many variables.

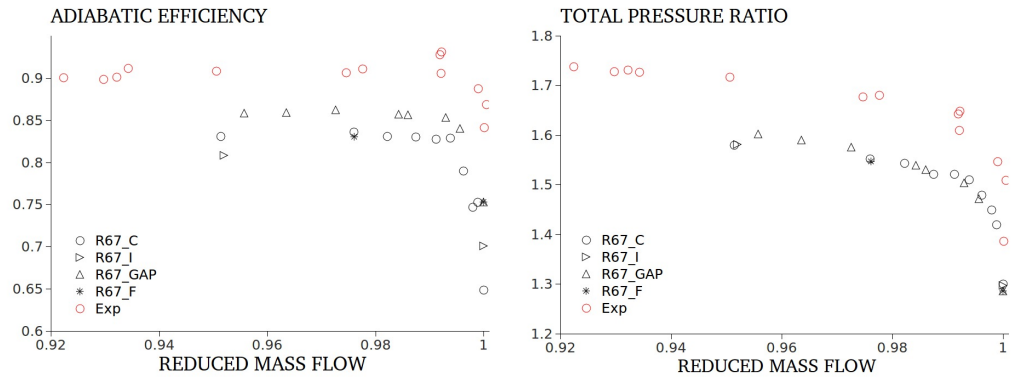
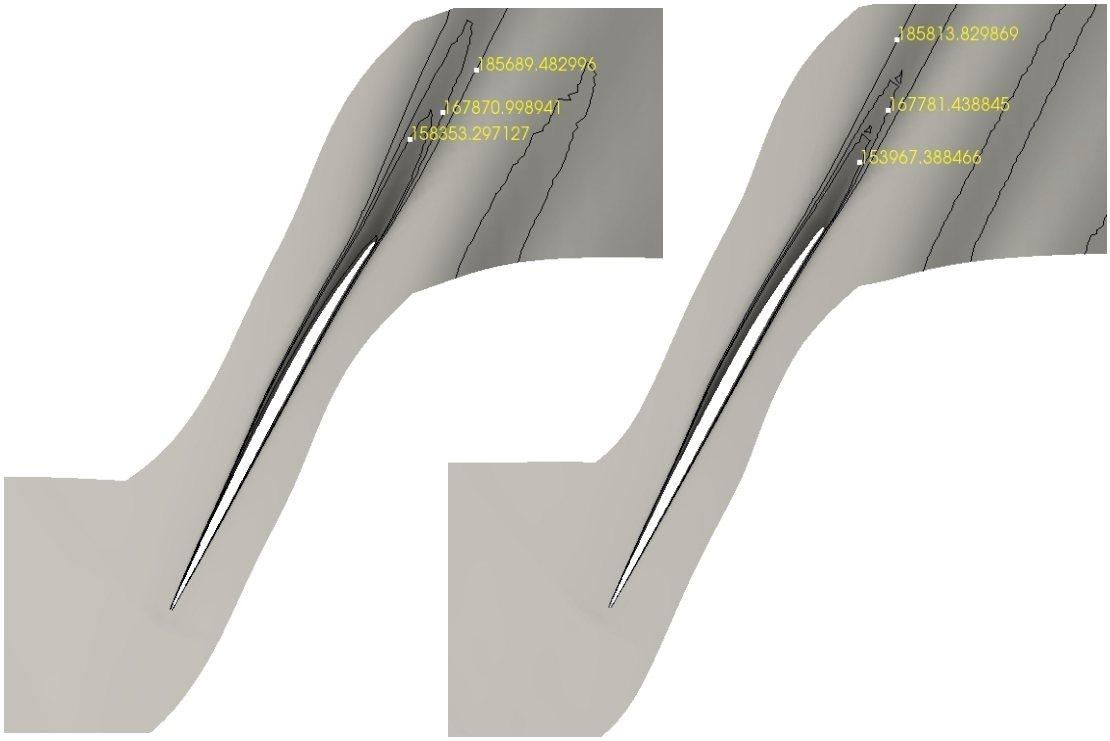



Figure 4.35: Compressor map

To justify why the R67_GAP has greater efficiency, Figure 4.36 plots the relative total pressure at 10% span for all the different meshes. Relative total pressure gives a measure of where and how much are the losses: in an ideal compressor, the relative total pressure is constant across the rotor. Thus, one can see less losses in R67_GAP which supports the greater overall aerodynamic performance.

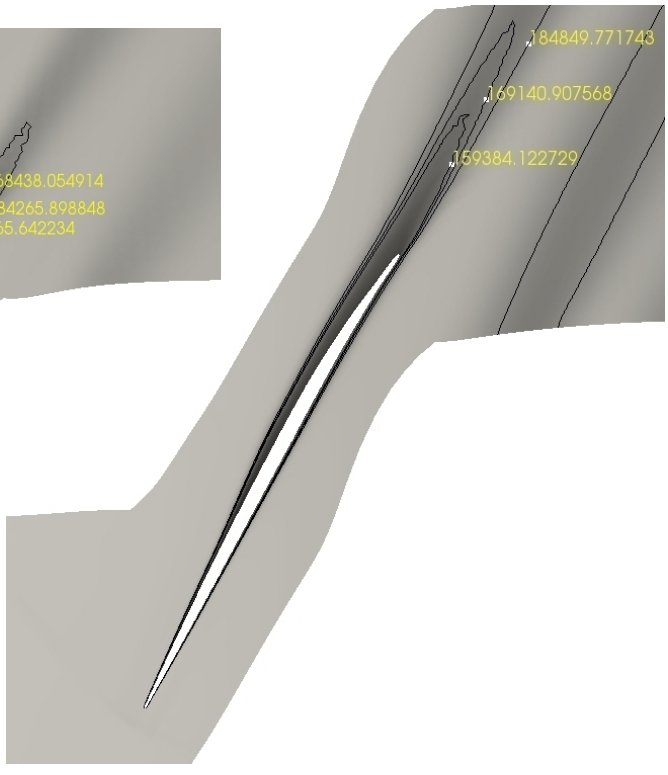



(a) R67_C




(b) R67_I


(c) R67_GAP




(d) R67_F

Figure 4.36: Relative total pressure at 10% span

### 4.4.3 Statistical analysis

This section presents some useful statistics that give a quantitative measure about how close numerical results relate to experimental data. Statistical tools are *BIAS*, *root mean square error* (RMSE) and *concordance index* (CI).

BIAS gives the average of difference between numerical and experimental data. It is useful to see if numerical data is over or under-predicting experimental results. The formulae reads

$$BIAS = \frac{\sum_{i=1}^{n}(y_i - x_i)}{n}. \tag{4.1}$$

The RMSE is similar to BIAS but measures instead the distance between numerical and experimental data. It is often used as an accuracy accessing tool. The formulae is given by Eq. 4.2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \tag{4.2}$$

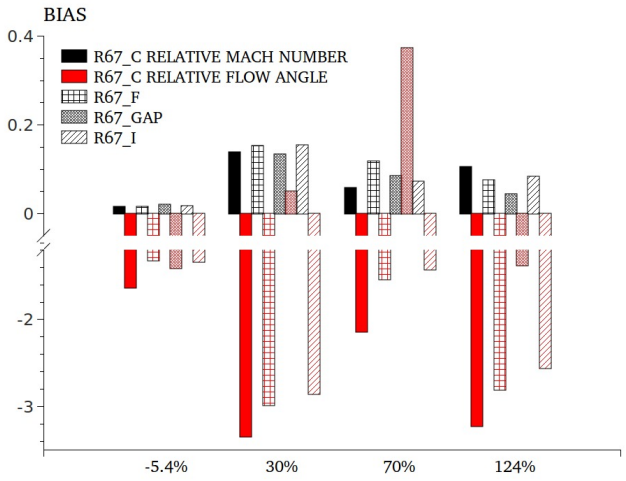
By last, CI shows how well numerical and experimental data match. Although it is a quantitative tool, its output can be interpreted as a qualitative information. If CI is 1 we are in presence of a perfect match between data while if CI is 0 the data clearly isn't related. Again, the formulae is given by Eq. 4.3.

$$CI = 1 - \frac{\sum_{i=1}^{n}|y_i - x_i|^2}{\sum_{i=1}^{n}(|y_i - \overline{x}| - |x_i - \overline{x}|)^2} \tag{4.3}$$

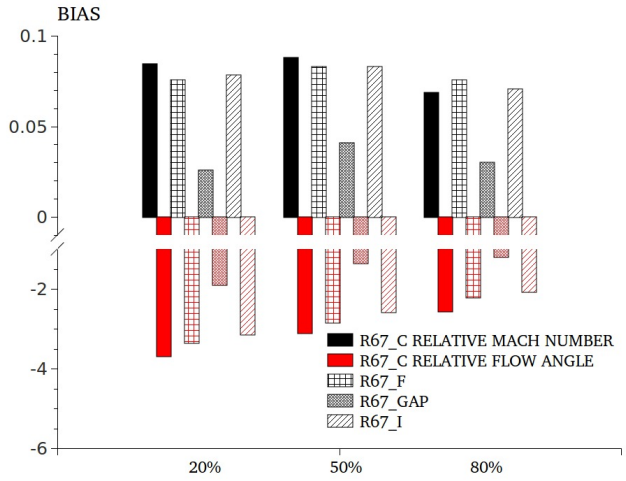
It is important to mention how the following plots were generated. To be able to apply such statistical tools, it is necessary to have pairs of data. A pair of data is called when numerical and experimental data have exactly the same X coordinate. Without those, comparing data becomes naturally useless.

Problems arise because the exact position of experimental data extraction will often be different from numerical, creating difficulties to get pairs. The solution was to interpolate all data, discretize it in 100 points and select a common range both to numerical and experimental data. This interpolation as well as all graphs were done in the open-source software *QTIPlot*.

Graphs are separated in pitch-wise and stream-wise data. Each graph plots only a statistical tool and compares all the different meshes. By last, red colored columns refer to statistics about relative flow angle and black colored columns refer to relative Mach number.

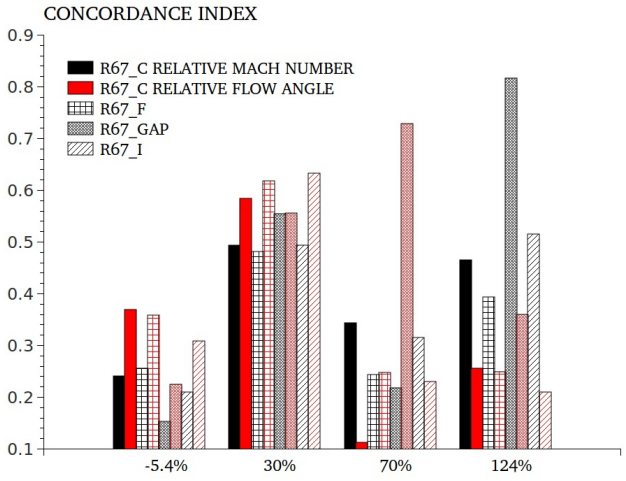



(a) Pitch-wise

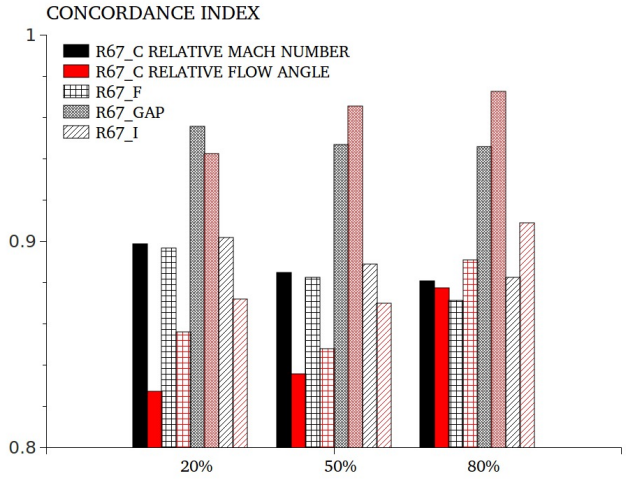



(b) Stream-wise

Figure 4.37: 10% Span BIAS

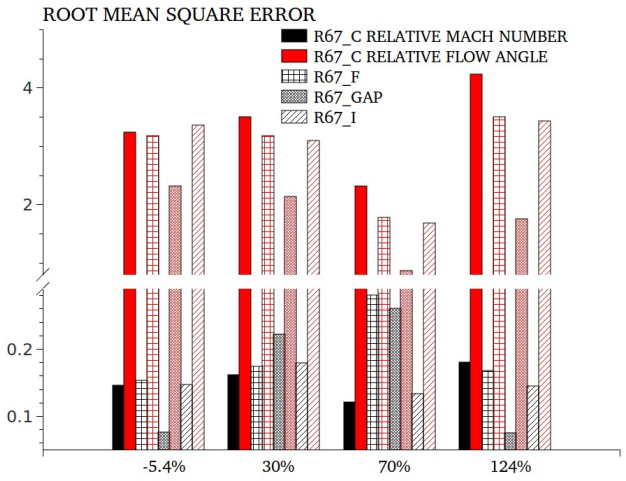



(a) Pitch-wise

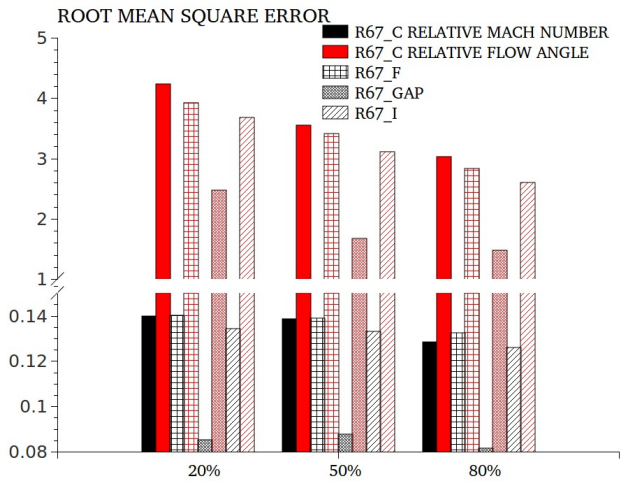



(b) Stream-wise

Figure 4.38: 10% Span CI

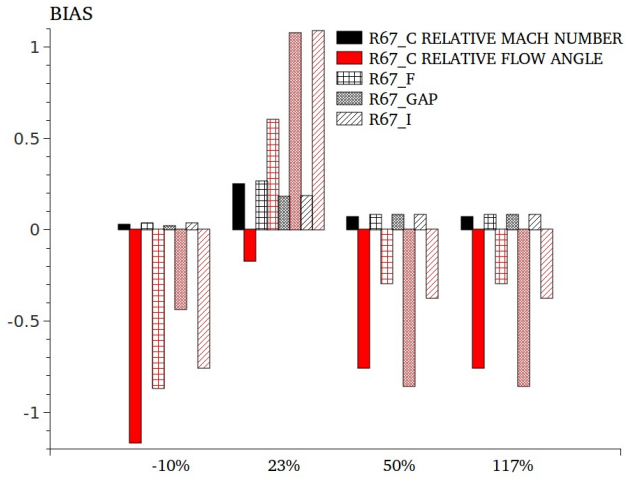



(a) Pitch-wise

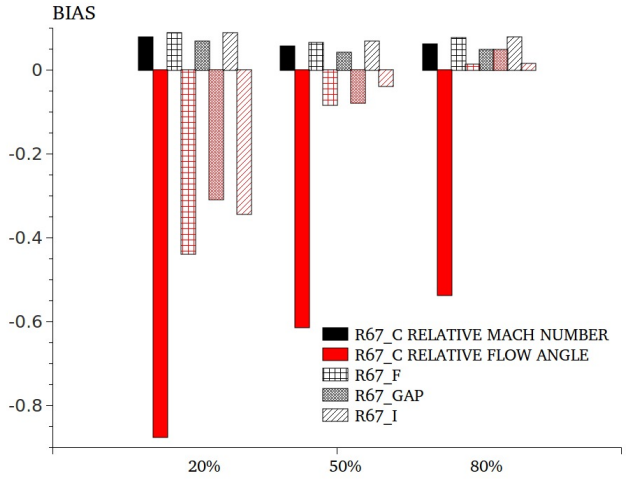



(b) Stream-wise

Figure 4.39: 10% Span RMSE

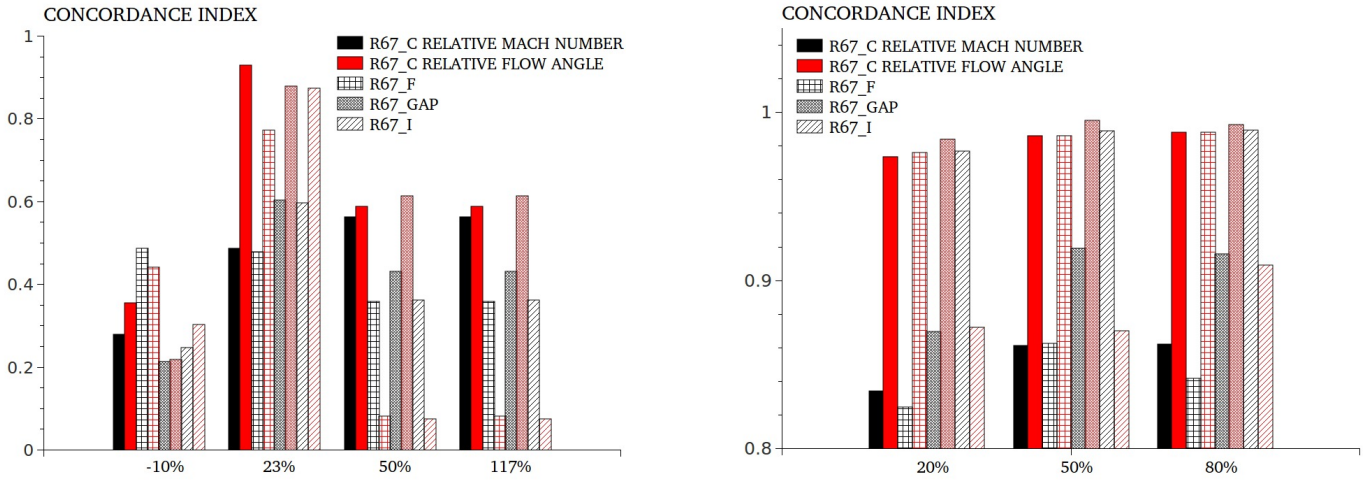



(a) Pitch-wise




(b) Stream-wise

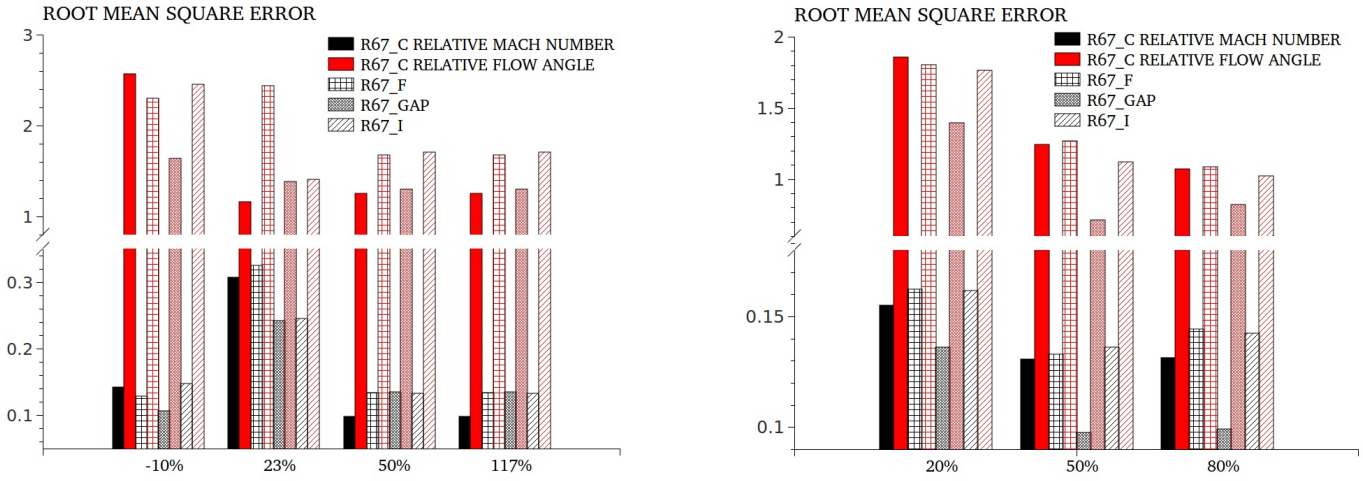
Figure 4.40: 30% Span BIAS




(a) Pitch-wise

(b) Stream-wise

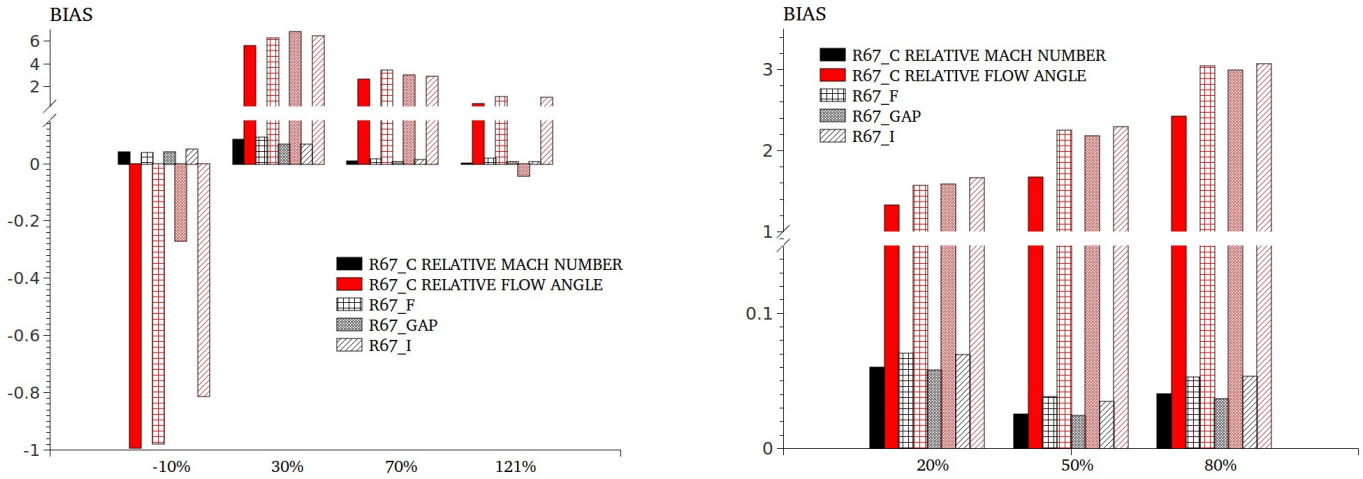
Figure 4.41: 30% Span CI




(a) Pitch-wise

(b) Stream-wise

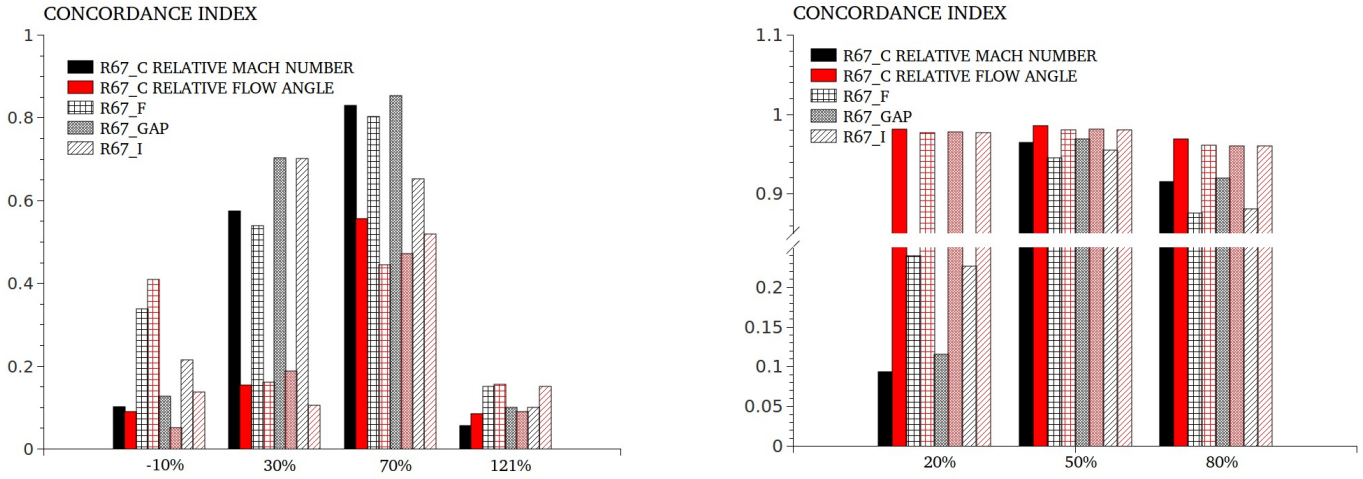
Figure 4.42: 30% Span RMSE




(a) Pitch-wise

(b) Stream-wise

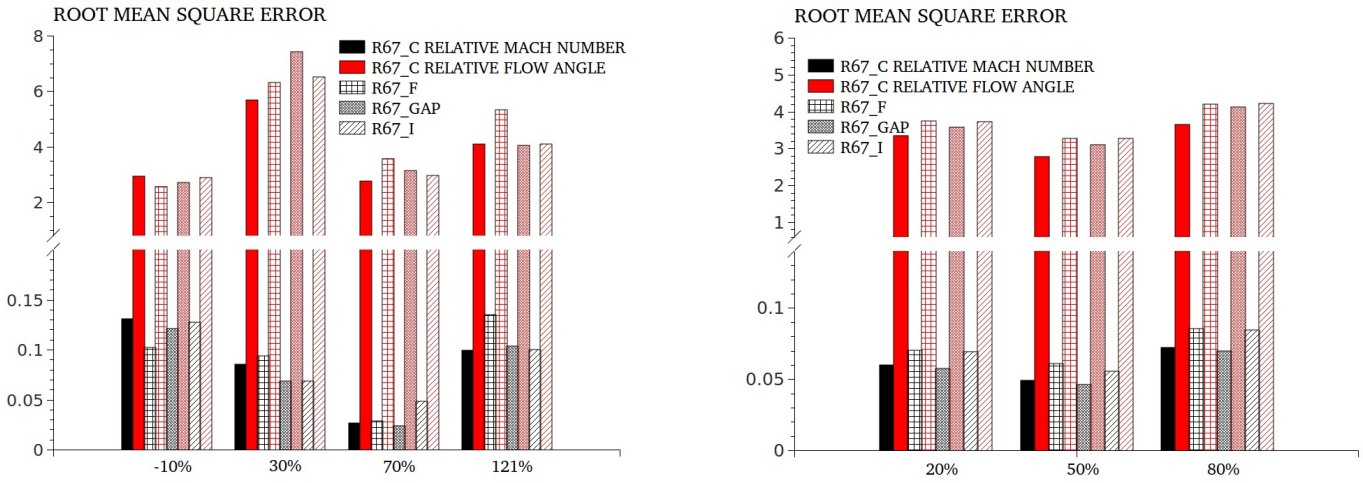
Figure 4.43: 70% Span BIAS




(a) Pitch-wise

(b) Stream-wise

Figure 4.44: 70% Span CI




(a) Pitch-wise

(b) Stream-wise

Figure 4.45: 70% Span RMSE

From the above plots, one can make several observations. We start by saying that indeed stream-wise have bigger CI then pitch-wise plots in all %span. This fact, as mentioned earlier, can be justified in part by the need of stacking linearly the blade sections which differs from the real geometry. LE and TE coordinates will be different and consequently at each span percentage, the axial chord length will be different. Again regarding CI, it is the R67_GAP that has the highest values thus becoming the best match to experimental data.

Paying attention to the RMSE, one can observe that in every span percentage, independently of pitch-wise or stream-wise plots, the solutions of the intermediate and fine meshes are always very close. This suggests some evidence of grid independence that should be further investigated. Continuing with the same statistical tool, the R67_GAP stays around the R67_F and gets better results at 10% span. This makes perfect sense since it is the closest survey plan to the tip gap.

By last, when inspecting the BIAS, similar conclusions to the RMSE can be drawn because these tools are very related. Nevertheless, additional information can be added: at 10% and 30% span, the relative Mach number is generally overestimated and relative flow angle is underestimated. At 70% span both relatives Mach number and flow angle are overestimated.

### 4.4.4 Vortex shedding

Figure 4.46 illustrates the particular tip gap case where vortex shedding can be seen. It should be noted that the result is not much accurate as vortex shedding is a transient phenomena with determined frequencies. Therefore, the steady state form of this phenomena displays the average energy contained in the vortex across space.

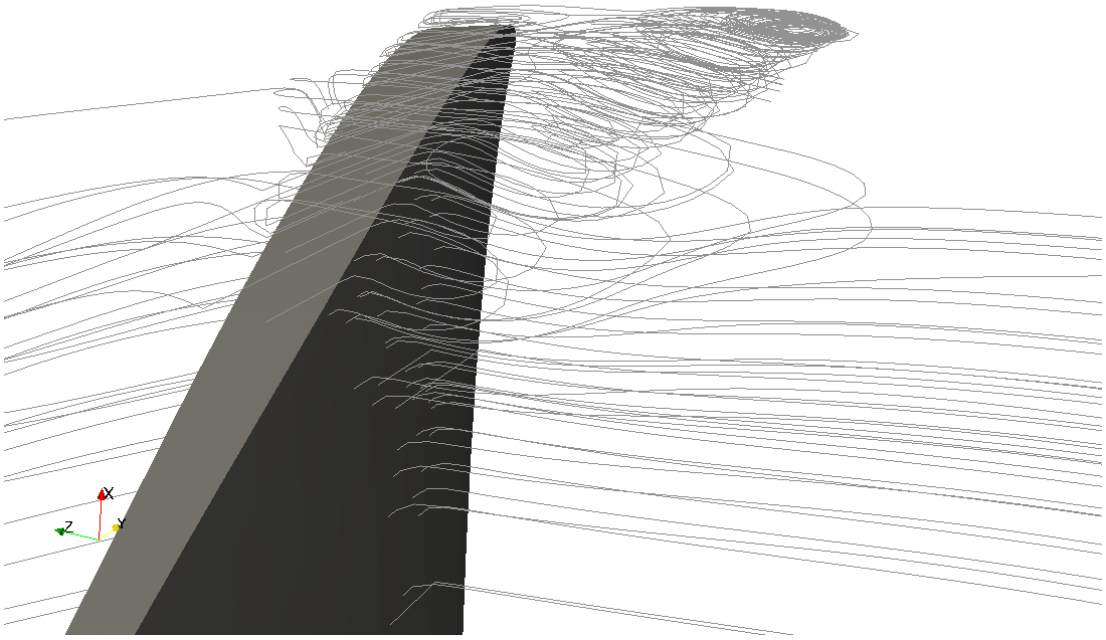



Figure 4.46: Tip gap vortex shedding

As mentioned earlier, this geometry also creates a vortex in the TE. Although this work is not meant to validate unsteady phenomena, or make any comparison whatsoever, Figure 4.47 shows, once again, that the HLLC can catch vortex as suggested in Chapter 3.

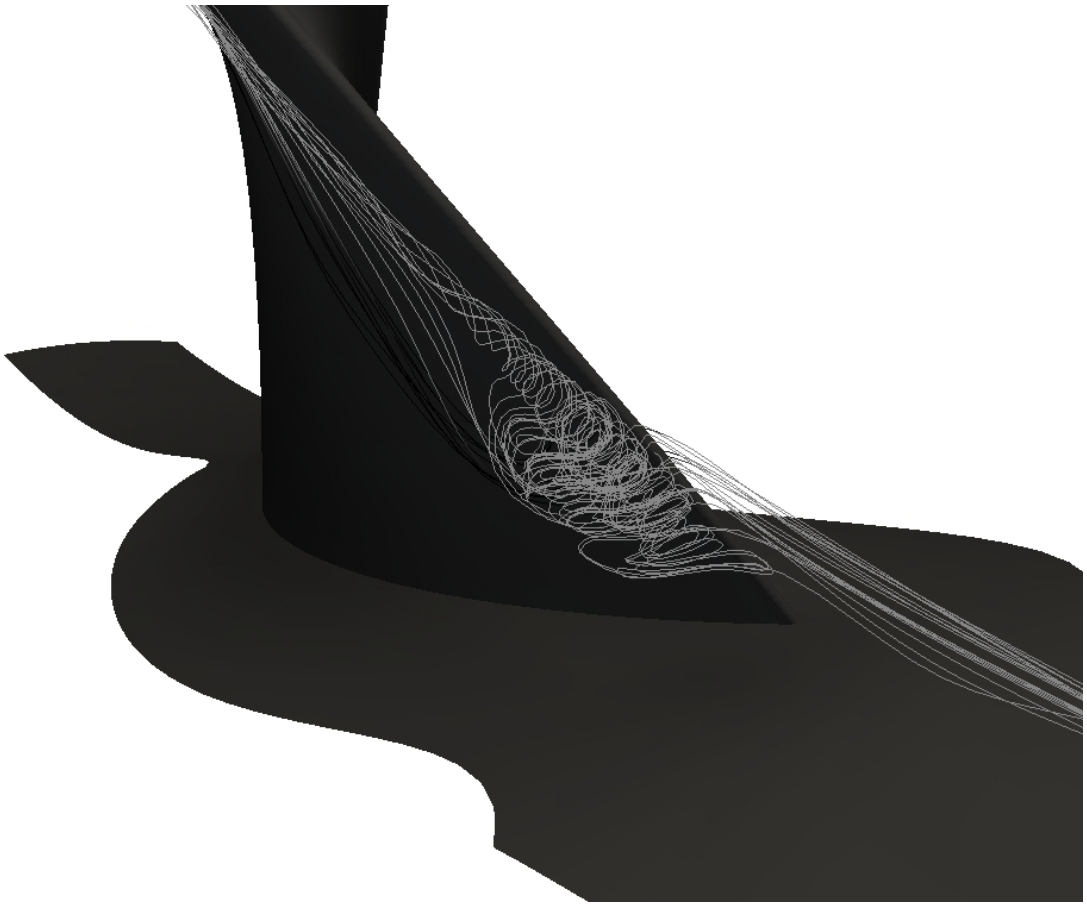


Figure 4.47: Trailing edge vortex shedding

### 4.4.5 Convergence Time

Convergence was introduced in Section 2.8 and it is achieved when $L^{\infty}$ and $L^{2}$ metrics drops by 3 orders of magnitude. Plus, the difference between mass flow at inlet and outlet of the rotor must be under 0.1%. Total pressure ratio must also be constant for a solution to be said convergent. Table 4.5 presents the simulations computational cost[5]

[5]Time based in a parallel computation with two cores of $3.10Ghz$ each; $8Gb$ RAM

| Case | Average time per iteration (seconds) |
|---|---|
| R67_C | 2.32 |
| R67_I | 3.45 |
| R67_GAP | 3.91 |
| R67_F | 6.59 |

Table 4.5: Convergence time

and to reach convergence, between 15000 and 25000 iterations need to be done with $C_{cfl} = 0.7$. The number of iterations depend with some factors, namely mesh quality, flow structure generated by the imposed boundary condition and initial conditions[6].

## 4.5 Remarks

R67 geometry was tested under Borm's implementation of a density-based solver in OpenFOAM. Steady state simulations were carried out with three different mesh levels, having the finest about $8 \times 10^5$ cells for a single passage.

Experimental data was compared near peak efficiency. The analysis was done at this point because it is where steady state assumptions are most correct, i.e. less transient phenomena and convergence is reached easier.

Thanks to LA technology, it was possible to identify the flow structure across the rotor, thus conclusions can be drawn when comparing experimental with numerical results. Indeed, absolute values are not similar but that is not alarming: giving the correct trend becomes equally important in CFD. The concordance index being the highest in the R67_GAP is a good result thus confirming what is well known: the tip-gap geometry is a very important parameter on turbomachinery [3].

The linear stacking assumption seems to be a plausible justification for data that match poorly, specially in positions "out-of-chord": lack of precision in LE and TE co-

[6] More will be said about this topic in Chap. 6.

ordinates causes different Mach number distributions and consequently every variables vary accordingly.

Regarding flow structures across the rotor, it is clear that the hyperbolic system is being well resolved and it doesn't need many cells to reach good accuracy: statistical tools suggested that grid independence from $4 - 8 \times 10^5$ is being reached and R67_C overall aerodynamic data doesn't differ much from both R67_I and R67_F.

Finally, every major flow structure known to exist a-priori was captured by the HLLC: tip gap vortex, trailing edge vortex and shock. This shows how robust the numeric methods implemented are and what can be expected when using this solver.

# CHAPTER 5

# NASA COMPRESSOR STAGE

This chapter is meant to test multi-row problems, namely with the use of the *Mixing-Plane* (MP) approach. A complete validation will not be attempted as a very coarse mesh will be used to show how this type of simulations is carried out in OpenFOAM. Nevertheless, overall results will be compared with experimental data to find out what can be expected of the MP.

The original geometry, tested in the same facility as the R67, is a two-stage fan designed to achieve 2.40 pressure ratio and a tip speed of 427 $\mathrm{m\,s^{-1}}$. It is designed to develop efficient, lightweight engines for short-haul aircraft and axial spacing is given between blade rows to reduce noise.

In this chapter, only one stage is meshed, as the computational procedure is the same for more stages and the computational resources are limited. Thus, the rotor has 43 blades while the stator has 34: the geometry is reported on [62] and the CAD model is presented in Figure 5.1.

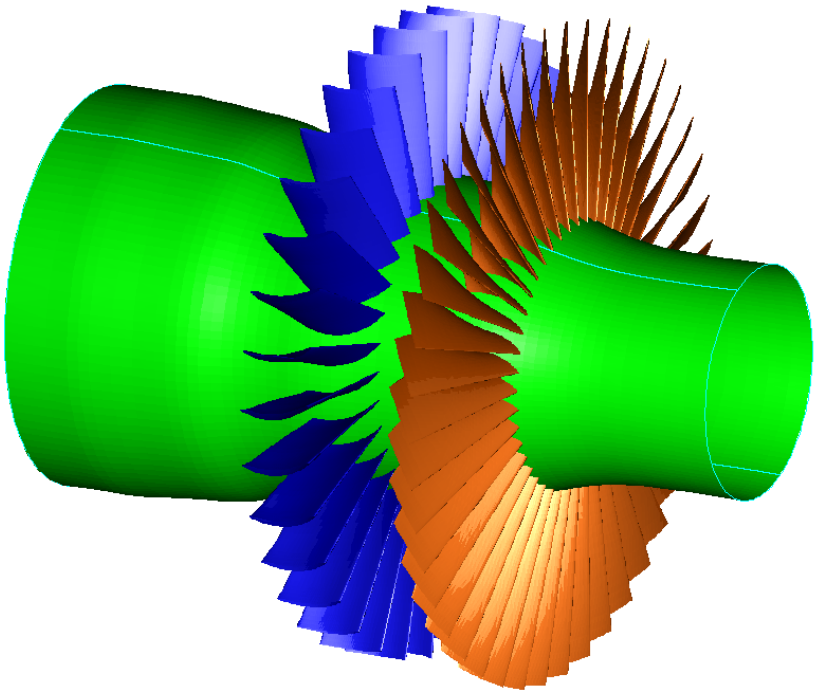


Figure 5.1: Complete stage: orange - rotor blades; blue - stator blades; green - hub

## 5.1 Data

In the experimental facility, radial measurements of flow quantities were made, namely $T_0$ and $P_0$. Probes were installed at the inlet and outlet of each row and that is why, for the sake of comparing data, there is no complication in just simulating one stage. This setting allowed the survey of eleven radial positions which provide the overall aerodynamic performance across each row. For more detailed information about the apparatus consult [62].

Concerning numerical data and how it is extracted to compare results, the procedures are equal to R67, presented in Chapter 4.

## 5.2 Mixing plane

The MP [63] is a common technique to simulate steady multi-row problems. It does an circumferential averaging of variables at the interface (Figure 5.2) which can be seen as

an inlet boundary condition to the second row. This method doesn't give a physically correct representation of the rotor-stator interaction as this is an unsteady phenomena and the main structures of the flow will be lost in the average. Nevertheless, it is one of the two feasible[1] ways of getting steady solutions of multi-row problems and it provides a reasonable representation of the problem. The other approach is the *frozen rotor*, where the difference to the MP is that is doesn't perform the circumferential averaging: the solution will depend on the relative position between rotor and stator.

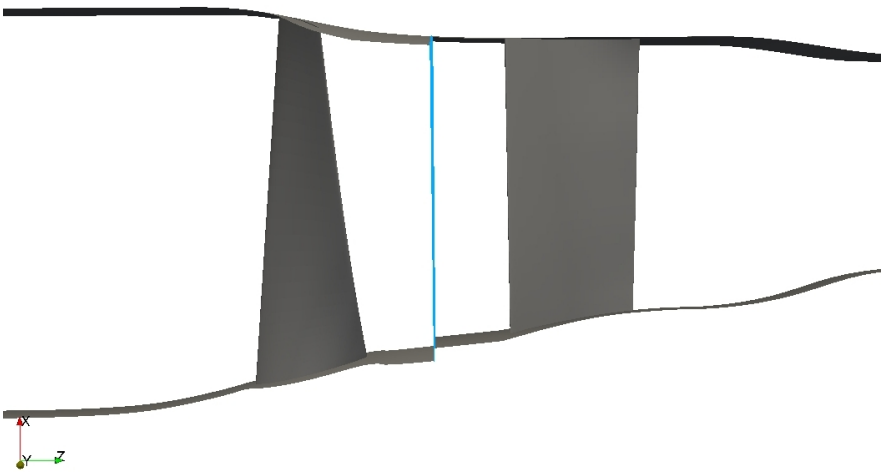



Figure 5.2: Stage layout: blue - mixing plane

## 5.3 Mesh

The procedure to input geometry from the tabular to both design and mesh softwares is the same as used in R67. One must have in mind when a geometry is described, it is usual the suction surfaces of every row to be faced to the same side, which naturally is a ill defined problem[2].

The most important issue when building the multi-row mesh is the interface between rows. In OpenFOAM, the MP has its own well defined interface plane[3], which the

[1] Full annulus simulations don't require a MP but are far too expensive

[2] Rotor's suction surface is usually in the opposite direction from the stator's (Figure 1.4)

[3] Cartesian, cylindrical

mesh must obey otherwise the simulation will blow up. These plane, called *ribbon* in OpenFOAM, has other specifications, namely the direction in which it will average. Finally, the MP interface must be in the same frame of reference, i.e. every cell attached to the ribbon must be in the rotating frame.

Figure 5.3 presents the interface boundaries of the rotor's outlet (also stator's inlet) and the following specifications[4] must be written in the *boundary* file:

```
outlet_0    {
    type            mixingPlane;
    nFaces          1936;
    startFace       1973984;
    shadowPatch     inlet_1;
    coordinateSystem
    {
        name            mixingCS;
        type            cylindrical;
        origin          (0 0 0);
        e1              (1 0 0);
        e3              (0 0 1);
    }
    ribbonPatch   {
        sweepAxis       Theta;
        stackAxis       R;
        discretization  bothPatches;
    } }
```

[4]Similar has to be defined for the inlet of the stator

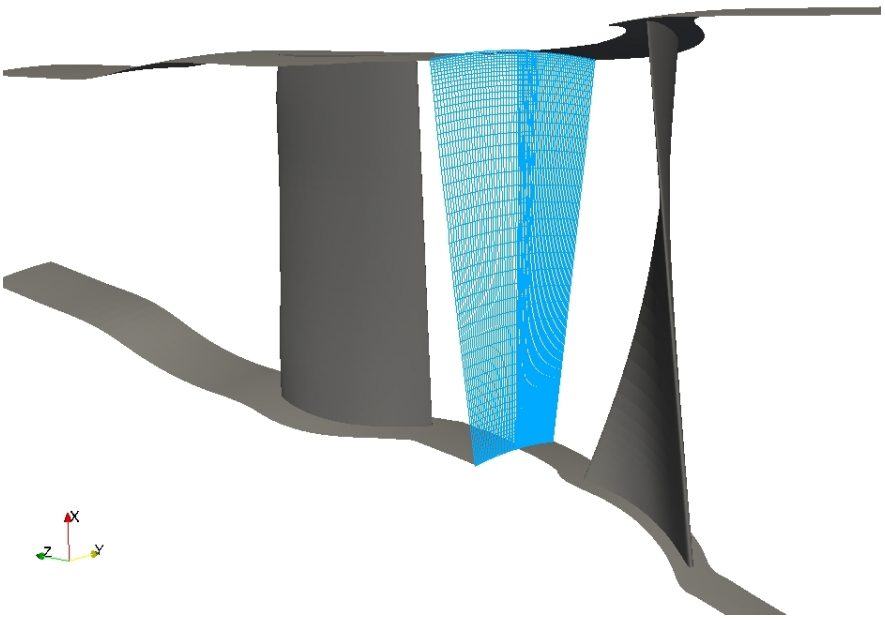



Figure 5.3: Stage layout: blue - mixing plane grid

*outlet_0* is the name of the boundary condition and it refers to the outlet of the rotor. *type* specifies the kind of boundary condition whether *nFaces* and *startFaces* refer to which faces the boundary are assigned to. *shadowPatch* is the name of the boundary which will make a pair for the MP. In *coordinate system*, type is chosen cylindrical where *e1* and *e3* define the directions of the cylindrical referential. As was mentioned, the ribbon patch will specify the direction in which it will perform the average (*sweepAxis*) and in which direction it will stack the cells (*stackAxis*).

Finally, the number of cells of this mesh is presented in Table 5.1. For simplicity and time reasons, wall functions are used as the mesh $Y^+$ is bigger than 11 and no tip-gap is reproduced.

| Row | Number of cells |
|---|---|
| Rotor | 451264 |
| Stator | 204672 |

Table 5.1: Rotor-stator mesh number

## 5.4 Numerical setup

The numerical setup is the same defined in R67, presented in Chapter 4. Naturally, there will exist extra boundary conditions as there exists an extra row but, apart from the MP boundary described above, there is nothing to add. Thus, the solver, the turbulence model, inlet boundaries, and mesh connection are all the same used in R67. For the outlet boundary condition, a back-pressure is set to 95000 Pa.

When running a simulation with the mixing plane, performing parallel computations is only allowed with first spatial order. The reason for this might be the limiters that prevent the wiggles that enter in conflict with the mixing plane averaging. Because there was short time to run the simulation, a solution to this problem was not seeked.

## 5.5 Results

Overall aerodynamic performance is presented in Table 5.2. Experimental data correspond to the reading 190 [62], at 100% blade speed of 16042 $rpm$, which is the best match between experimental and numerical mass flow.

| | **Near Peak Efficiency** | |
|---|---|---|
| | Experimental | Numerical |
| Rotor total pressure ratio (-) | 1.526 | 1.469 |
| Stage total pressure ratio (-) | 1.502 | 1.403 |
| Rotor total temperature ratio (-) | 1.153 | 1.153 |
| Stage total temperature ratio (-) | 1.153 | 1.153 |
| Rotor adiabatic efficiency (%) | 0.836 | 0.759 |
| Stage adiabatic efficiency (%) | 0.803 | 0.664 |
| Rotor polytropic efficiency (%) | 0.845 | 0.772 |
| Stage polytropic efficiency (%) | 0.814 | 0.679 |
| Mass weight ($\mathrm{kg\,m^{-3}}$) | 32.87 | 30.02 |
| Flow coefficient (-) | 0.453 | 0.347 |

Table 5.2: Overall aerodynamic performance

Due to the lack of time, appropriate validation could not be done. There might

exist better points to match experimental data although pressure ratio decreases with increasing mass flow. Nevertheless, given the coarse level of the mesh the results are acceptable.

Experimental data from reading 190 at approximately 10% span shows a flow angle of $0\,^{\circ}$and $41\,^{\circ}$at rotor's inlet and outlet, respectively. Concerning the stator, experimental data shows a flow angle of $-37.5\,^{\circ}$and $-0.2\,^{\circ}$at stator's inlet and outlet, respectively. Numerical flow angle at 10% span is illustrated in Figure 5.4 and shows good agreement with experimental data described above.

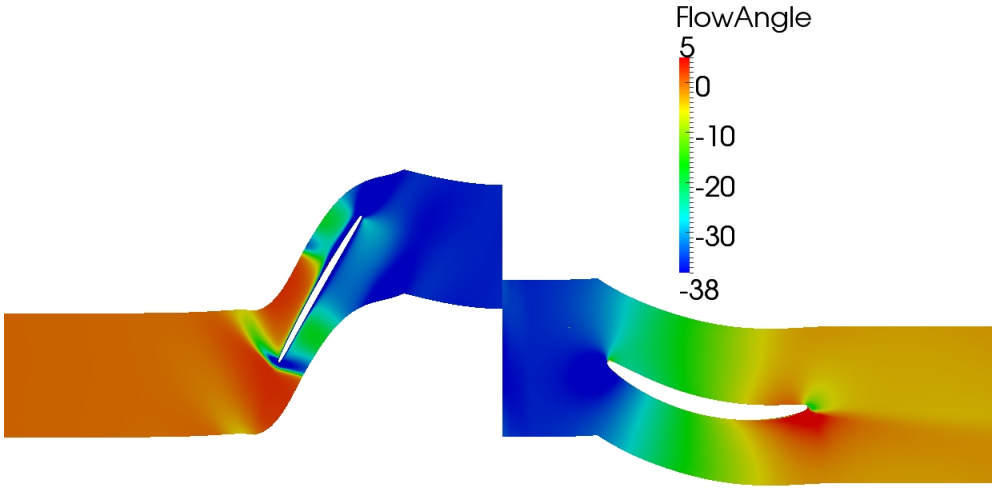



Figure 5.4: Flow angle at 10% span; left blade - rotor; right blade - stator

### 5.5.1 Remarks

This section was meant to use the Mixing Plane approach to simulate multi-row problems. A validation was not attempted due to the lack of time and even the unique simulation presented here can't be said converged. Indeed, there was a reduction of three orders of magnitude in $L^2$ and $L^\infty$ metrics but mass flow didn't remain constant below 0.1%. A very likely justification is the level of the mesh being so coarse and too high static pressure as outlet boundary condition. Nevertheless, despite the simplicity of the simulation, i.e. wall functions, no tip-gap and coarse mesh, preliminary

results seem promising and very interesting work could be done with OpenFOAM, when solving multi-row problems.

# CHAPTER 6

# CONCLUDING REMARKS

The validation of an airplane's fan, the NASA Rotor 67, has been made with the open-source software OpenFOAM. A density based algorithm supported by a Godunov-like scheme discretized the average Navier-Stokes equations, which were the main model of this numerical simulation.

The motivation for this work had two components: the first was the need of more validation cases to safely predict the physics of machines that surround us everyday and which we end up to depend on. The second was being able to do so with a software that is thought to be used by the scientific community, free of charge, and proves itself as a great tool not only to industry but also for academic research and teaching.

A big range of subjects was mentioned here, from pure mathematics to the fundamentals of fluid dynamics and engineering applications. All of these were thought to be presented with such a structure that the answers to the following questions could be induced: how can *Computational Fluid Dynamics* predict a compressor flow; can it be proved that it is correct and what can be expected to see as a result.

One dimensional Riemann problems were done to see how much accurate could this solver be in presence of shocks and how much cells would it need to represent one: good results arose, with considerable gains by switching on the spatially second order extensions.

Regarding the NASA Rotor 67, numerical results are good if one pays attention to statistical tools. Along the streamlines, all plots of the R67_GAP have a *Concordance Index* of, at least, 90% and goes above 94% at 10% span: the closest survey to the tip gap. Indeed, the mesh that reproduces a tip-gap passed expectations as it turned out to predict the flow structure across the rotor considerably better than a finner mesh. From the pitch-wise plots no significant trend can be seen and results are worse than expected. This is attributed to the assumption of the linear stacking of blade's sections, which can cause the wiggles seen near 0% and 100% chord of stream-wise plots. Nevertheless, every major flow structure known to exist a-priori was captured by the HLLC: tip gap vortex, trailing edge vortex and shock, thus proving the reliabiliy of this solver.

There were some difficulties to extract numerical data in the correct positions because open-source tools are still in development. Nevertheless, it is remarkable what can be already done with free software. Difficulties also arise when installing and learning OpenFOAM but essentially, "all" that is needed is patience and persistence.

Finally, a complete compressor stage (multi-row) was run to show that OpenFOAM has a variety of features, namely the *Mixing-Plane*, that allows complex simulations to be made. Although only one simulation was carried out and at a coarse level, overall aerodynamic data agreed reasonably.

## 6.1 Future Work

It was stressed that particularly related to this work, the convergence time and initial data were the major issues that could be addressed. One can reduce the convergence time by implementing other accelerating convergence schemes like IRS or *Multigrid*, mentioned earlier. Multigrid is already under development but the IRS is relatively simple and reduces the convergence time to half [64].

About the initial data, Borm also implemented an open-source *C++* code that runs outside OpenFOAM called *Streamline Curvature Method*. Basically, it is a two dimensional solver for turbomachinery that runs very fast and, after some converting

steps, can be mapped into OpenFOAM's initial fields. This is important to reduce convergence time but it has other purpose: in highly deviated blades, a bad initial data can blow up the simulation otherwise a scale of blade velocity and $C_{cfl}$ must be done, in order to keep the solver stable.

The present work concerned almost only steady state simulations, which is not a true assumption when simulating turbomachinery. One considers that unsteady phenomena dominate when in presence of multi-row problems but to be precise, as pointed out earlier, secondary flows inside turbomachines are seen as the difference between the inviscid and viscid solution. Because secondary flows appear as vortices through the rotor, it isn't definitely a steady problem, even if we are in the blade frame of reference. Of course, computational costs are the major problem to accomplish this.

Unsteady simulations are, if computationally affordable, a very interesting future work. Application to a single row follows naturally whether for multi-row one has different possibilities. The *Domain Scaling* (DS) approach to catch the rotor-stator interaction is already implemented in OpenFOAM with the boundary condition *overlapGGI*. The DS scales the number of blades in the stator with the number of blades in the rotor and only the result is meshed. The other option which is harder to implement, but with significant reduction in computational cost, is the *Nonlinear Harmonic Method* [65]. The conservative variable is decomposed into a time-averaged value and into a sum of periodic perturbations which in turn can be decomposed into $N$ harmonics, characteristic of the problem itself (blade passing frequency, vortex shedding frequency).

Aeroelastic problems can also be addressed with FSI simulations which are implemented in OpenFOAM's transonic density based solvers, with some limitations [8]. Finally, all of the mentioned above plus the main object of this thesis enclosures the turbomachinery's numerical simulation *state of art*.

# BIBLIOGRAPHY


[1] R. Royce, *The Jet Engine*. Rolls Royce, -.

[2] S. A. Korpela, *Principles of Turbomachinery*. Wiley, 2011.

[3] G. Tang, *Measurements of the Tip-gap Turbulent Flow Structure in a Low-speed Compressor Cascade*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2004.

[4] R. A. Van den Braembussche, "Optimization of radial impeller geometry," *NATO*, no. RTO-EN-AVT-143, 2006.

[5] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 2nd ed., 1999.

[6] A. J. Strazisar *et al.*, "Laser anemometer measurements in a transonic axial-flow fan rotor," Tech. Rep. 2879, NASA, 1989.

[7] A. Bakker, *Lecture 11 – Boundary Layers and Separation*, 2002.

[8] O. Borm *et al.*, "Density based navier stokes solver for transonic flows," *OpenFoam 6th workshop*, 2011.

[9] D. Anderson Jr, *Fundamentals of Aerodynamics*. McGraw-hill, 1984.

[10] F. M. White, *Fluid Mechanics*. McGraw-hill, 1994.

[11] W. R. Hawthorn, "Secondary vorticity in stratified compressible fluids in rotating systems," vol. CUED/A-Turbo/TR 63.

[12] O. Borm *et al.*, "Transonic density-based solver." http://openfoam-extend.git.sourceforge.net/git/gitweb-index.cgi, (Consulted 09.03.13).

[13] H. Luo, J. D. Baum, and R. Löhner, "On the computation of multi-material flows using ale formulation," *J. Comput. Phys.*, vol. 194, pp. 304–328, Feb. 2004.

[14] B. Van Leer, "Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection," *J. Comput. Phys.*, 1977.

[15] H. K. Versteeg and W. Malalasekera, *An Introdutcion to Computational Fluid Dynamics, The Finite Volume Method*. Longman Scientific & Technical, 1995.

[16] B. Landmann, *A parallel discontinuous Galerkin code for the Navier-Stokes and Reynolds-averaged Navier-Stokes equations*. PhD thesis, Universitat Stuttgart, 2008.

[17] O. Zanotti and G. M. Manca, "A very short introduction to godunov methods." Albert Einstein Institute, Max-Planck-Institute for Gravitational Physics, Potsdam, Germany, 2010.

[18] T. Y. Hou and G. L. Flock, "Why non conservative solutions converge to wrong solutions : error analysisfloch," *MATHEMATICS OF COMPUTATION*, vol. 62, pp. 497–530, April 1994.

[19] P. D. Lax and B. Wendroff, "Difference schemes for hyperbolic equations with high order of accuracy," *Communications on Pure and Applied Mathematics*, vol. 17, no. 3, pp. 381–398, 1964.

[20] K. G. William, "Lectures in turbulence for the 21st century."

[21] P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," tech. rep., AIAA-92-0439, 1992.

[22] D. Wilcox, "Reassessment of the scale-determining equation for advanced turbulence models," *AIAA Journal*, vol. 26, no. 11, pp. 1299–1310, 1988.

[23] F. R. Menter, "Zonal two equation k-omega turbulence models for aerodynamics flows," *American Institute of Aeronautics and Astronautics*, 1993.

[24] F. Schmitt, "About boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," *Comptes Rendus Mécanique*, vol. 335, pp. 617–627, 2007.

[25] B. Eisfeld, "Computation of complex compressible aerodynamic flows with a reynolds stress turbulence model," *Int. Conference on Boundary and Interior Layers*, 2006.

[26] Gönç *et al.*, "Computation of turbulent flows around rotating bodies using unstructured grids," *Ankara International Aerospace Conference*, no. AIAC-2005-078, 2005.

[27] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics friedrichs," *Mathematische Annalen*, vol. 100, pp. 32–74, 1928.

[28] A. Harten, P. D. Lax, and B. van Leer, "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *SIAM Review*, vol. 25, no. 1, pp. 35–61, 1983.

[29] E. Toro, M. Spruce, and W. Speares, "Restoration of the contact surface in the HLL-Riemann solver," *Shock Waves*, vol. 4, pp. 25–34, July 1994.

[30] H. Hugoniot, "Sur la propagation du mouvement dans les corps et spécialment dans les gaz parfaits," *Journal de l'École polytechnique*, p. Part 1, 1887.

[31] J. M. McDonough, *Introductory Lectures on Turbulence*. Departments of Mechanical Engineering and Mathematics, University of Kentucky, 2007.

[32] U. Frisch, *Turbulence. The legacy of A. N. Kolmogorov.* Cambridge University Press, 1995.

[33] D. A. Johnson and L. S. King, "Mathematically simple turbulence closure model for attached and separated turbulent boundary layers," *AIAA Journal*, vol. 23, no. 11, pp. 1684–1692, 1985.

[34] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA Journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

[35] L. Davidson, "Turbulence modelling," tech. rep., Chalmers University of Technology, 2006.

[36] F. R. Menter, "Improved two-equation k-omega turbulence models for aerodynamic flows," *NASA TM 103975*, 1992.

[37] P. R. Spalart and C. L. Rumsey, "Effective inflow conditions for turbulence models in aerodynamic calculations," *AIAA Journal*, vol. 45, no. 10, pp. 2544–2553, 2007.

[38] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems.* Wiley-Interscience, 2nd ed., 1967.

[39] V. Venkatakrishnan, "Convergence to steady state solutions of the euler equations on unstructured grids with limitersvenkatakrishn," *J. Comput. Phys.*, vol. 118, pp. 120–130, 1995.

[40] T. J. Barth and D. C. Jespersen, "The design and application of upwind schemes on unstructured meshes," *Aerospace Sciences Meeting, 27th*, no. AIAA 89-0366, 1989.

[41] P. Roe, "Characteristic-based schemes for the euler equations," *Annual Review of Fluid Mechanics*, vol. 18, pp. 337–365, 1986.

[42] A. Jameson *et al.*, "Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes," *AIAA 14th Fluid and Plasma Dynamic Conference*, 1981.

[43] A. Jameson, "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings," *AIAA 10th Computational Fluid Dynamics Conference*, no. AIAA 91-1596, 1991.

[44] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001.

[45] A. Andrea, M.-S. Liou, and L. A. Povinelli, "Multigrid time-accurate integration of navier-stokes equations," Tech. Rep. NASA TM 106373, NASA, 1993.

[46] A. Jameson, "Transonic flow calculation," Tech. Rep. MAE Report 1651, Princeton University, 1983.

[47] R. C. Swanson, E. Turkel, and J. A. White, "An effective multigrid method for high-speed flows," Tech. Rep. ICASE Report No. 91-56, NASA, 1991.

[48] V. Venkatakrishnan and D. J. Mavriplis, "A 3d agglomeration multigrid solver for the reynolds-averaged navier-stokes equations on unstructured meshes," Tech. Rep. ICASE Report No. 95-30, NASA, 1995.

[49] D. Strauss and J. L. F. Azevedo, "On the development of an agglomeration multigrid solver for turbulent flows," *J. of the Braz. Soc. of Mech. Sci. & Eng.*, 2003.

[50] E. Kreyszig, *Introductory Functional Analysis with Applications*. JOHN WILEY & SONS, 1978.

[51] H. Brezis, *Functional Analysis, Sobolov spaces and Partial differential equations*. Springer, 2010.

[52] H. Jasak and H. Rusche, "Five basic classes in openfoam," *Wikki*, vol. Advanced Training at OpenFOAM workshop, 2010.

[53] H. Jasak *et al.*, "A tensorial aproach to computational continuum mechanics using object oriented tecnhiques," *American Institute of Physics*, 1998.

[54] H. Jasak, "Handling parallelisation in openfoam," *Cyprus Advanced HPC Workshop*, 2012.

[55] G. A. Sod, "A survey of several finite difference methods for systems of non linear hyperbolic conservation laws," *J. Comput. Phys.*, vol. 27, pp. 1–31, 1977.

[56] R. society (GB), "Philosophical transactions of the royal society of london : giving some accompt of the present undertakings, studies, and labours of the ingenious in many considerable parts of the world," *Royal society (London)*, vol. 170, pp. 277–288, 1870.

[57] H. Hugoniot, "Sur la propagation du mouvement dans les corps et spécialment dans les gaz parfaits," *Journal de l'École polytechnique*, p. Part 2, 1889.

[58] P. Woodward and P. Colella, "The numerical simulation of two-dimensional fluid flow with strong shocks," *J. Comput. Phys.*, vol. 54, pp. 115–173, 1984.

[59] O. Borm *et al.*, "turbovtk python scripts." http://sourceforge.net/projects/turbopost/, (Consulted 09.03.13).

[60] O. Borm, "Unsteady aerodynamics of a centrifugal compressor stage – validation of two different cfd solvers," *Proceedings of ASME Turbo Expo*, 2012.

[61] Open Source CFD International Conference, *Development of a Generalized Grid Interface for Turbomachinery Simulations with OpenFOAM*, 2008.

[62] W. S. Cunnan *et al.*, "Design and performance of a 427-meter-per-second-tip-speed two stage fan having a 2.4 pressure ratio," Tech. Rep. 1314, NASA, 1978.

[63] M. Beaudoin *et al.*, "Sig turbo machinery openfoam and mixing plane," *7th OpenFOAM Workshop*, 2012.

[64] M. Anderson, "Turbomachinery validation with the density based openfoam solver," *OpenFoam 7th workshop*, vol. IntegralX Corporation, 2012.

[65] S. Vilmin *et al.*, "The nonlinear harmonic method for rotor-stator interactions applied to thermally perfet gas," *Proceedings of the 8th International Symposium on Experimental and Computational Aerothermodynamics of Internal Flows*, 2007.

# APPENDIX A

# MESH QUALITY REPORT

Here are presented the quality reports of the meshes used in this thesis. *Mesh stats* informs about the number of points, cells and boundaries while *Mesh topology* informs if they are well connected. *Checking patch topology* informs the name of the boundaries and *Checking geometry* informs about skewness, non-orthogonality, aspect ratio, etc, which are closely related to mesh quality. Finally, it gives the output *Mesh OK* if every quality test is passed.

## A.1 R67_C

```
...
Mesh stats
    all points:            290779
    live points:            290779
    all faces:             835424
    live faces:             835424
    internal faces:    798976
    cells:             272400
    boundary patches: 7
    point zones:       0
    face zones:        4
    cell zones:        8
```

```
Overall number of cells of each type:
    hexahedra:     272400
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0

Checking topology...
    Boundary definition OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces ...
    Patch              Faces    Points   Surface topology
    per1               7900     8080     ok (non-closed singly connected)
    per2               7900     8080     ok (non-closed singly connected)
    blade              10000    10100    ok (non-closed singly connected)
    shroud             2724     2879     ok (non-closed singly connected)
    hub                2724     2879     ok (non-closed singly connected)
    inlet              2800     2929     ok (non-closed singly connected)
    outlet             2400     2525     ok (non-closed singly connected)

Checking geometry...
    This is a 3-D mesh
    Overall domain bounding box (0.0859758 -0.0383107 -0.0684196) (0.257
        0.150445 0.354423)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Mesh (non-empty, non-wedge) dimensions 3
    Boundary openness (3.44438e-16 -7.23005e-16 -2.64524e-16) Threshold =
        1e-06 OK.
```

```
    Max cell openness = 2.43211e-15 OK.
    Max aspect ratio = 555.072 OK.
    Minumum face area = 1.98755e-09. Maximum face area = 8.71777e-05.
        Face area magnitudes OK.
    Min volume = 1.49226e-13. Max volume = 1.36261e-07.  Total volume =
        0.00275568.  Cell volumes OK.
    Mesh non-orthogonality Max: 60.4634 average: 18.3499 Threshold = 70
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 2.2858 OK.

Mesh OK.
...
```

## A.2 R67_I

```
...
Mesh stats
    all points:            442986
    live points:            442986
    all faces:             1277976
    live faces:             1277976
    internal faces:    1227624
    cells:             417600
    boundary patches: 7
    point zones:       0
    face zones:        4
    cell zones:        8

Overall number of cells of each type:
    hexahedra:      417600
    prisms:         0
    wedges:         0
    pyramids:       0
    tet wedges:     0
    tetrahedra:     0
```

```
    polyhedra:      0

Checking topology...
    Boundary definition OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces ...
    Patch               Faces    Points   Surface topology
    per1                10600    10807    ok (non-closed singly connected)
    per2                12200    12423    ok (non-closed singly connected)
    blade               12800    12928    ok (non-closed singly connected)
    shroud              4176     4386     ok (non-closed singly connected)
    hub                 4176     4386     ok (non-closed singly connected)
    inlet               4000     4141     ok (non-closed singly connected)
    outlet              2400     2525     ok (non-closed singly connected)

Checking geometry...
    This is a 3-D mesh
    Overall domain bounding box (0.0860314 -0.0352895 -0.068584) (0.257012
        0.149137 0.354423)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Mesh (non-empty, non-wedge) dimensions 3
    Boundary openness (-3.10694e-16 -2.21482e-15 5.05216e-16) Threshold =
        1e-06 OK.
    Max cell openness = 2.66918e-15 OK.
    Max aspect ratio = 302.924 OK.
    Minumum face area = 1.9965e-09. Maximum face area = 4.68775e-05.  Face
        area magnitudes OK.
    Min volume = 1.49544e-13. Max volume = 7.32467e-08.  Total volume =
        0.00278774.  Cell volumes OK.
    Mesh non-orthogonality Max: 61.1455 average: 17.6942 Threshold = 70
    Non-orthogonality check OK.
```

```
    Face pyramids OK.
    Max skewness = 2.29269 OK.

Mesh OK.
...
```

## A.3 R67_GAP

```
...
Mesh stats
    all points:            512078
    live points:            512078
    all faces:             1481180
    live faces:             1481180
    internal faces:    1426660
    cells:             484640
    boundary patches: 7
    point zones:       0
    face zones:        4
    cell zones:        10

Overall number of cells of each type:
    hexahedra:      484640
    prisms:         0
    wedges:         0
    pyramids:       0
    tet wedges:     0
    tetrahedra:     0
    polyhedra:      0

Checking topology...
    Boundary definition OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).
```

```
Checking patch topology for multiply connected surfaces ...
    Patch               Faces    Points   Surface topology
    per1                12900    13130    ok (non-closed singly connected)
    per2                11300    11514    ok (non-closed singly connected)
    blade               13312    13389    ok (non-closed singly connected)
    shroud              6184     6338     ok (non-closed singly connected)
    hub                 4424     4653     ok (non-closed singly connected)
    inlet               4000     4141     ok (non-closed singly connected)
    outlet              2400     2525     ok (non-closed singly connected)

Checking geometry...
    This is a 3-D mesh
    Overall domain bounding box (0.0860282 -0.0352269 -0.0768108)
        (0.257014 0.149137 0.354423)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Mesh (non-empty, non-wedge) dimensions 3
    Boundary openness (-6.89563e-16 -1.69678e-16 -1.45567e-16) Threshold =
         1e-06 OK.
    Max cell openness = 3.63818e-15 OK.
    Max aspect ratio = 420.229 OK.
    Minumum face area = 2.66526e-11. Maximum face area = 5.55851e-05.
        Face area magnitudes OK.
    Min volume = 1.01696e-15. Max volume = 1.22299e-07.  Total volume =
        0.00285691.  Cell volumes OK.
    Mesh non-orthogonality Max: 66.438 average: 17.1283 Threshold = 70
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 2.51257 OK.

Mesh OK.
...
```

## A.4 R67_F

```
...
Mesh stats
    all points:           830624
    live points:           830624
    all faces:            2417536
    live faces:            2417536
    internal faces:   2344064
    cells:            793600
    boundary patches: 7
    point zones:      0
    face zones:       4
    cell zones:       8

Overall number of cells of each type:
    hexahedra:     793600
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0

Checking topology...
    Boundary definition OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces ...
    Patch               Faces    Points   Surface topology
    per1                15600    15857    ok (non-closed singly connected)
    per2                15600    15857    ok (non-closed singly connected)
    blade               18400    18584    ok (non-closed singly connected)
    shroud              7936     8224     ok (non-closed singly connected)
    hub                 7936     8224     ok (non-closed singly connected)
```

```
    inlet                4800      4949      ok (non-closed singly connected)
    outlet               3200      3333      ok (non-closed singly connected)

Checking geometry...
    This is a 3-D mesh
    Overall domain bounding box (0.0857991 -0.0316234 -0.0749792)
        (0.257028 0.143075 0.354423)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Mesh (non-empty, non-wedge) dimensions 3
    Boundary openness (-1.43693e-16 2.53705e-15 6.174e-16) Threshold = 1e
        -06 OK.
    Max cell openness = 1.80276e-15 OK.
    Max aspect ratio = 253.388 OK.
    Minumum face area = 1.94024e-09. Maximum face area = 2.8986e-05.  Face
         area magnitudes OK.
    Min volume = 1.47858e-13. Max volume = 4.51918e-08.  Total volume =
        0.00282741.  Cell volumes OK.
    Mesh non-orthogonality Max: 68.4792 average: 17.2376 Threshold = 70
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 2.29946 OK.

Mesh OK.

...
```

# APPENDIX B

# RIEMANN PROBLEM'S SIMULATION PARAMETERS

A mesh was created in the *blockMeshDict* with 10 meters, composed by 1000 uniform cells. By setting all directions except $X$ to empty, one can make one dimensional simulations in OpenFOAM. The dictionary reads

```
convertToMeters 1;

vertices
(
    (-4.5 -0.1 -0.1)\\V0
    (5.5 -0.1 -0.1)\\V1
    (5.5 0.1 -0.1)\\V2
    (-4.5 0.1 -0.1)\\V3
    (-4.5 -0.1 0.1)\\V4
    (5.5 -0.1 0.1)\\V5
    (5.5 0.1 0.1)\\V6
    (-4.5 0.1 0.1)\\V7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (1000 1 1) simpleGrading (1 1 1)
);
```

```
edges
(
);

patches
(
    patch sides
    (
        (1 2 6 5)
        (0 4 7 3)
    )
    empty empty
    (
        (0 1 5 4)
        (5 6 7 4)
        (3 7 6 2)
        (0 3 2 1)
    )
);

mergePatchPairs
(
);
```

Althought our domain of interest in the Riemann problem is just $1m$ around $x_0$, a bigger numerical domain was choosen so that shock waves can't reflect and interfere with the solution.

Then, because we will use the *transonicUnsteadyMRFDyMFoam*, some dictionaries have to exist despite no specification is there: it is the case of *MRFzones*. The *turbulenceProperties* dictionary is set to RASModel and *RASModel* dictionary is set to laminar. The *thermophysicalProperties* is set as follows

```
thermoType      hPsiThermo<pureMixture<constTransport<specieThermo<
    hConstThermo<perfectGas>>>>>;

// name, nMoles, mol weight, CP, Hf, mu, Pr;
```

```
//mixture          air 1 28.9 1007 0 0 0.7;
mixture          normalisedGas 1 11640.3 2.5 0.0 0.0 1.0;
```

It can be seen the gas is modeled as a normalized gas. This is done so that we can use the original problem initial states and compare with its solution. Note that dynamic viscosity (mu) is also set to zero thus we are only solving the Euler equations which is the objective: evaluate the performance of the HLLC to solve a hyperbolic system.

Finally, the time step in *controlDict* is taken to be 0.001 with a CFL of 0.18 in agreement with Toro [5].