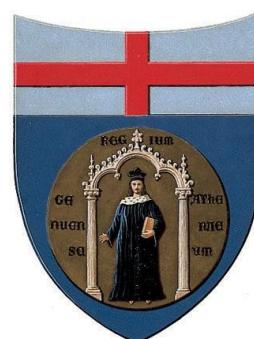


UNIVERSITY OF GENOVA
POLYTECHNIC SCHOOL
DIME
Department of Mechanical, Energy, Management
and Transportation Engineering



MASTER OF SCIENCE THESIS
IN
MECHANICAL ENGINEERING

**Development of a Navier Stokes Axisymmetric
Throughflow Model in Numeca Openlabs Environment**

Supervisor:

Chiar.^{mo} Prof. Ing. Pietro Zunino

Co Supervisor:

Dott. Ing. Luca Ratto

Candidate:

Riccardo Rubini

Settembre 2016

Abstract

The aim of this project at Von Karman Institute is to develop a throughflow model for the analysis of axial turbomachinery. In particular I focused my attention on the implementation and physical validation of the mathematical model.

All the work has been developed in NUMECA OpenLabs; OpenLabs is a module of FINE/Open that allows the user to operate on the flow balance equations.

Three meshes have been created with HEXPRESS in order to reach the grid independence of the model both for the inviscid calculations and the viscous ones.

Regarding the central part of my work as first thing some inviscid calculations have been performed, with a model based on Euler equations and compared with a mid-line code in order to ensure the reliability of the code.

Subsequently viscous calculations have been performed developing a model based on the Navier Stokes equations averaged according to the theory proposed by Adamczyk.

In the last part viscous losses are introduced by empirical correlations and the modelling of deterministic stresses addressed.

In order to account for the deterministic unsteadiness affecting the flow, a Bousinnesq like hypothesis has been done introducing a new transport equation for the deterministic viscosity.

Results have been qualitatively compared with other computations obtained before and the impact of the deterministic stresses has been evaluated.

Contents

List of Figures	iv
List of Symbols	vi
1 Introduction	1
1.1 ThroughFlow modelling	1
1.2 Euler based Throughflow	2
1.3 Navier Stokes based Throughflow	2
1.4 High Order Throughflow	2
2 Throughflow Model	4
2.1 Adamczyk averaging	4
2.1.1 Ensemble averaging	5
2.1.2 Passage-to-passage averaging	7
2.1.3 Circumferential averaging operator	8
2.2 Throughflow model closure: Inviscid Blade Forces	9
2.2.1 Blade blockage contribution	10
2.2.2 Inviscid blade forces	11
2.2.3 Leading edge treatment	13
2.2.4 Shock capturing properties	13
2.3 Viscous Blade Forces	14
2.3.1 Distributed loss model	14
2.3.2 Profile losses	15
2.3.3 Endwalls Boundary Layer Losses	15
2.3.4 3D Losses	16
2.3.5 Effect of the Tip Clearance on the Flow Deflection	17
2.4 Deterministic and circumferential stresses	17
3 Numerical Tools	21
3.1 Space discretization	21
3.1.1 Flow domain	22
3.1.2 Inviscid Flux	23
3.1.3 Viscous Flux	25
3.2 Time Discretization	25
3.3 Boundary conditions	26

4 Openlabs	28
4.1 Openlabs Environment	28
4.2 Throughflow implementation in Openlabs	30
4.2.1 Constant Properties Definitions	31
4.2.2 Equations	31
4.2.3 Source Terms	34
5 Results	36
5.1 Comparison with mid-line code	36
5.1.1 Radial Sections	37
5.2 Comparison with NISRE and CFD calculation	41
5.2.1 Radial and meridional sections	42
6 Conclusions and future work	48
References	51
A Appendix A	54
A.0.1 CONSTANT	54
A.0.2 EQUATIONS	56
A.0.3 CUSTOM BOUNDARY CONDITIONS	61
A.0.4 ALGEBRAIC DIFFUSION COEFFICIENTS	61
A.0.5 INITIAL PROFILES	61
A.0.6 SOURCETERMS	61
A.0.7 AUXTERMS	63
B Appendix B	66
B.1 Impact of Deterministic Stresses	66
C Appendix C	70
C.1 Deterministic Viscosity Closure	70
C.2 Source Term Computation	70

List of Figures

2.1	Axial distribution of the blockage factor	11
2.2	S_1 and S_2 surfaces	11
2.3	lean angle	12
2.4	Loss coefficient and deviation angle	17
2.5	Radial distribution of momentum and forces	18
2.6	Deflection with blade force modification (blue) and without (red) . .	19
3.1	Example of viscous mesh	22
3.2	Boundary layer	23
3.3	Diamond control volume approach for viscous computation	25
3.4	Conceptual illustration of wall function (left) and near wall treatment(right)	26
4.1	Openlabs' illustration	28
4.2	Openlabs' editor GUI	29
5.1	Absolute flow angle stator outlet	38
5.2	Relative flow angle stator outlet	38
5.3	Tangential velocity stator outlet	38
5.4	Axial velocity stator outlet	38
5.5	Static pressure stator outlet	39
5.6	Static temperature stator outlet	39
5.7	Absolute Mach number stator outlet	39
5.8	Specific massflow stator outlet	39
5.9	Relative flow angle rotor outlet	40
5.10	Absolute mach number rotor outlet	40
5.11	Tangential velocity rotor outlet	40
5.12	Axial velocity rotor outlet	40
5.13	Static pressure rotor outlet	41
5.14	Static temperature rotor outlet	41
5.15	Mach number in a blade to blade section at hub	42
5.16	Mach number in a blade to blade section at tip	42
5.17	Static Pressure from CFD calculation	43
5.18	Static Pressure from throughflow calculation	43
5.19	Static temperature from CFD calculation	44

5.20	Static temperature from throughflow calculation	44
5.21	Absolute flow angle stator outlet	44
5.22	Absolute velocity stator outlet	44
5.23	Axial velocity stator outlet	45
5.24	Tangential velocity stator outlet	45
5.25	Absolute Mach number stator outlet	45
5.26	Relative mach number stator outlet	45
5.27	Static pressure stator outlet	45
5.28	Total pressure stator outlet	45
5.29	Absolute flow angle rotor outlet	46
5.30	Relative flow angle rotor outlet	46
5.31	Axial velocity rotor outlet	46
5.32	Tangential velocity rotor outlet	46
5.33	Static pressure rotor outlet	46
5.34	Total pressure rotor outlet	46
5.35	Static temperature rotor outlet	47
5.36	Total temperature rotor outlet	47
B.1	Absolute Velocity	67
B.2	Absolute Mach Number	67
B.3	Relative Mach Number	67
B.4	Static Temperature	67
B.5	Total Temperature	68
B.6	Total Pressure	68
B.7	Radial-Radial Stress	69
B.8	Axial-Axial Stress	69
C.1	Example of the blade row interaction	71
C.2	Basic Idea of the model	71

List of Symbols

Acronyms

LDST	Lumped deterministic stress tensor
RANS	Reynolds averaged equations
TF	Throughflow model
VKI	von Karman Institute
N-S	Navier-Stokes equations

Roman symbols

\mathcal{D}	diffusion coefficient	-
Da	Damköhler number	-
h	specific enthalpy	J/kg
H	total entalpy	$J/kg J$
mass flux	$kg/(m^2 s)$	
k	thermal conductivity	$W/(kgK)$
p	pressure	Pa
q	heat flux	W/m^2
R	radius	m
S_i	balance equations' source terms	
S_g	entropy generated	$J/kg/K T$
temperature	K	
u	x- (radial) component of velocity	m/s
U	absolute value of absolute velocity	m/s
v	specific volume	m^3/kg
W	absolute value of relative velocity v	y - (tangential) component of velocity
m/s		
w	z- (axial) component of velocity	m/s

Greek symbols

α	absolute flow angle	-
β	relative flow angle	-
Γ	general diffusion coefficient	Pa s
δ	boundary layer thickness	m
δ^*	displacement thickness	m
λ	Jacobian eigenvalue	
μ	dinamic viscosity	Pa s
ρ	density	kg/m^3

Sub- and Superscripts

amb	value relative to the ambient
d	deterministic
eff	effective
tot	total
w	value referred to the wall

Chapter 1

Introduction

1.1 ThroughFlow modelling

The meridional analysis of the turbomachinery has a quite long history; the idea of radial equilibrium, coupled with a vortex law to guess the circumferential velocity distribution, was probably born around 1950's. After a decade of the introduction of the idea of radial equilibrium the more complex throughflow model began to replace the simpler radial equilibrium codes; in this period the most widespread throughflow model was certainly the streamline curvature method. Subsequently, thanks to the development of the CFD techniques and the growing computational power of the computers the first 3D CFD computations started to appear. Due to the increasing development of the computer capabilities the throughflow models ceased to be the only design tool in the development of turbomachinery and began to be the first step of the complex chain of development of compressors and turbines. However the throughflow remains the backbone of the preliminary design of turbomachinery: in fact it provides a first insight of the global components functioning. The throughflow model may be classified as a dynamic method of approximation in which a set of equations are solved; the equations which are not solved are replaced by assumption coming from experience or from more detailed computations. The major drawback of the throughflow modelling is that it heavily relies on empirical inputs, such as empirical correlations in order to capture 2,3-D viscous effects and flow structure. The empirical tuning needed to calibrate the throughflow model allows it to perform quite accurate predictions of the flow field inside the whole machine, however this is only possible for machines with geometrical and physical parameters close to the reference machine that has been used to calibrate the model. In this work the Euler throughflow was first considered and the inviscid blade forces and the blockage factor were implemented. The following step was to take into account the viscous forces and 2D-3D losses. In order to introduce the viscous losses in the model there are two main ways: the first is to introduce viscous losses in the Euler equations the second is to solve directly the Navier-Stokes equations, in this work the second way has been followed. Then we introduce correlations for the secondary losses at hub and tip; at last deterministic stresses have been taken into account according to the average passage set of

equations developed by Adamczyk in the early eighties(1984).

1.2 Euler based Throughflow

The first throughflow method based on the Euler equation was already presented around 1980 by Spurr. Spurr's interest was to develop a tool for the analysis of the steam turbines with Mach number around two, value extremely high for such a kind of machine. The most interesting features of Spurr's model is the fact that Spurr used the time marching technique in order to solve the Euler system and consequently applies it also to the blade forces. In 1944 Nigmatullin and Ivanov also presented a throughflow model based on the Euler equations, this model is very interesting under two aspects: the first is the use of a monotone second order Godunov scheme, this is one of the very few examples of the application of an exact Riemann solver in a CFD code; the second interesting aspect is the fact that the author recognized the possibility to describe discontinuous blade forces where shocks are captured in analysis mode. One of the most interesting features of the Euler based throughflows is the possibility to capture shocks, predict mass choking and cope with more realistic geometries characterizing real engine. Despite the fact that the throughflow model showed very interesting features however the numerical robustness of the model wasn't well understood. As an example it could require an elaborate start up procedure and numerical problems could arise when modelled physical features are introduced in the model such as incidence and flow deviation are corrected. Liu and al.(2000) demonstrated that the throughflow model is compatible with a wide range of numerical schemes; this author uses Van Leer's flux vector splitting coupled with an explicit predictor corrector time integrating. Nowadays the reference works based on an Euler code can be found in Sturmayr(2004) and Rosa Taddei (2008).

1.3 Navier Stokes based Throughflow

Certainly the throughflow models based on Euler equations show some very interesting features such as shocks capturing properties and the capability to predict choked massflow. A drawback of the Euler based models is the impossibility to describe annulus end-wall viscous losses. A step further in order to reach a better physical reliability is to develop a throughflow model based on the Navier-Stokes equations that is able to resolve the viscous flow at the annulus end-walls and automatically compute the flow blockage due to boundary layer at the end-walls. The major work in this direction is performed by Simon(2007).

1.4 High Order Throughflow

Another step in order to make the throughflow models less empiricism dependent, is the use of the Adamczyk's(1984) average procedure; the result of this process is a steady flow field which contains the averaged effects of the unsteadiness. The

resulting equation system describes a steady and periodic flow field but contains the mean effects of turbulence, unsteadiness and aperiodicity. These additional terms are Reynolds stresses, deterministic stresses, passage to passage stresses and blade forces. Many authors studied how to deal with these terms and how to understand which terms is predominant on the others and in which physical situation. In literature many authors deal with the problem of the closure of deterministic stresses, the most followed approach is to exploit the mathematical similarity between Reynolds turbulent stresses and deterministic stresses. The deterministic stresses must be correlated with the local strain rate as the strain rate is known in the steady simulation Charbonnier(2004) and Marshall Paul (2008). Stollenwerk (2013) made the assumption that the deterministic stress tensor can be put in relation with the strain rate of the mean field through the "deterministic viscosity". In order to model the evolution of deterministic viscosity in the flow field Stollenwerk proposes a scalar transport equation to add to the existing Navier-Stokes equations. Another more general approach to model deterministic stresses is proposed by Charbonnier and Lebouf (2004); this authors present a set (six) of transport equations to describe each component of the deterministic stress tensor. Another approach to the closure of the deterministic stresses problem is proposed by Galbraith Orkwis (2008) and Sondak Dorney Davis (1996) called LDST (Lumped Deterministic Source Terms) in which the deterministic stresses are interpreted as the residual of the steady state solution. However it's still unclear how to deal with this terms and in which situation you can or cannot ignore part of them or them all, a lot of work on this topic has been performed by J F. Simon and O. Leonard (2008) in which the blade forces and deterministic stresses are determined from 3D steady and unsteady CFD stage calculations and integrated in the Throughflow model, then the results are compared with a more "classical throughflow approach".

Chapter 2

Throughflow modelling

In this chapter will be presented the theoretical model underlying the throughflow modelling of axial turbomachinery. The first step to do in order to describe the flow in a turbomachinery (fully 3D and non stationary) is to perform the Adamczyk (1964) averaging thanks to which the flow-field inside a turbomachinery can be modelled with the axisymmetric Navier Stokes equations after the application of three averaging operators; in order to remove the dependency from the temporal and the circumferential coordinates; unfortunately, due to the non linearity of the equations, this set of operators introduces extra terms in the equations that need a semi empirical modelling. This procedure is widely explained in "Model equations for simulating flow in multistage turbomachinery". Subsequently the inviscid blade forces, viscous blade forces, blockage factor, 2D and 3D losses will be presented. In the last part of this chapter the mathematical and physical aspects of deterministic stresses will be introduced and a closure model for the deterministic stresses based on Boussinesq's like hypothesis will be presented.

2.1 Adamczyk averaging

The flow field in multistage turbomachinery is extremely complex. It is highly unsteady; the time scales ranging from a fraction of the shaft speed to several times the highest blade passing frequencies. The length scales are also very different, they range from the circumference of the machine to a fraction of the blades chord. The great spread between the characteristic scales of the problem makes a direct numerical simulation practically impossible. In order to deal with this complexity and make the simulation less computational demanding, Adamczyk proposed to reduce the 3D unsteady model to a steady axisymmetric one using three averaging operators.

The starting point of the analysis are the non stationary 3D Navier-Stokes equations.

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0 \quad (2.1)$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j + p \delta_{ij})}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \quad (2.2)$$

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho e + p) u_i}{\partial x_i} = \frac{\partial(\tau_{ij} u_j)}{\partial x_i} + \rho f_i u_i + \frac{\partial(q_i)}{\partial x_i} + r \quad (2.3)$$

In cylindrical coordinates, the axial component of the instantaneous Navier-Stokes equations is written:

$$\frac{\partial(\rho V_z)}{\partial t} + \frac{1}{r} \frac{\partial(r \rho V_z V_r)}{\partial r} + \frac{1}{r} \frac{\partial(r \rho V_\theta V_z)}{\partial \theta} + \frac{\partial(\rho V_z V_z + p)}{\partial z} = \frac{1}{r} \frac{\partial(r \tau_{zr})}{\partial r} + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial r} + \frac{\partial \tau_{zz}}{\partial z} \quad (2.4)$$

2.1.1 Ensemble averaging

The first step is to perform a Reynolds ensemble-averaging in order to remove all flow features that are non correlated with the rotation of the shaft, in particular the target of this averaging is to remove every non deterministic fluctuation from the flow field (turbulence, rotating stall or surge) in order to decouple the unsteady deterministic part from the stochastic one. The Reynolds ensemble averaging is obtained by averaging consecutive sets of data taken over a revolution of the shaft.

$$\bar{f}^e(z, r, \theta, t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N f(z, r, \theta, T(n-1)) \quad (2.5)$$

where T is the period of the revolution of the shaft. Applying this operator to the Navier-Stokes equations every field is now decomposed in a mean part and a fluctuating no-deterministic part:

$$f(z, r, \theta, t) = \bar{f}^e(z, r, \theta, t) + f'(z, r, \theta, t) \quad (2.6)$$

For the analysis of a compressible flow is useful to define a density-average (Favre averaging) ensemble averaging instead of the Reynolds one:

$$\tilde{f}^e = \lim_{N \rightarrow \infty} \frac{1}{N \bar{\rho}} \sum_{n=1}^N \rho_n f_n \quad (2.7)$$

where $\bar{\rho}$ is the ensemble average density and ρ_n is the n^{th} realization of the density field. This operation applied on the Navier-Stokes equations introduces extra terms: the Reynolds stresses $\overline{\rho V'_i V'_j}^e$ in the momentum equations and the enthalpy fluxes $\overline{\rho V'_i H'}^e$ in the energy equation.

Time-Averaging Procedure

The second average operation lead to time-averaged flow equations, this operator is applied to the whole fluid domain including the regions with the blade rows. The traditional definition of this operator is:

$$\bar{f}^e = \frac{1}{T} \int_{t_0}^{t_0+T} f^e dt \quad (2.8)$$

where T is the period of revolution of the shaft and t_0 a reference time; this operator needs to be extended to the reference frame also. According to Adamczyk to extend this averaging to all the flow field we must define a gate function for each blade rows.

$$H(z, r, \theta, t) = U(t - t_0) + \sum_{n=0}^{N-1} U(t - t_0 - t_1) - \sum_{n=0}^{N-1} U(t - t_0 - t_2) \quad (2.9)$$

where

$$t_i = \frac{1}{\omega} ((\theta_i(r, z) + \theta_{ref} - \frac{2\pi n}{N}) - \theta) \quad (2.10)$$

The gate function is composed by a finite sum of heavyside functions so that the gate function is equal to 1 inside the flow field and equal to zero inside the blade. The integral of the gate function over the time is equal to the blockage factor.

$$b = \frac{1}{T} \int_{t_0}^{t_0+T} H(r, \theta, z, t) dt = \frac{\theta_{suction} - \theta_{pressure}}{\frac{2\pi}{N}} \quad (2.11)$$

In such a way we obtain a time-averaging operator that is applicable to every region of the space:

$$\bar{f}^e = \frac{\frac{1}{T} \int_{t_0}^{t_0+T} H(r, \theta, z, t) \bar{f}^e(z, r, \theta, t)}{\frac{1}{T} \int_{t_0}^{t_0+T} H(r, \theta, z, t) dt} \quad (2.12)$$

In this way the flow field is decomposed in a steady part and in an unsteady fluctuation part:

$$\bar{f}^e(r, \theta, z, t) = \bar{f}^e(z, r, \theta, t) + f''(r, \theta, z, t) \quad (2.13)$$

It is important to notice that the operator (2.12) outside a rotor passage reduces to the operator (2.8) and under these circumstances the commutator between the time operator and the differentiation is zero. However inside the rotor passage the differentiation operator doesn't commute with the time operator due to the presence of the blades. The rules that links the time averaging operator and the order of spatial differentiation is the following:

$$\frac{\partial \bar{f}^e}{\partial x_i} = \frac{1}{bT} \int_{t_0}^{t_0+T} H(r, \theta, z, t) \frac{\partial \bar{f}^e(r, \theta, z, t)}{\partial x_i} dt \quad (2.14)$$

this can be rearranged as:

$$\frac{\partial \bar{f}^e}{\partial x_i} = \frac{1}{bT} \int_{t_0}^{t_0+T} \frac{\partial (H \bar{f}^e)}{\partial x_i} dt - \frac{1}{bT} \int_{t_0}^{t_0+T} \frac{\partial H}{\partial x_i} \bar{f}^e dt \quad (2.15)$$

Since the limit of integration do not depend from the spatial coordinates the first term left hand side is equal to:

$$\frac{1}{bT} \int_{t_0}^{t_0+T} \frac{\partial(H\bar{f}^e)}{\partial x_i} dt = \frac{1}{b} \frac{\partial b\bar{f}^e}{\partial x_i} \quad (2.16)$$

The second term left hand side contains the derivative of the gate function that is a sum of Dirac's delta; the spatial derivative of the gate function is equal to

$$\frac{\partial H}{\partial x_i} = \frac{1}{\omega} \sum_{n=0}^{N-1} (\delta(t - t_0 - t1) \frac{\partial \theta_1}{\partial x_i} - \delta(t - t_0 - t2) \frac{\partial \theta_2}{\partial x_i}) \quad (2.17)$$

inserting the expression just obtained inside the equation (2.15) and reminding the properties of the integrals containing Dirac's delta we obtain:

$$\frac{\overline{\partial f^e}}{\partial x_i} = \frac{1}{b} \frac{\partial b\bar{f}^e}{\partial x_i} + \frac{1}{2\pi b} \sum_{n=0}^{N-1} (\bar{f}_1^e \frac{\partial \theta_1}{\partial x_i} - \bar{f}_2^e \frac{\partial \theta_2}{\partial x_i}) \quad (2.18)$$

The additional term right hand side is not zero for the physical quantities that are non zero on the blade surface, pressure, shear stresses and heat flux for example.

The application of the time averaging on the RANS equation produces additional terms:

- the deterministic stresses $\overline{\rho^e V_i'' V_j''}^t$ in the momentum equations and the deterministic enthalpy flux $\overline{\rho^e V_i'' H''}^t$ in the energy equations, these terms represent the effects of the unsteady (deterministic) part of the flow on the steady averaged part.
- the blockage factor b due to the geometrical blockage of the row
- blade forces f_b^R and f_v^R (inviscid and viscous blade forces) in the momentum equation e_b^R and e_v^R (work done by the inviscid and viscous blade forces) in the energy equation, these terms are due to the pressure field, shear stresses and heat flux acting on blades surfaces.

2.1.2 Passage-to-passage averaging

In the most general case the number of blades of stages of the machine can change from row to row so the steady flow will not be identical in each passage of a given row. For this reason a third passage operator needs to be introduced: this operator will be developed using of the mathematical tools associated with the Fourier analysis. The aim of this operator is to decompose the flow field in a mean part, periodic in the reference blade row and an aperiodic fluctuating part. However in this thesis we will focus only on a single stage of a machine so we will not need to apply this operator on the equations (for further information a deep discussion is done by Adamczyk (1964)).

2.1.3 Circumferential averaging operator

The last element to obtain an axisymmetric representation of the flow field inside the machine is to circumferential average the averaged-passage equations. In this way we obtain only one flow field common to all average passages flow. The circumferential average operator is performed on the average passage flow of the blade row.

$$\overline{\overline{f}^e}^t = \frac{\frac{1}{\Delta\theta} \int_0^{\Delta\theta} F(r, \theta, z) \overline{f}^e(r, \theta, z) d\theta}{\frac{1}{\Delta\theta} \int_0^{\Delta\theta} F(r, \theta, z) d\theta} \quad (2.19)$$

where $\Delta\theta$ is equal to $\frac{2\pi}{N}$ with N the number of blades, F is a gate function with the value 1 inside the blade row and 0 outside it, such that:

$$b = \frac{1}{\Delta\theta} \int_0^{\Delta\theta} F(r, \theta, z) d\theta \quad (2.20)$$

After this averaging operator the flow field is decomposed:

$$\overline{f}^e(r, \theta, z) = \overline{\overline{f}^e}^t(r, z) + f'''(r, \theta, z) \quad (2.21)$$

In the same way we proceeded for the temporal-average operator this operator don't commute with the partial derivative operator inside the blade passage, for this reason we need to develop a rule that links the pitch-averaging operator with the order of differentiation.

$$\overline{\overline{\overline{f}^e}^t}^c = \frac{1}{b} \int_0^{\Delta\theta} F(r, \theta, z) \frac{\partial \overline{f}^e}{\partial x_i} d\theta \quad (2.22)$$

in a similar manner as (2.14) it becomes

$$\overline{\overline{\overline{f}^e}^t}^c = \frac{1}{b} \frac{\partial b \overline{f}^e}{\partial x_i} - \frac{N}{2\pi b} \left(\overline{f}_s \frac{\partial \theta_s}{\partial x_i} - \overline{f}_p \frac{\partial \theta_p}{\partial x_i} \right) \quad (2.23)$$

with '*s*' and '*p*' indicate the suction and pressure side of the blade.

Due to this averaging additional terms appear in the equations:

- the circumferential stresses in the momentum equations $\overline{\overline{\rho}^e}^t V_i''' V_j'''^c$ and the enthalpy flux $\overline{\overline{\rho}^e}^t V_i''' H'''^c$ in the energy equations, they represent the effect of the non axisymmetric part of the flow on the axisymmetric one.
- the blockage factor resulting from the blade thickness.
- the momentum equations will contain the blades forces f_b f_v due to the pressure field and the shear stresses acting on the blades surface while the energy equation will contain e_b e_v extra terms.

2.2 Throughflow model closure: Inviscid Blade Forces

At the end of the averaging procedure showed previously, the continuity momentum and energy equations take the form:

$$\frac{1}{b} \frac{\partial(b\overline{\rho^e}^t \tilde{V}_j^{e,t,c})}{\partial x_j} = 0 \quad (2.24)$$

$$\frac{1}{b} \frac{\partial(b\overline{\rho^e}^t \tilde{V}_{ij}^{e,t,c} + p\delta_{ij})}{\partial x_j} = \frac{1}{b} \frac{\partial(b\overline{\tau_{ij}}^e \tilde{V}_j^{e,t,c} - \overline{\rho V'_i V'_j}^e - \overline{\rho^e V''_i V''_j}^t - \overline{\rho^e t V'''_i V'''_j}^c)}{\partial x_i} + f_b^j + f_v^j \quad (2.25)$$

$$\begin{aligned} \frac{1}{b} \frac{\partial(b\overline{\rho^e}^t \tilde{V}_i^{e,t,c} \tilde{H}_i^{e,t,c})}{\partial x_i} &= \frac{1}{b} \frac{\partial(b\overline{\tau_{ij}}^e \tilde{V}_j^{e,t,c} - \overline{q^e}^t + \overline{\tau'_{ij} V'_j}^e + \overline{\tau''_{ij} V''_j}^t + \overline{\tau'''_{ij} V'''_j}^c)}{\partial x_i} + \\ &\quad - \frac{1}{b} \frac{\partial b(\overline{\rho V'_i H'}^e + \overline{\rho^e V''_i H''}^t - \overline{\rho^e t V'''_i H'''}^c)}{\partial x_i} + e_b + e_v \end{aligned} \quad (2.26)$$

For sake of completeness we refer the reader to the work of Adamczyk and J.F. Simon where he can find the whole set of equations developed in cylindrical coordinates.

The set of equations obtained applying to Adamczyk procedure have been developed rigorously from the time dependent Navier Stokes equations without any other assumption. The equations are expressed in terms of the averaged conservative variables and describe the mean axisymmetric flow in a single or multistage turbomachinery. However the circumferential average equations don't contain enough information to determine the solution. In order to close this system one must construct a mathematical model for the additional terms that appear in the equations:

- the total stress tensor $R_{ij} = -\overline{\rho V'_i V'_j}^e - \overline{\rho^e V''_i V''_j}^t - \overline{\rho^e t V'''_i V'''_j}^c$ this is probably the most challenging term to model, it contains the Reynolds stresses, the deterministic and the circumferential stresses.
- the inviscid blade forces: they describe the effects of the potential pressure field acting on the blade walls.
- the viscous blade forces: they reproduce the effect of the losses mechanisms acting in real machines; such as boundary layer, end wall losses and 3D losses.
- additional terms in the energy equation such as product between velocity fluctuation and total enthalpy or velocity fluctuation and stresses tensor.

Developing a closure model for all of these terms represents a tremendous task and it is a target far beyond the purpose of this work. In this study we will make an attempt to develop a simple model to describe this extra terms. Moreover we will introduce a simple transport model to fetch the deterministic stresses and in particular the effects of the rotor wake.

2.2.1 Blade blockage contribution

The first source term to add to the axisymmetric Navier Stokes equation is the term due to contraction of the flow path through the row. It has to restore the effect of the dependency from θ coordinate of the geometry that has been erased by the circumferential averaging. It assumes the following form:

$$\mathbf{S}_v = \begin{bmatrix} 0 \\ \frac{p}{b} \frac{\partial p}{\partial r} \\ 0 \\ \frac{p}{b} \frac{\partial p}{\partial z} \\ 0 \end{bmatrix} \quad (2.27)$$

In order to implement this features in OpenFOAM we compose a patchwork of functions according with the expression 2.28

$$b(z) = 1 - f(z) \quad (2.28)$$

with $f(z) \sim b_0 e^{-(\frac{z-z_0}{\sigma})^c}$ where b_0 z_0 σ c are constants defined by the user according to the geometry of the machine.

In particular we define three Gaussian like functions the first from the inlet of the domain to the throat of the stator the second from the throat of the stator to the throat of the rotor and the last one to the throat of the rotor to the outlet of the domain. We decide to use the gaussian like functions for their smoothness, it is important to stress out that in equation 2.27 the first derivative of the blockage factor appears so it is important that the function that defines the blockage factor belongs to the C^1 class in order to not generate discontinuities in the source terms and consequently nonphysical phenomena like the generation of mass. Obviously the definition of the blockage factor as a patchwork of Gaussian functions is not the only possible choice, an other smart one could be to implement it with parametric curves.

As example of the definition of the blockage factor we report an axial distribution of it along with the z coordinate.

In figure 2.1 we plot the definition of the blockage factor along with the axial coordinate. The four vertical lines are the LE and the TE respectively of the stator and of the rotor. We can observe that the blockage factor is equal to 1 outside the blade row and assumes a value between 0 and 1 inside the blade row. In order to mathematically describe its physical meaning it is important that the blockage factor goes rapidly to zero crossing the TE of the blade.

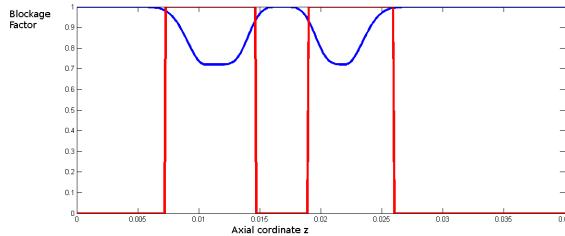
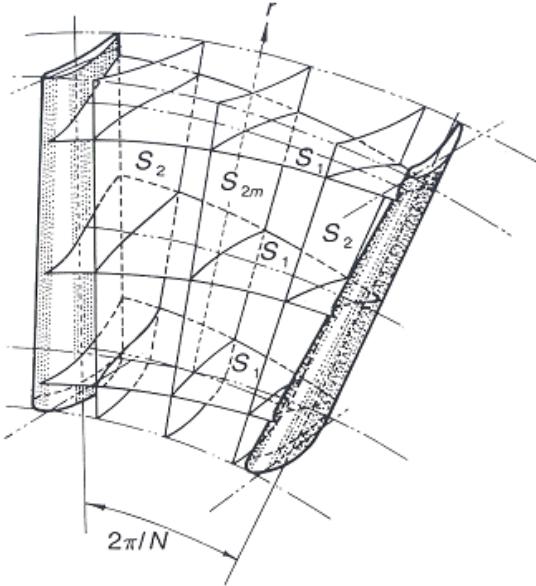


Figure 2.1: Axial distribution of the blockage factor

2.2.2 Inviscid blade forces

This component of the forces is responsible to the flow deflection across the blades row, the inviscid blade forces are due to the pressure field acting on the blade surfaces. They deflect the flow without generating losses, for this reason the inviscid blade forces have to be normal to the streamlines. To describe mathematically this condition we introduce S_1 and S_2 surface Fig. 2.2, where the blade-to-blade S_1 are the revolutionary streamsurface surface for the z axis while the hub-to-tip S_2 surface are obtained by a rigid rotation of a mean surface along the same axis.

Figure 2.2: S_1 and S_2 surfaces

we use this definition

$$(V_i - \omega_i r e_i) \frac{\partial S_2}{\partial x_i} = 0 \quad (2.29)$$

The inviscid blade forces act normal to the streamlines so we have to impose

the parallelism between the normal vector of S_2 and them.

$$\epsilon_{ijk} f_j \frac{\partial S_2}{\partial x_k} = 0 \quad (2.30)$$

If the row presents lean angle Fig. 2.2 we have to introduce an auxiliary relationship between the radial and the circumferential component of the inviscid blade forces:

$$\frac{f_{br}}{f_{b\theta}} = \tan(\lambda) \quad (2.31)$$

where λ is the angle between the normal to the blade camber surface and the θ axis.

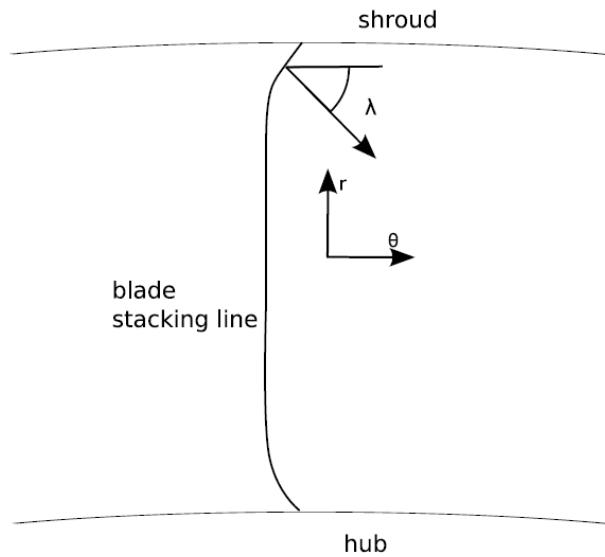


Figure 2.3: lean angle

In the analysis formulation where the metal angle is known inside the bladed region we can calculate the circumferential component of the inviscid blade forces applying to the angular momentum conservation law as:

$$f_{b\theta} = \frac{\rho V_z}{r} \frac{\partial r V_z}{\partial m} \quad (2.32)$$

where m is the meridional direction. Imposing that the inviscid blade forces is always normal to the streamlines we can calculate every component of the inviscid blade forces vector source term. In addition to the three components of the force we must add the work done or extracted by the blade on the flow in the energy equation.

$$\mathbf{S}_b = \begin{bmatrix} 0 \\ f_{br} \\ f_{b\theta} \\ f_{bz} \\ f_{b\theta\omega r} \end{bmatrix} \quad (2.33)$$

2.2.3 Leading edge treatment

In section 2.2.3 the circumferential averaging operator has been introduced and applied on the Navier-Stokes equation. His effect of the balance equations is to eliminate every azimuthal dependency from the flow variables, the drawback is that after its enforcement there are no solid walls due to the blade inside the flow domain. The presence of the walls is restored including in the distributed blade forces in the model. For its axisymmetric nature the throughflow model is not able to describe the prerotation of the flow approaching the leading edge of a blade. If the analysis method is used (the flow angles are imposed) a discontinuity occurs at the leading edge because the flow is not aligned with the camber. It has been demonstrated by Simon that the leading edge discontinuity can generate an unacceptable amount of numerical loss generation and moreover it can lead to a discontinuous distributions of blade forces that can cause numerical instability and difficulties to reach the convergence. In order to avoid these problems we follow the approach proposed by Rosa Taddei(2008), the idea is to modify the blade camber surface in order to link softly the last nodes of the domain where there are no blades with the first node of the block containing the blade. In this work the camber angle has been described by functions in the form:

$$f(z) = A + Be^{-((x-\mu)/\sigma)^c} \quad (2.34)$$

where A and B are real number introduced in order to give the blade the right inlet and outlet angles, μ , σ and c describe the rapidity of variation of the flow.

2.2.4 Shock capturing properties

One of the major advantages of the Navier-Stokes/Euler throughflow against the Streamline Curvature Method is that in the first there is more physics than in the latter. For this reason there is no need of introducing a model in order to describe the loss generated by the shock inside the blade passage. A very interesting feature of the shock-capturing properties of the Navier-Stokes/Euler throughflow is established by Simon(2007) and it states that these types of models are able to capture shock when the relative Mach number is superior than one. Moreover the intensity of the captured shock is equal to the intensity of a normal shock in blade to blade plane for the same upstream Mach number. This means that the throughflow model is able to catch only the normal shock to the stream wise direction. The shock capturing properties of the Euler/Navier-Stokes throughflow allows these models to predict the chocking massflow in a very natural way without

external empirical assumptions or the need of different solution loops as it happens in the Streamline Curvature solvers.

2.3 Viscous Blade Forces

Due to the axisymmetric description of the turbomachinery carried on in this work there is no solid wall representing the blade but only distributed forces. The effect of the viscous forces of the flow field must be restored including in the model a distribution of viscous losses.

2.3.1 Distributed loss model

The aim of distributed loss model is to describe only the profile losses due to the development of the boundary layer on the blade surfaces. The most suitable model for this purpose is the distributed losses model proposed by Hirsch and Stollenwerk (2000) and applied by Rosa Taddei (2008); in this model the viscous forces only product entropy inside the flow field without performing work on fluid, in this way the distributed loss model is able to describe the losses generated from the boundary layers. In order to describe these losses it is convenient to introduce one more equation describing the entropy transport and generation inside the fluid domain.

$$\frac{\partial(\rho S)}{\partial t} + \frac{\partial(\rho V_i S)}{\partial x_i} = \frac{1}{T} (\tau_{ij} \frac{\partial V_i}{\partial x_j} - \frac{\partial q_i}{\partial x_i}) + \sum_{m=0}^M \dot{S} \quad (2.35)$$

where the variable are obviously Favre/Reynolds averaged and the way to introduce the effects of the deterministic stresses will be explained later, M is the number of entropy sources introduced in the model. From the field of entropy we can calculate the forces using the equation of entropy itself:

$$f_v = -\rho T \frac{V_i \partial_i S}{V} \quad (2.36)$$

where V is the absolute value of the velocity field $\sqrt{u^2 + v^2 + w^2}$. In the same way we can define the loss in the relative reference frame replacing the absolute velocity with the relative one. It's important underlying that this relation is rigorously valid only for the Euler system and in absence of shear stresses. In the regions of high gradients described by the Navier Stokes equations this relation doesn't describe the physics in the proper way. In the model it will be excluded in this regions. Finally the vector of viscous distributed blade forces can be described as:

$$\mathbf{S}_v = \begin{bmatrix} 0 \\ f_{vr} \\ f_{v\theta} \\ f_{vz} \\ f_{v\theta\omega r} \end{bmatrix} \quad (2.37)$$

2.3.2 Profile losses

The profile losses are generated by a 2D flow, living on the S_1 surfaces. To model these losses we follow the work performed by Denton (1990) and Lampard (2009). In this work the authors link the viscous losses to the entropy generation. For example Denton explains that in a 2D boundary layer the entropy generation can be linked to the other flow variables by:

$$\dot{S} = \frac{\partial}{\partial x} \int_0^\delta \rho V_x s dy \quad (2.38)$$

Introducing empirical correlation this expression can be simplified in

$$\dot{S} = \frac{C_D \rho V^3}{T} \quad (2.39)$$

where the C_D is the Drag coefficient. The value of the drag coefficient very seldom can be computed analytically, fortunately there are a lot of empirical and semi empirical correlations that help to estimate it. In this work we follow the path proposed by Denton (1999) and we use the following expression to compute numerically the value of the drag coefficient:

$$C_D = 0.0056 Re_\theta^{-\frac{1}{6}} \quad (2.40)$$

Where $Re_\theta = \frac{U_{riff} \theta}{\nu}$ is the Reynolds number computed on the momentum thickness and θ is the momentum thickness. In this way it is possible to estimate the entropy production in the boundary layer as :

$$\dot{S} = \sum_0^{N_s} c_{ax} \int_0^1 C_D \frac{\rho V^3}{T} d\left(\frac{x}{c_{ax}}\right) \quad (2.41)$$

where the c_{ax} is the axial chord and N_s is the number of wet surfaces i.e. the sum of suction and pressure side of the blades.

2.3.3 Endwalls Boundary Layer Losses

The Navier-Stokes equations own the capabilities of describing the fluid-wall viscous interaction. The main modelling task in order to describe this kind of losses is the selection of the proper turbulence model. This kind of problem involves a flow in a duct with no (or very low) separation, for these reasons a RANS Spalart-Almaras turbulence model has been chosen. The Spalart Allmaras is a one equations model that describes the evolution of the eddy viscosity. The main advantage of the Spalart Allmaras model compared to the algebraic model is that the turbulent eddy field is always continuous ; the model compared to the two equation models is more robust and has less computational cost.

The first cell has been placed at a distance with the value of $y+$ around 30-40 so the use of a wall function is compulsory. For further details about the turbulence model we refer to Numeca's user guide.

2.3.4 3D Losses

Source terms that correlate the losses with entropy sources are used to model the 2D boundary layer losses. The hub and tip endwall losses are fully described by the N-S equations and with the correct choice of the turbulence model. The last losses we need to model are the 3D losses; this phenomena arises from the interaction between the annulus walls and the blade and cannot be captured in an axisymmetric model. In this work the correlations of Roberts et al. 1986 have been implemented in order to insert the 3-D effects inside the throughflow model. The total losses will be the sum between the 2-D and the 3-D losses. The correlations of Roberts give the maximum value of the loss coefficient and the location of the max along the span, these correlations are given as a function of the main geometrical and fluid dynamic parameters of the stage. With ω we intend the max value of the loss coefficient, with S_{max} the location of the max along the span and finally with S_{rt} the extension of the losses as a fraction of the span (this parameter can be easily interpreted as the variance of the a Gaussian function). We report the correlation for the shroud:

$$\omega_{max} = 0.25 \tanh \sqrt{10^3 \delta^* \overline{TC}} \quad (2.42)$$

$$S_{max} = \frac{\omega_{max}}{2} \quad (2.43)$$

$$S_{rt} = 2.5(S_{max}) \quad (2.44)$$

Where $\overline{\delta^*}$ is the displacement thickness and \overline{TC} the tip clearance as a fraction of the span. Similar expression are given for the correlation at hub:

$$\omega_{max} = 0.2 \tanh \sqrt{15 \frac{\phi \overline{\delta^2}}{AR\sqrt{\sigma}}} \quad (2.45)$$

$$S_{max} = 0.1 \quad (2.46)$$

$$S_{rt} = 2.5 S_{max} \quad (2.47)$$

where AR is the aspect ratio and σ the solidity of the row. However this correlations don't obtain the whole distribution along the span but only some typical point of these distributions. These points have been linked with Gaussian like functions Fig. 2.4, we selected this kind of function because it is quite easy to fix the standard deviation, the maximum value and its radial position; moreover they (Gaussian functions) decay quickly after few standard deviations so the peaks are not influenced by each others. In figure 2.4 on the left the radial distribution of losses is plotted linked with the functions the value calculated with the correlations 2.42-2.47 with a Gaussians like function. In figure 2.4 left the modification of the absolute angle at the outlet of the rotor is plotted, in this work we don't act directly on this parameter imposing the modification of the flow angle but we modify the value of the inviscid blade forces in order to obtain a physically correct

behaviour of the deflection at hub and tip. The basic idea of this procedure is explained in the following subsection 2.3.5.

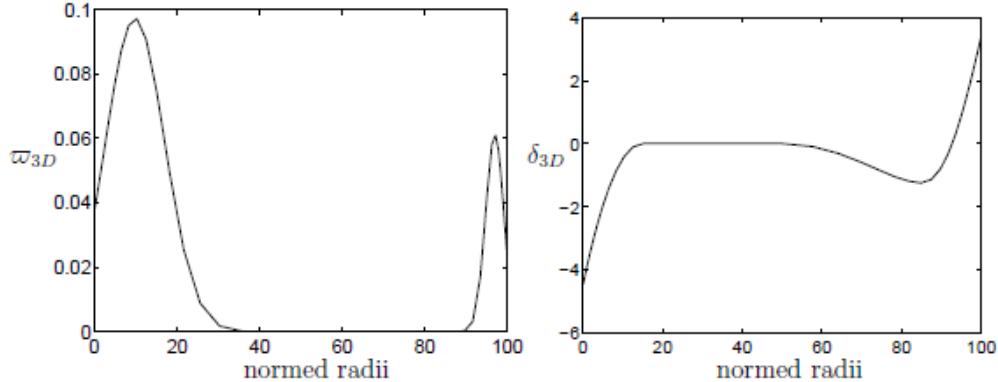


Figure 2.4: Loss coefficient and deviation angle

2.3.5 Effect of the Tip Clearance on the Flow Deflection

The last 3-D effects that have to be introduced manually in an axisymmetric model are the effects related to the tip clearance on the flow deflection. This is a very demanding task since the effects of the tip clearance are strongly 3-D. In his paper Gallimore (1997) proposed to modify the circumferential component of the inviscid blade force, this modification has been proposed in the basis of the experiments performed by Dring(1993). Dring sperimentally observed that the circumferential component of the blade force varies smoothly approaching the endwall, on the other hand the circumferential component of the momentum shows rapid changes Fig 2.5. The physical explanation of this behaviour is that the static pressure field is practically constant across the boundary layer (moving normally to the wall).

After this consideration we can develop a simple model for the evolution of the circumferential inviscid forces at the endwall:

- for the hub region the circumferential component of the blade forces is kept constant for a given region starting from the hub.
- for the tip region the blade force is set to zero in the gap.

In order to understand how this approximation affects the model two viscous simulations are performed the first with the 3-D correlation of Roberts only, the second with the 3-D correlation of Roberts and the blade force modification.

According to this model the blade force modification allows to describe in a more accurate way the tip flow, the deflection now is very similar to the correlation provided by Roberts for the deflection Fig. 2.4 and 2.6.

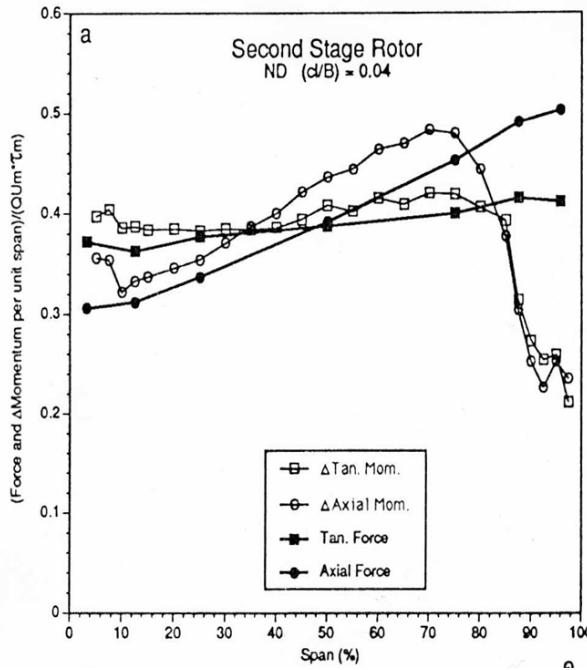


Figure 2.5: Radial distribution of momentum and forces

2.4 Deterministic and circumferential stresses

The last task in order to close the throughflow model is the modelling of the deterministic and circumferential stresses. This extra terms arise from the last two averaging operators of the Adamczyk's procedure (introduced at the beginning of this chapter). The aim of these terms is to describe the effects of the unsteady deterministic fluctuations and the non-axisymmetric fluctuations on the steady and axisymmetric flow field described by the throughflow model. Develop a complete and reliable closure model for this terms represent a very demanding task and it is far beyond the scope of this work. However a lot of researchers have developed, during the years, a lot of closure models that can be divided mainly in three categories:

- The first category is called LDST (Lump deterministics source term), in this approach the researchers (Galbraith and Orkwis 2008) propose to use an equation analogous to the Reynolds stress equations to describe the evolutions of the determinist stresses.
- Another approach has been proposed by Charbonnier (2004). In his work he develops a set of six transport equations (one for every component of the stress tensor). This technique is supposed to be quite accurate, however it is computational expensive and it still relies on empirical datas or more accurate CFD simulations.

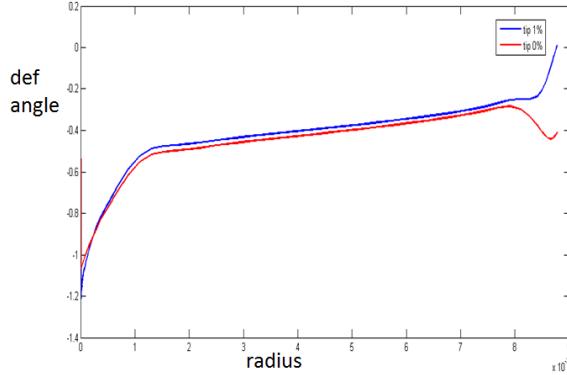


Figure 2.6: Deflection with blade force modification (blue) and without (red)

- The last approach is the one proposed by Stollenwerk and Kugeler (2004), they make a Boussineq's like hypothesis to link the deterministic stress tensor to the mean field deformation rate by the introduction of an auxiliary scalar field called deterministic viscosity. Moreover he proposes a single scalar equation in order to describe the evolution of the deterministic viscosity inside the flow field.

In this work a first approach of the deterministic stresses modelling has been performed choosing the last approach which can be a good compromise between physical accuracy and computational costs. The basic idea proposed by Stollenwerk is to correlate the additional terms due to the unsteady deterministic part of the flow to the deformation of the mean flow field by the introduction of the deterministic viscosity.

$$\tau_{ij}^d = \overline{\rho u_i'' u_j''} = \mu_d \left(\frac{\partial x_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.48)$$

Since the strain rate deformation is known from a steady simulation, only μ_d has to be modelled in order to compute the deterministic stresses. The evolution of the deterministic viscosity can be described by a scalar transport equation.

$$\bar{\rho} \frac{\partial \mu_d}{\partial t} + \bar{\rho} \tilde{u}_j \frac{\partial \mu_d}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\mu_{tot} \frac{\partial \mu_d}{\partial x_j} \right) + S_{\mu_d} \quad (2.49)$$

where μ_{tot} is the sum of molecular, turbulent and deterministic viscosity and S_{μ_d} is a source term introduced to describe the production/destruction of the deterministic viscosity. The source term is the product between of the streamline curvature and an axial distribution.

$$S_{\mu_d} = f(\chi) e^{-\frac{2\pi}{L} \sqrt{1-(\frac{\omega r}{a})^2} (z-z_{LE})} \quad (2.50)$$

This source term has the only aim to reproduce the potential effects of the interaction between blades and wakes. It has been computed exploiting the assumption that the total pressure downstream to the rotor has a wave like form

$e^{ik(x-ct)}$ and that the wake spread downstream according the non dispersive wave equation. Further details of this calculation can be found in the work of Stollenwerk (2000) and in appendix C. It is important to stress that all quantities involved in the modelling of the source term are known from a steady state simulation. This additional equation has to be numerically solved in order to close the system.

Chapter 3

Numerical Tools

In the previous chapter the physical and mathematical model and its assumptions have been presented. The purpose of this chapter is to introduce the numerical technique exploits in order to solve the mathematical model. The throughflow model is based both on the Navier-Stokes and Euler equations so for sake of completeness will be presented the major features of the solver: inviscid and viscous flux and time stepping techniques.

3.1 Space discretization

The governing equations presented in the previous chapter has been discretized on an unstructured hexaedral mesh create by HEXPRESS. Nowadays a lot of numerical techniques are available in order to solve the balance equations, such as finite elements finite differences and finite volumes. In this work the finite volumes technique has been selected for its ability to handle with discontinuous flow features such as shock waves. The governing equations can be rearranged in the following form:

$$\frac{\partial U}{\partial t} + \frac{\partial(F - F_v)}{\partial z} + \frac{\partial(G - G_v)}{\partial r} = S \quad (3.1)$$

where U is the vector of the conservative variables F F_v the inviscid and viscous flux along the axial direction, G G_v the inviscid and viscous flux along the radial direction and S is the vector that include all the source terms.

The finite volume techniques involve the integration of the balance equations on an elementary control volume. This yields to:

$$\int_V \frac{\partial U}{\partial t} dV + \int_V \left(\frac{\partial(F - F_v)}{\partial z} + \frac{\partial(G - G_v)}{\partial r} \right) dV = \int_V S dV \quad (3.2)$$

Thanks to the Gauss-Green theorem we can rearrange this expression as:

$$\int_V \frac{\partial U}{\partial t} dV + \int_{\partial V} ((F - F_v)n_z + (G - G_v)n_r) d\Sigma = \int_V S dV \quad (3.3)$$

The convective and diffusive flux can be computed as a summation of the contributions over all discrete faces bounding the elementary control volume. This assumption leads to:

$$\frac{\partial U}{\partial t} = \frac{1}{V} \sum_{j=1}^{N_j} E_j \Sigma_j + S \quad (3.4)$$

where E is a symbolic notation to indicate the totality of flux across the boundary faces.

3.1.1 Flow domain

One of the main advantages of an axisymmetric throughflow analysis is the simple computational geometry that leads to a low number of cells and a low computational effort. The 2-D domain discretization of a turbine or a compressor stage has a very simple geometrical feature, that allows us to discretize it with a single-block mesh.

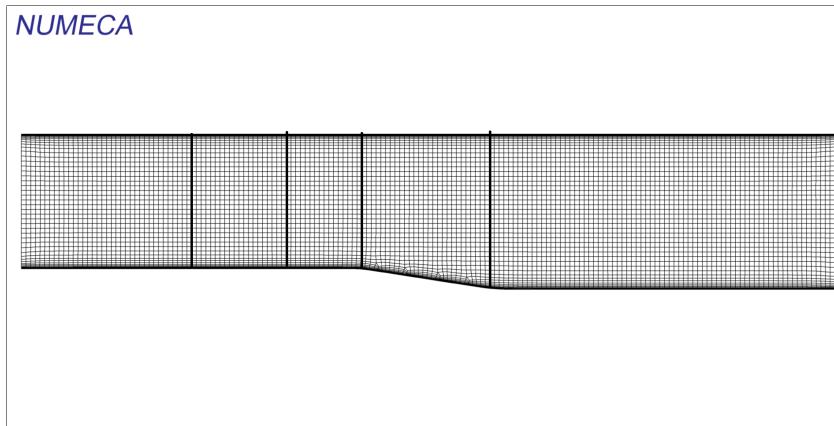


Figure 3.1: Example of viscous mesh

The mesh Fig 3.1 to solve the throughflow model is quite simple and general; the only thing the user has to care about is the boundary layer to the endwalls. This has to be inserted if the N-S equations are resolved instead of the Euler equations. For the next simulations a wall function approach has been chosen with a $y+$ value around 30 at the wall Fig 3.2. Generating the grid we followed two main guidelines in order to obtain a good convergence with reliable results:

- The first guideline is the choosing of the axial spacing that allows a good resolution in proximity to the LE of the blades. This zone of the mesh is quite complicated as explained in section 2.2.3
- The second guideline is about the radial spacing. Apparently there are no constraints on radial spacing however it is quite an important parameter. In case of Euler inviscid simulation it is important not to select a radial spacing too different from the axial one in order to avoid non homogeneous cells. In case

of a Navier Stokes simulation with boundary layer two aspects are important: the first is to choose radial spacing in order to avoid nonhomogeneous cells the second is to choose the radial spacing compatible with the aspect ratio of the boundary layer, in other words the user must avoid to juxtapose very thin cells (the ones of the boundary layer) with the bigger ones outside the boundary layer.

A grid sensitivity analysis has been performed showing that the optimal amount of cell is: around five thousand for the viscous calculation and around three thousand for the inviscid one. The time needed to reach the convergence is in the order of minutes.

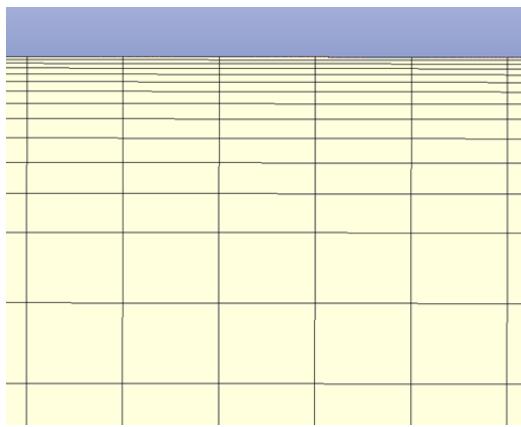


Figure 3.2: Boundary layer

Because of the axisymmetric approximation there is no solid wall that represents the blade wall; for this reason we need to store in the centroid of the cells some parametres that describe the blade geometry such as deflection flow angle and the value of blockage factor. To obtain these quantities a linear interpolation from the cell center is performed.

3.1.2 Inviscid Flux

For the computation of the inviscid flux across the boundary faces a FDS (Flux Difference Splitting) has been chosen. The FDS schemes calculate the numerical fluxes by computing an approximate solution of the Riemann problem at the interface. The basic idea of the Roe's schemes is to calculate the flux across the interface linearizing the Jacobian matrix of the system at every cell interfaces. Let $F(U)$ be the flux's vector along a prescribed direction and U_L and U_R the value of the conservative variable vector in the left and right cell of any cell interface, the Jacobian matrix can be defined as follows:

$$A(U) = \frac{\partial F(U)}{\partial U} \quad (3.5)$$

The basic idea is to substitute at the cell interface the real Jacobian matrix with an approximate expression $\hat{A}(U_L, U_R)$ that satisfies the following properties.

- the approximate matrix \hat{A} maintains the hyperbolic nature of the original system, id est it has real eigenvalues and a complete set of eigenvectors
- \hat{A} is consistent with the original matrix A when $U_R, U_L \rightarrow U$
- The scheme is conservative for every U_R and U_L , id est $F_R - F_L = \hat{A}(U_R - U_L)$

Let the eigenvector and eigenvalue of $\hat{A}(U_R, U_L)$ be e_k and λ_k we can project the variable jump across the interface as:

$$U_R - U_L = \sum_k \alpha_k e_k \quad (3.6)$$

$$F_R - F_L = \sum_k \alpha_k \lambda_k e_k \quad (3.7)$$

where the α_k is the wave strength across the interface. Considering the interface between two cells $i + \frac{1}{2}$ the flux at the interface can be computed by the summation on the negative and positive wave speed.

$$F_{i+\frac{1}{2}} = F_R - \sum_{k=1}^{\lambda_k > 0} \alpha_k \lambda_k e_k \quad (3.8)$$

$$F_{i+\frac{1}{2}} = F_L + \sum_{k=1}^{\lambda_k < 0} \alpha_k \lambda_k e_k \quad (3.9)$$

by taking the arithmetic mean of the previous expressions finally we obtain the expression for the inviscid flux.

$$F_{i+\frac{1}{2}} = \frac{(F_R + F_L)}{2} - \sum_{k=1}^m \alpha_k \lambda_k e_k \quad (3.10)$$

where m is the number of the eigenvalues. The whole argument has been presented for a 1-D mesh but it can be easily extended to more dimensions calculating each independent spatial direction using the 1-D method.

The Roe's scheme has a very low amount of numerical dissipation and can be considered a non diffusive scheme for grid aligned flows, Kerman and Plet [2001]. However the scheme can exhibit some unphysical solutions as expansion shock, the reasons of this behaviour is well explained by E.G. Plett and M.J. Kermani and it is basically due to the fact that one of the eigenvalue (one of the acoustic one) goes to zero in the zones of the flow field in which the Mach number approach to one.. To avoid this behaviour an entropy fix must be introduced in the scheme, a lot of effort has been devoted to solve this problem, a good solution has been proposed by Harten and Hyman [1983]. They observed that the unphysical solution is due to the local vanishing of the numerical dissipation (due to the vanishing of one of the eigenvalue) so they proposed to substitute the small value of the numerical dissipation with a larger value.

3.1.3 Viscous Flux

The calculation of the viscous fluxes requires the knowledge of the value of the variables and of the value of the gradient of the variables on the boundary of the cells.

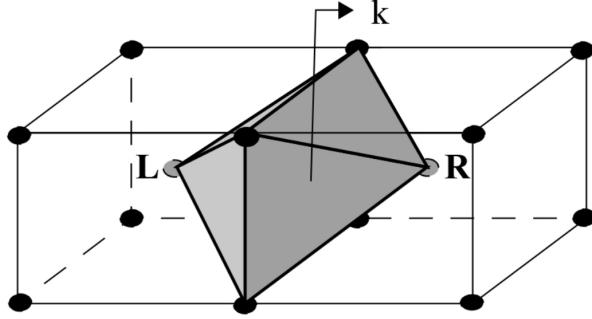


Figure 3.3: Diamond control volume approach for viscous computation

The viscous fluxes are calculated in a purely central way, in order to compute gradients on the cell faces the Gauss Green theorem to the so called diamond path. The diamond path Fig 3.3 is a control volume joining the vertexes of an edge and the centroid of the cells sharing that edge. The value of the variables in the face vertexes is calculated by linear interpolation of the neighbour cells.

3.2 Time Discretization

At the beginning of this chapter the throughflow equations have been introduced, it is important to notice that they are solved under their unsteady form despite the fact that only the steady solution is of interest. This method is the so called time marching method and has been developed in order to avoid the difficulties linked to the fact that the equations change their mathematical behaviour with the flow regime. In this work a 4 stages explicit Runge Kutta scheme has been used, the more general for a multistage method for the equation:

$$\frac{\partial U}{\partial t} = F(U) \quad (3.11)$$

$$U^1 = U^n + \alpha_1 \Delta t F(U^N)$$

$$U^2 = U^n + \alpha_2 \Delta t F(U^1)$$

$$U^q = U^n + \alpha_q \Delta t F(U^{q-1})$$

$$U^{n+1} = U^q$$

The coefficients α_i determine the stability area. Cause the explicit method is conditionally stable the temporal and spatial parameters must satisfy the following relation.

$$\left| \lambda_{max} \frac{\Delta t}{\Delta x} \right| < \nu_{max} \quad (3.12)$$

Where the ν_{max} is the maximum allowable Courant's number (function of the parameters of the scheme) and λ_{max} is the max eigenvalues of the Jacobian matrix computed in section 3.1.1.

3.3 Boundary conditions

In order to close mathematically the problem we need the right boundary conditions, the number and the position of the boundary conditions are defined by the mathematical nature of the equations. The patch that needs boundary conditions are the endwalls, the inlet and the outlet:

- The boundary conditions on the wall are $\vec{V} \cdot \vec{n} = 0$ for the Euler Wall and $\vec{V} = V_{wall}$ for the N-S wall. The wall are adiabatic so $\nabla \cdot \vec{n}$, the other variables such as $S_g \mu_t \mu_d$ are set equal to zero. To make the simulation less computational expensive a wall function treatment of the wall is chosen.

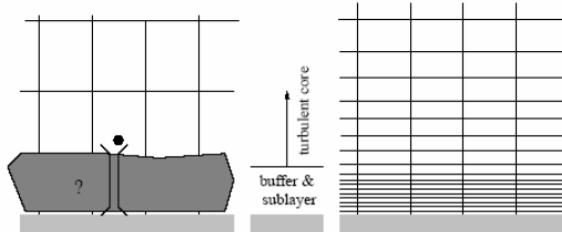


Figure 3.4: Conceptual illustration of wall function (left) and near wall treatment(right)

- To set up the boundary conditions of the inlet and outlet, the inviscid nature of the flow prevails on the viscous effects. For the Euler's equations the number of boundary conditions on the boundary is equal to the number of incoming characteristic curves.
- Inlet: for the subsonic inlet we have seven incoming characteristic so the users have to give seven information at the inlet: $P_T \ T_T \ atan(\frac{V_z}{v_{theta}}) \ atan(\frac{V_r}{V_z}) \ \mu_t \ \mu_d \ S_g$
- Outlet: for the subsonic outlet there is only one incoming characteristic so only one information has to be given at the outlet. At the outlet the static pressure is prescribed. However the distribution of the static pressure along the outlet boundary patch cannot be chosen in an arbitrary

way. The right physical distribution of the static pressure can be obtained thanks to the momentum equation projected at the outlet along the radial direction.

$$\frac{\partial p}{\partial r} = \frac{\rho u_\theta^2}{r} - \frac{\rho u_r^2}{2} - \frac{\partial \rho u_r^2}{\partial r} - \frac{\partial \rho u_x u_r}{\partial x} \quad (3.13)$$

Assuming the radial component of the velocity negligible respect to the axial and to the tangential one, finally we get the expression:

$$\frac{\partial p}{\partial r} = \frac{\rho u_\theta^2}{r} \quad (3.14)$$

This is the so called static pressure with radial equilibrium boundary condition and it is one of the most widespread outlet boundary condition for the analysis of the axial turbomachinery.

Chapter 4

Openlabs

In this chapter the numerical implementation of the mathematical model introduced in the second chapter will be exploited. All the implementation has been performed with the help of Numeca FINE/Open with Openlabs; particularly the mesh, boundary conditions, numerical schemes, turbulence modelling and convergence criteria have been chosen in the main environment of Numeca FINE/Open. Then the more particular features of the axisymmetric throughflow model such as: blade forces, flow angles definition, and new transport model are introduced to describe entropy generation and deterministic stresses have been added to the solver thanks to Openlabs.

4.1 Openlabs Environment

Openlabs is a new module of the FINE/Open environment that allows the user to modify a physical model already implemented or add a new physical model defined by the user. Basically Openlabs gives the user the possibility to access the physical model of FINE/Open solver.

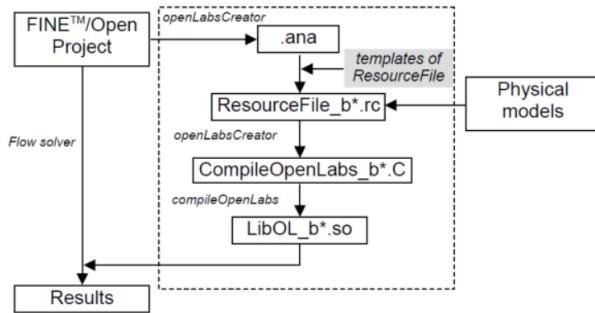


Figure 4.1: Openlabs' illustration

In order to run a simulation with a customized physical model the following steps (shown in Fig 4.1) are needed:

- The first step is the creation of a project: first create the mesh in HEXPRESS then set the boundary conditions, numerical schemes, turbulence model fluid properties.
- When the project is ready , the user can activate Openlabs. The first step is to analyze the existing project in order to see CFD variables, algebraic and differential expressions and boundary conditions
- Once analyzed the exiting project the user can write his own piece of solver in the resource file thanks a very intuitive metalanguage; after compiling the ".rc" file is transformed in a *C++* file, the *C++* file is compiled to generate a dynamic library.
- After the generation of the dynamic library the solver can be launched as usual.

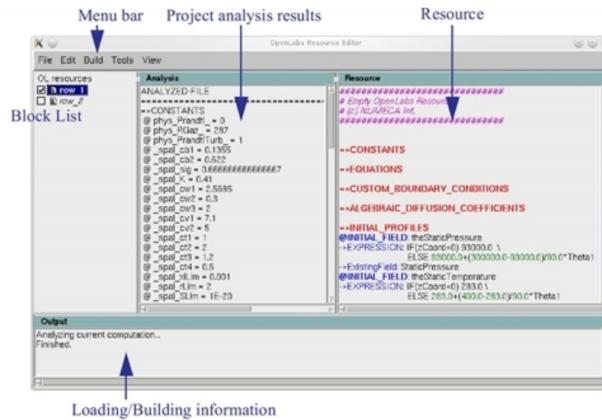


Figure 4.2: Openlabs' editor GUI

The Resource manager's GUI Fig 4.2 is divided in four main windows :

- Block list : Openlabs is block dependent, the user has to activate the block in which he wants to customize the solver.
- Project analysis results : thanks to the project analysis results the user can analyze the actual setting of FINE/Open solver.
- Resource: the user can add or modify the existing project; it is divided in eight sections:
 - Constant: here every constant properties of the project has to be declared
 - Equations : in this section the user has to enter the equation that he wants to add to the solver, here both algebraic and differential equation can be added.

- Custom boundary conditions : in this section the user can modify existing boundary conditions and insert boundary conditions for the new equations.
- Algebraic diffusion coefficients : this section helps to modify or add new equations to compute the value of the thermophysical properties
- Initial profiles : this sections allows the user to modify or create new initial condition for the transport equations.
- Source Terms : this section allows the user to add source terms to the existing PDE.
- Aux terms and additional information : the user can write big mathematical expressions in smaller pieces, the expressions written here are computed but not shown in the post processing.
- Building information: this section shows the compiling information.

So far the main Openlabs features have been explained, later in the chapter we will go deeper in the details of the implementation.

4.2 Throughflow implementation in Openlabs

As explained in the second chapter the axisymmetric analysis of turbomachinery allows to perform a fast and quite accurate analysis of the flow field inside a turbomachinery stage. The low computational cost of a throughflow analysis is due to the assumption done in describing the flow, the most important are: steady flow and axisymmetric flow. This assumption leads the introduction of extra terms that don't appear in the original Navier-Stokes equations. More in detail after the Adamczyk averaging procedure the governing equation have this form:

$$\frac{\partial \rho u_i}{\partial x_i} = S_b \quad (4.1)$$

$$\frac{\partial(\rho u_i u_j + p \delta_{ij})}{\partial x_j} = \frac{\partial(\tau_{ij} + \tau_{ij}^t + \tau_{ij}^d)}{\partial x_j} + f_{vi} + f_i + S_{bi} \quad (4.2)$$

$$\frac{\partial H u_i}{\partial x_i} = \frac{\partial(\tau_{ij} + \tau_{ij}^t + \tau_{ij}^d) u_j}{\partial x_i} + \frac{\partial(q_i + q_i^t + q_i^d)}{\partial x_i} + e_v + e_{nv} + S_b \quad (4.3)$$

$$\frac{\partial S_g u_j}{\partial x_j} = \frac{1}{T} (\tau_{ij}^{tot} \frac{\partial u_i}{\partial x_j} + \frac{\partial q_j}{\partial x_j}) + \sum_{n=1}^N \dot{S}_{source} \quad (4.4)$$

$$\frac{\partial \mu_d u_j}{\partial x_j} = \frac{\partial}{\partial x_j} (\mu_{tot} \frac{\partial \mu_d}{\partial x_j}) + S_{\mu_d} \quad (4.5)$$

In the overlying system we can recognize the classic Navier-Stokes equation and more extra terms with the aim to describe: the blockage factor S_{bi} , the inviscid $f_i e_i$ and viscous blade forces $f_{vi} e_{vi}$, the turbulent and deterministic stresses clustered in

μ_{tot} . The extra terms that need to be implemented are both algebraic or differential expressions and transport equations. This two terms need to be implemented with a different syntax and for this reason they will be explained in two different sections.

4.2.1 Constant Propertie Definitions

The first step of the throughflow implementation in Openlabs is the definition of the constant. That has to be done in the CONSTANT section of the Resource manager. In the troughflow model the constant information are: the axial position of the stator and rotor block, the deflection angles and the parameters that are necessary to define blade angles. The syntax is the following:

```

1 #rotor and stator blocks
2 @z_Sini=0.073
3 @z_Sfin=0.11
4 @z_Rini=0.1289
5 @z_Rfin=0.173
6
7 #deflection angles
8 @DS=72
9 @DR=110

```

4.2.2 Equations

In this section the user can define the additional equations that wants to add to the solver. In this section has been implemented :

- Viscous and inviscid blades forces according to the definition given in the chapter 2.
- The additional transport equations for the evolution of the entropy and the deterministic viscosity.
- The function that defines the absolute and relative flow angle and the distribution of the blockage factor inside the mesh.

For sake of completeness we report below the implementation of the inviscid blade forces for the stator in Openlabs Language.

```

1 #inviscid blade forces
2
3 @NONDIFFEXPRESSION:ThetaBladeForceStat
4 ->EXPRESSION:F_theta= \\
5 Density*Vz/rCoord*GRAD(rCoord*Vz*tan(alpha),z)
6
7 @NONDIFFEXPRESSION:AxialBlabeForceStat
8 ->EXPRESSION:F_axial= - F_theta*tan(alpha)

```

The imposed flow angle $\alpha(z, r) = \text{atan}(\frac{V_\theta}{V_z})$ is defined as a patchwork of some Gaussian functions pasted in such a way to make the resulting function as smooth as possible. In order to implement in the model the blade angle a function $\text{alpha}(z, r)$ has been built in this way:

$$\alpha(z, r) = \alpha_{mid}(z)(1 + \tau(r)\epsilon(z)) \quad (4.6)$$

Where $\alpha_{mid}(z)$ is the blade angle at the mid line, $\tau(r)$ is the twist of the blade and $\epsilon(r, z)$ is an auxiliary function that allows the swirl to vary along the axial direction. In the model is also implemented the lean angle that is treated separately. It is important to stress out that the whole throughflow is implemented in analysis mode; this means that the geometry is known and the performances are calculated. In this implementation the inviscid blade forces are calculated considering the informations of the flow angle then the inviscid blade forces are inserted in the momentum equations and the flow deflection are computed. Exactly the same considerations are true for the relative flow angle β

```

1 @NODIFFEXPRESSION:RelativeFlowAngle
2 ->Expression:beta0=IF(zCoord>cr ANDzCoord<z_Rfin)
3 ((DR*pi/180*exp(-pow(((zCoord-cr)/krot),qr))
4 -(R0*pi/180))*(1-rs*(16.45*rCoord-5.8291)*(-22.7*zCoord+3.927)))
5 ELSE 0.0

```

The most powerful feature of Openlabs is probably the implementations of new PDEs. Thanks to Openlabs the customization of the solver by adding new PDEs is straightforward. The basic idea is to exploit the "standard form" of the transport equations. The more general transport equation can be written as a sum of four terms:

- Local time derivative $\frac{\partial(\rho\bullet)}{\partial t}$
- Convective term $\vec{\nabla} \cdot (\rho \vec{U} \bullet)$
- Diffusion term $\vec{\nabla} \cdot (\Gamma \vec{\nabla} \bullet)$
- Source Terms

An example of the implementation of the transport equation for the deterministic viscosity is presented. First has been reported the mathematical formulation of the problem:

$$\begin{cases} \frac{\partial \rho \mu_d}{\partial t} + \frac{\partial \rho u_j \mu_d}{\partial x_j} = \frac{\partial}{\partial x_j} (\mu_{tot} \frac{\partial \mu_d}{\partial x_j}) + S_{u_d} \\ \mu_d(r_{wall}, z, t) = 0 \\ \mu_d(r, z_{in}, t) = \mu_{d,inlet} \\ \mu_d(r, z, 0) = \mu_{d,init} \end{cases} \quad (4.7)$$

This differential problem in Openlabs language is written as:

```

1
2 @PDE:Deterministic_Stresses_Transport
3 ->EXPRESSION:DDT(muD)+CONV(muD)=
4 DIFF(muD_diffcoeff)+
5 SOURCE(muD_source)
6 ->Convective_Scheme:UpwindHighOrderConervative
7 ->UseCustomBC:muDinlet,1
8 ->UseCustomBC:muDwall,0,2
9 ->InizializeTo:muDinit

```

With the keyword \rightarrow *EXPRESSION* we can identify the three differential operator introduced previously and the source terms. At first we have to define the diffusion coefficient for the diffusion term, this one can be defined in the *@ALGDIFFCOEFF* section:

```

1
2 @ALGDIFFCOEFF:muD_diffcoeff
3 ->EXPRESSION:Viscosity+EddyVisc+muD

```

The source terms can be implemented in the *@SOURCE* section:

```

1
2 @SOURCE:muD_source
3 ->EXPRESSION:SmuD_Source

```

where *SmuDSource* is the function described in section 2.4, its analytical expression can be defined both in *@EQUATION* section or in *@AUXTERM* section.

The field \rightarrow *ConvectiveScheme* allows to choose the spatial discretization scheme while the temporal one is a Runge-Kutta by default. In \rightarrow *UseCustomBC* we define the value or expression of the boundary conditions and the patch (in this case the INLET has PI 1 while WALLS is PI 0,2), the boundary conditions expression is defined in the section *@CUSTOMIZEBOUNDARYCONDITIONS* as follows:

```

1
2 @CUSTOMIZED_BOUNDARY_CONDITION:muDinlet
3 ->EXPRESSION:0

```

The same parametres apply to the wall boundary conditions while the initial conditions must be defined ,exactly in the same way,in the *@INITIALPROFILE* section.

Another important feature of the section *@ALGDIFFCOEFF* is that the user can also modifie the diffusion coefficient of every equation of the system; this is particularly important because in this model a Boussinesq's like hypothesis has been applied to model turbulence and deterministic stresses. Once the eddies viscosity equation and the deterministic viscosity equation have been solved we must add deterministic and turbulent viscosity to molecular viscosity in order to obtain the effective viscosity and the effective conductivity, this can be done with the instructions:

```

1
2 @ALGDIFFCOEFF : EffectiveViscosity
3 -> EXPRESSION (MomentumXEquation) : Viscosity+EddyVisc+muD
4 @ALGDIFFCOEFF : EffectiveConductivity
5 -> EXPRESSION (EnergyEquation) : Conductivity +
6 (Cp+EddyVisc)/PrandtlTurb

```

It is important to stress that the effective viscosity is added to the whole system of equation even if in the resource manager it is written *MomentumXEquation*.

4.2.3 Source Terms

The last step to build the troughflow model in Openlabs is the implementation of the source terms to the existing equation (continuity, momentum and energy). Basically the source terms we need to add are due to three different physical phenomenas:

- Inviscid blade forces due to the pressure field acting on the blade surface. The vector of the source terms has been computed in chapter 2. The syntax in Openlab languages is the follow:

```

1   @SOURCE: Bladeforce_teta
2   -> EXPRESSION: F_theta
3   -> AddToExistingPde: MomentumYEquation
4
5   @SOURCE: Bladeforce_zeta
6   -> EXPRESSION: F_axial
7   -> AddToExistingPde: MomentumZEquation
8
9
10  @SOURCE: Sourceangolodilean
11  -> EXPRESSION: -F_theta*tan(lambda)
12  -> AddToExistingPde: MomentumXEquation
13
14
15  @SOURCE: power
16  -> EXPRESSION: -P
17  -> AddToExistingPde: EnergyEquation

```

The value of the force has been calculated theoretically in chapter 2 and numerically in the *@EQUATIONS* section.

- The viscous blade forces are due to the viscous forces acting on the blade surface, the vector of viscous blade forces is formally the same of the inviscid blade forces. For this reason the implementation in Openlab is exactly the same.
- The last source term is the one due to the blockage factor; it has been theoretically calculated in chapter 2; it appears both on the RHS and the LHS of the equations. To make the implementation simpler it is convenient to move the blockage factor on the RHS of the equation. This can be done developing

by part the convective operator and doing some simplifications. After some algebra it leads to:

$$\mathbf{S}_{\text{block}} = \begin{bmatrix} -\rho u_z \frac{\partial_z b}{b} \\ -\rho u_z u_r \frac{\partial_z b}{b} \\ -\rho u_z u_\theta \frac{\partial_z b}{b} \\ -\rho u_z^2 \frac{\partial_z b}{b} \\ -\rho u_z H \frac{\partial_z b}{b} \end{bmatrix} \quad (4.8)$$

This vector can be implemented in the resource manager in exactly the same way of the inviscid forces so for sake of brevity only the continuity source term will be reported.

```

1 @SOURCE:BLF_cont
2 ->EXPRESSION: -Density*Vz/block*dblock
3 ->AddToExistingPde: ContinuityEquation
4

```

Chapter 5

Results

This chapter is dedicated to the application of the throughflow model built in the previous chapter on two test-cases. The first is an inviscid calculation performed thanks to an mid-line code with radial equilibrium, in the second comparison a NISRE code has been used to generate the geometry and to do the first calculation, then the geometry has been imported inside Numeca FINE/Turbo for the CFD calculation, the three different results have been compared.

5.1 Comparison with mid-line code

The first comparison has been performed between a mid-line plus radial equilibrium (with exponential vortex law) code and the throughflow code developed in the previous chapter. The calculation is an inviscid calculation. In order to run an inviscid calculation the user has to set flow model Euler in Numeca FINE/Open and then inside Openlabs switch off the entropy equation and the deterministic viscosity equation. In order to switch off these equations there are mainly two ways: the first one is to comment inside the script everything related to them, the second one is to set to zero source terms and initial and boundary conditions. The simulation is inviscid, no boundary layer is needed at the end-walls, for this reason the finer grid presented in chapter 3 is used without the adding of the boundary layer at tip and hub. For the reasons explained above in an inviscid calculation there are less equations to solve compared to a viscous one (three against eight) so the computation is very cheap and fast. The mid-line code has the aim to calculate the performance of the first stage of an helicopter turbine. The boundary conditions are four at the inlet P_T T_T $\text{atan}(\frac{u_\theta}{u_z})$ $\text{atan}(\frac{u_r}{u_z})$ and static pressure with radial equilibrium at the outlet.

	Inlet	Oulet
P_T [Pa]	1290881	
T_T [Pa]	1600	
$\text{atan}(\frac{u_\theta}{u_z})$ [deg]	0	
$\text{atan}(\frac{u_r}{u_z})$ [deg]	0	
P_s [Pa]		573485

First of all we check the global quantities such as massflow, Power output, rotor and stator deflection.

	Mid-line	Isoentropic Throughflow
\dot{m} [kg/s]	1.926	1.932
P_{output} [kW]	597	594
Stator deflection	73	73
Rotor deflection	113	110

There is a quite good agreement between the output from the mid-line and the ones from the throughflow. To go deeper into details of the simulations some radial sections have been extracted from the simulation. In particular the attention has been focused on the outlet stator inlet rotor section and the outlet rotor section. For every section the following quantities has been extracted: absolute and relative flow angle, axial and tangential component of velocity, static pressure and temperature, and also the specific massflow defined as $\dot{m}_{spec} = \rho u_z$

5.1.1 Radial Sections

The next step of our analysis is to compare some radial sections obtained from both codes. The mid-line code has been set up to perform an isoentropic computation, the results from the mid-line computation are extended to radial distribution applying the exponential vortex law. In order to perform an isoentropic calculation in the throughflow model everything related to viscosity and losses has been switched off. The viscous blade forces are set to zero, the entropy equation, the deterministic viscosity equation and the turbulent viscosity are switched off and the wall boundary conditions on the velocity are changed from Navier-Stokes' wall to Euler's wall. The only force field in the model are now the blockage factor contribution and the inviscid blade forces.

Below some radial sections of some kinematic and thermodynamic quantities downstream the stator and the rotor are reported the continuous line represents the throughflow output while the dotted line is the results of the mid-line.

From a first comparison the two calculations have a quite good agreement. The qualitative trend of the variables has been achieved however some variables seem to be shifted. In particular there is a good agreement between the flow angle in the upper part of the span while the trends are a little bit different in the lower part; axial and tangential velocity computed from the throughflow code are lower than the ones computed from the mid-line; this affects the value of the static pressure that is a little bit higher in th same way the static temperature and the Mach number that are lower. However the specific massflow is perfectly reproduced. The analysis of the radial sections confirms the the two codes are in agreement in the calculation of the total quantities (massflow, deflection angle, power output). The local differences in the stations inside the stage can be due to the different approach between the two codes, in particular the first is a mid-line calculation to whom as been applied a vortex law, while the throughflow is more close to a CFD calculation and has the capabilities to fetch more detailed flow features and a more detailed physic.

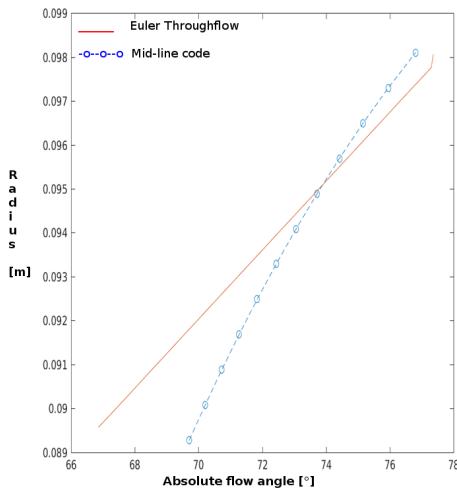


Figure 5.1: Absolute flow angle stator outlet

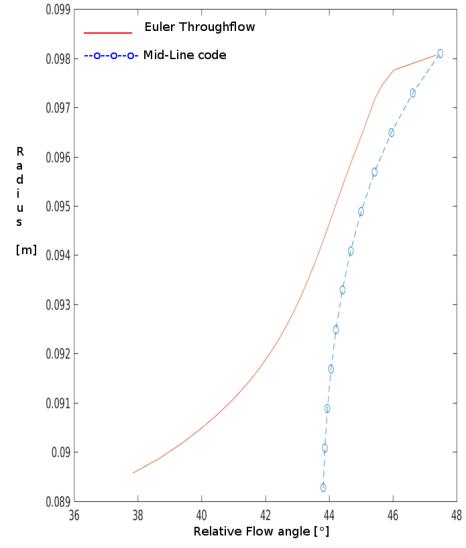


Figure 5.2: Relative flow angle stator outlet

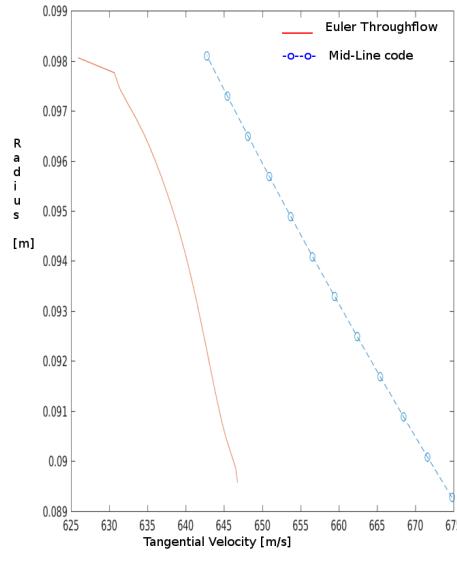


Figure 5.3: Tangential velocity stator outlet

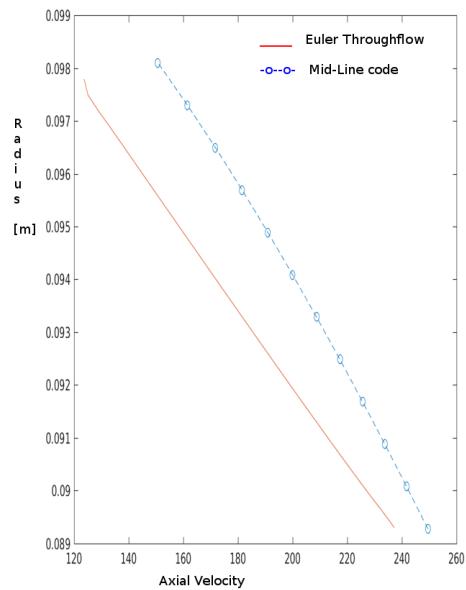


Figure 5.4: Axial velocity stator outlet

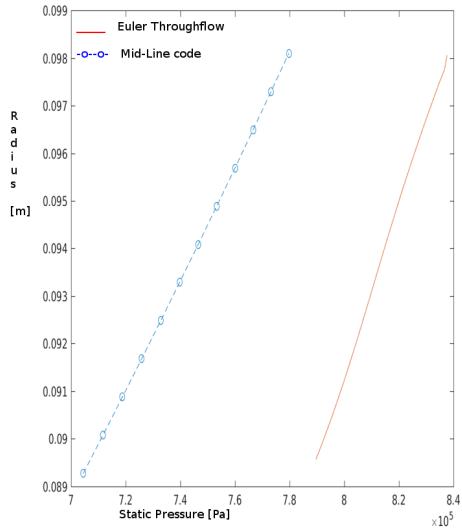


Figure 5.5: Static pressure stator outlet

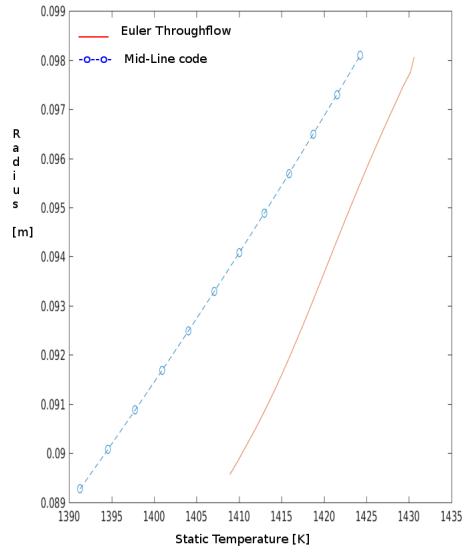


Figure 5.6: Static temperature stator outlet

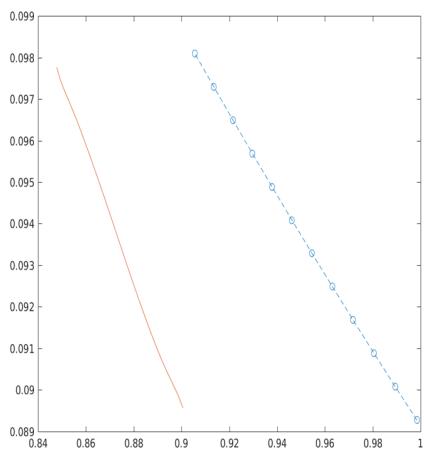


Figure 5.7: Absolute Mach number stator outlet

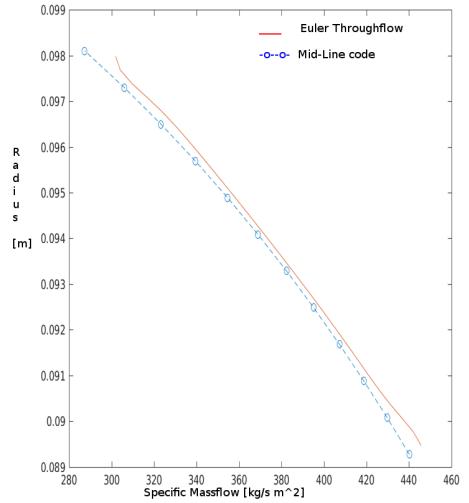


Figure 5.8: Specific massflow stator outlet

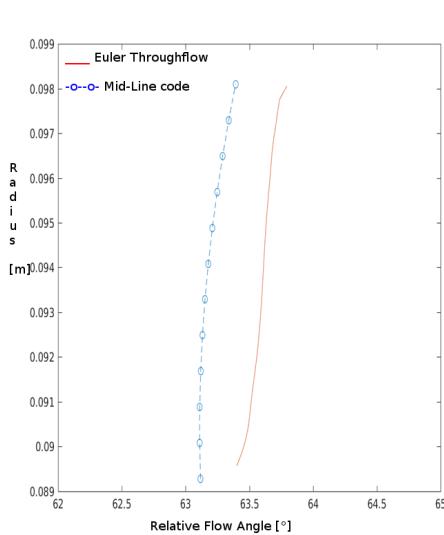


Figure 5.9: Relative flow angle rotor outlet

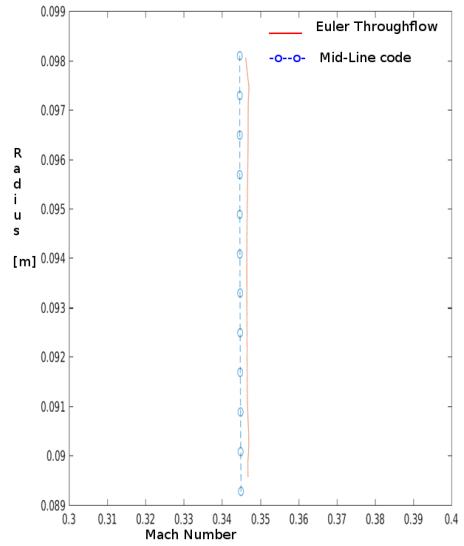


Figure 5.10: Absolute mach number rotor outlet

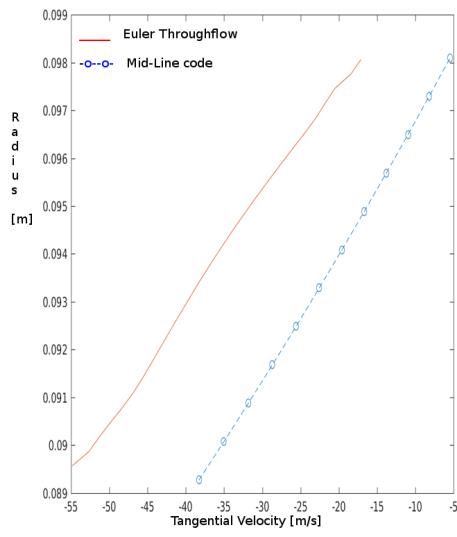


Figure 5.11: Tangential velocity rotor outlet

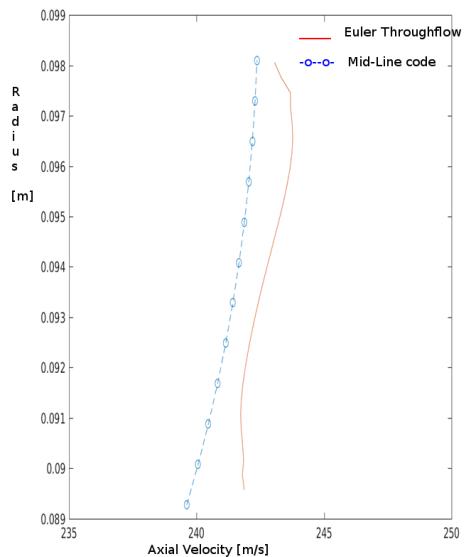


Figure 5.12: Axial velocity rotor outlet

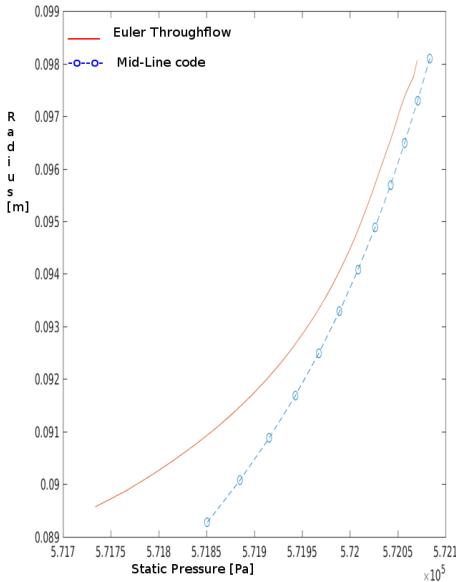


Figure 5.13: Static pressure rotor outlet

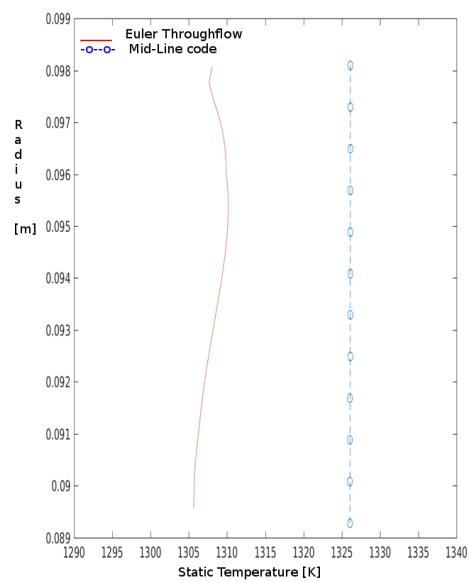


Figure 5.14: Static temperature rotor outlet

5.2 Comparison with NISRE and CFD calculation

The further step in the validation of the throughflow code is the comparison between the results obtained between a NISRE code, the thoughtflow Navier-Stokes and CFD calculation. In this simulation all the features of the throughflow are restored: correlation for losses are implemented, the equation for entropy transport, turbulence and deterministic stresses are switched on and the behaviour of the tip and hub flow is taken into account with the approach proposed by Gallimore presented in chapter 2. The boundary conditions and the geometry are the same for the three model.

	Inlet	Oulet
P_T [Pa]	162000	
T_T [Pa]	438	
$\tan(\frac{u_\theta}{u_z})$ [deg]	0	
$\tan(\frac{u_r}{u_z})$ [deg]	0	
P_s [Pa]		60000

In the figure above are reported a section of the blade to blade plane from a CFD calculation. The calculation shows that the geometry generated from the NIRSE is not the optimal one. We can notice that it generates a separation on the rotor suction side a tip, this could make the comparison on the local variables a little bit tricky since the throughflow model is not able to fetch separation because

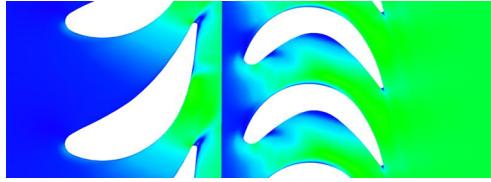


Figure 5.15: Mach number in a blade to blade section at hub

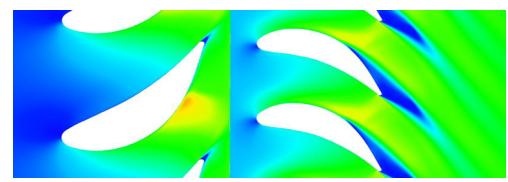


Figure 5.16: Mach number in a blade to blade section at tip

of the ensemble averaging performed on the equation (chapter 2) that removes from the flow every non deterministic features.

5.2.1 Radial and meridional sections

Here we report some meridional sections generated from the CFD calculation (left) and from the throughflow calculation (right).

	NISRE	TF	CFD
P_{output} [kW]	1,05	1,05	1,05
\dot{m} [kg/s]	12	12,05	12,3
$\frac{p_{01}}{p_3}$	2,68	2,7	2,7

As we did before with the mid-line code some thermodynamic and kinematic quantities at the outlet of the stator and of the rotor have been compared. The blue line represents the CFD calculation, the red one is the throughflow calculation and the dotted line is the NISRE.

All the quantities taken into account show a very good agreement between the three codes both from a qualitatively and quantitatively point of view. The only differences from the CFD calculation and the throughflow are the shape at the end-walls and the distribution of the axial velocity; the first one could be due to a non optimal description of the end-walls flow in the throughflow model, the second one is probably due to blade shape that is another big approximation in the throughflow model.

At the rotor outlet there is a quite good agreement from a qualitative point of view between all quantities. However the axial velocity and the total pressure show the same mean value but different distributions along the radius. This could be due both to the not optimal description of the end-wall and 3-D losses done by the correlation implemented in the throughflow model and the non optimal shape of the blades generated by the NISRE. In figure 5.16 a tip flow separation is highlighted, this kind of phenomena (stochastic) aren't taken in account in a throughflow model since we assume a axisymmetric deterministic and steady model.

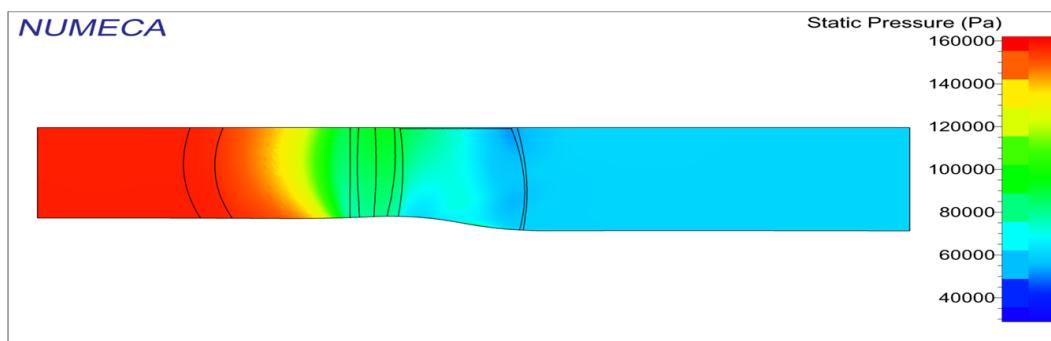


Figure 5.17: Static Pressure from CFD calculation

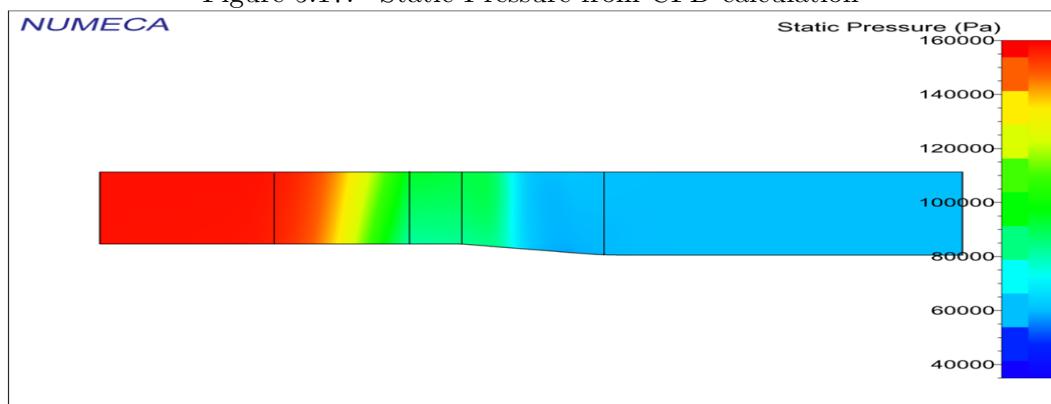


Figure 5.18: Static Pressure from throughflow calculation

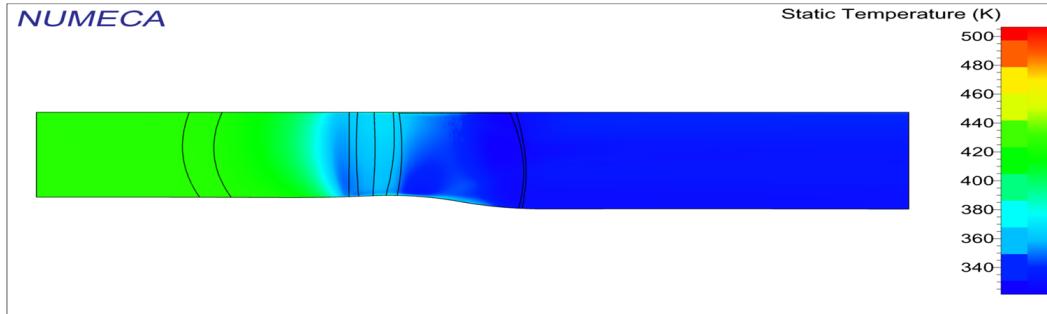


Figure 5.19: Static temperature from CFD calculation

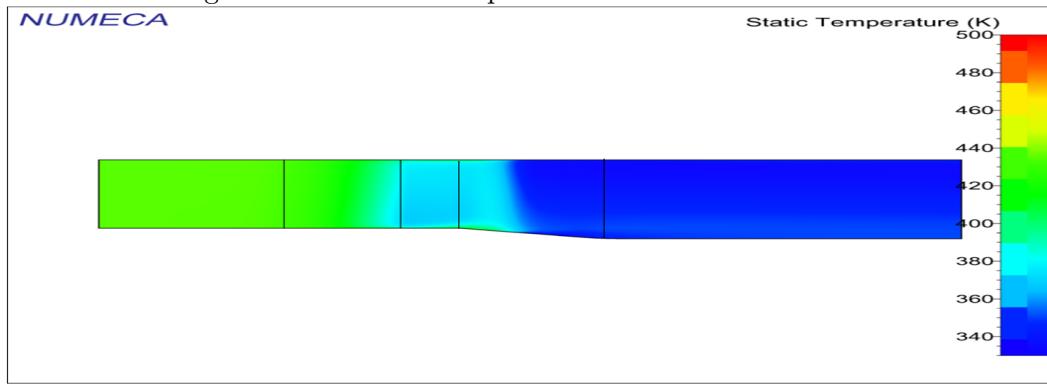


Figure 5.20: Static temperature from throughflow calculation

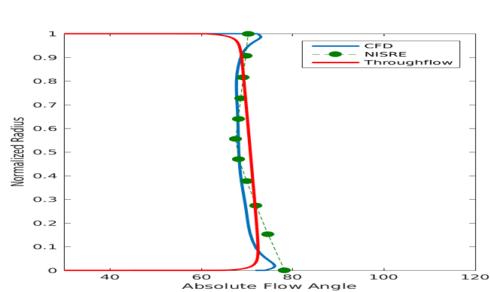


Figure 5.21: Absolute flow angle stator outlet

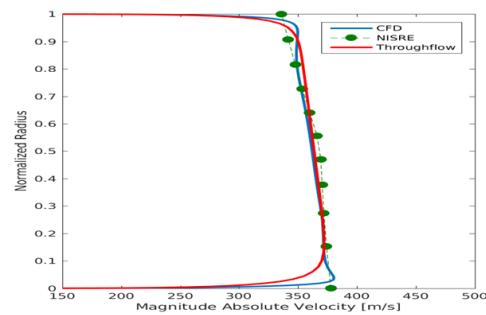


Figure 5.22: Absolute velocity stator outlet

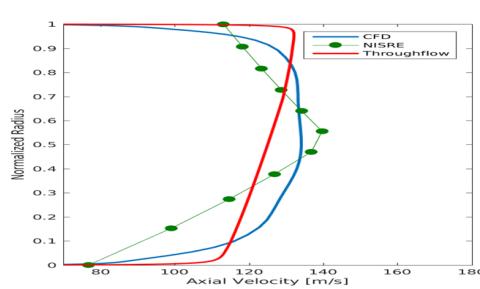


Figure 5.23: Axial velocity stator outlet

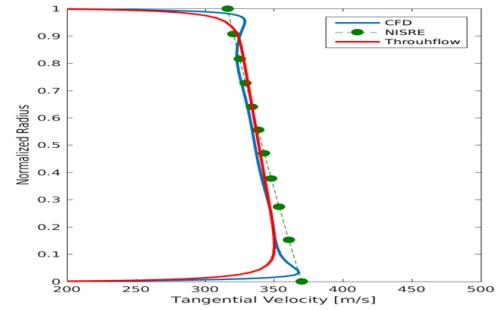


Figure 5.24: Tangential velocity stator outlet

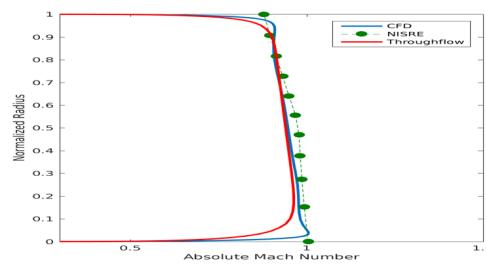


Figure 5.25: Absolute Mach number stator outlet

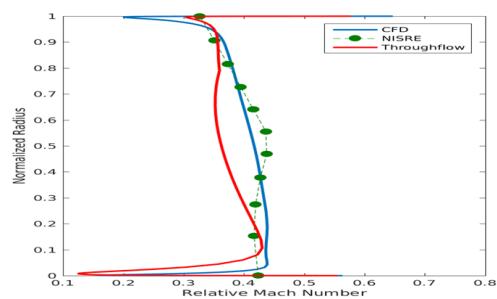


Figure 5.26: Relative Mach number stator outlet

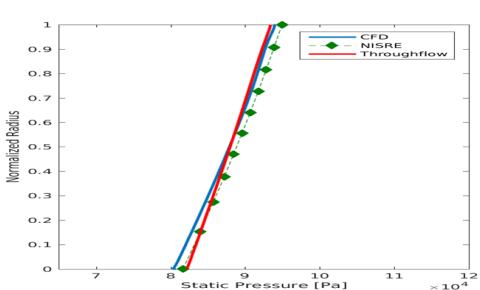


Figure 5.27: Static pressure stator outlet

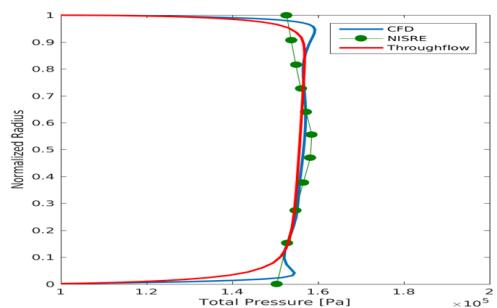


Figure 5.28: Total pressure stator outlet

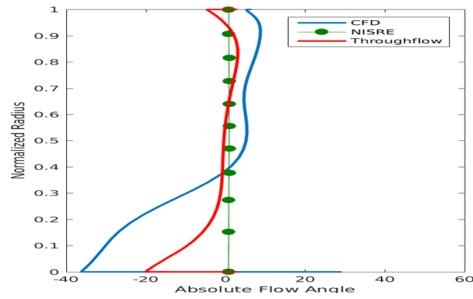


Figure 5.29: Absolute flow angle rotor outlet

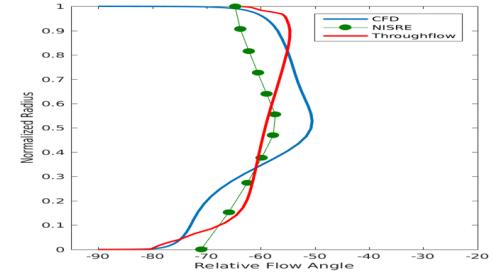


Figure 5.30: Relative flow angle rotor outlet

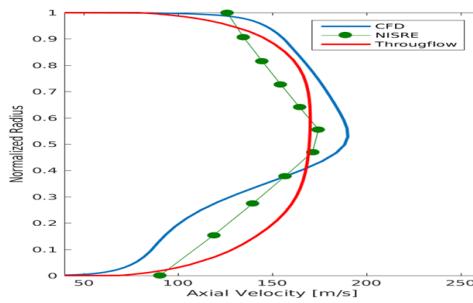


Figure 5.31: Axial velocity rotor outlet

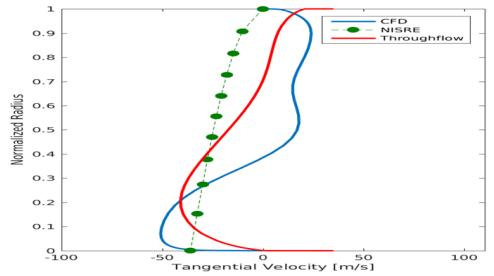


Figure 5.32: Tangential velocity rotor outlet

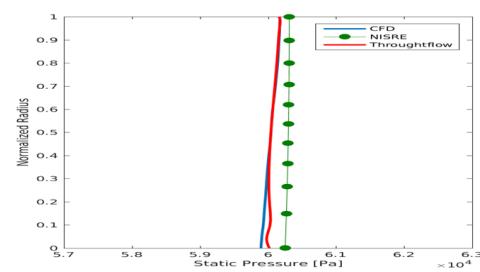


Figure 5.33: Static pressure rotor outlet

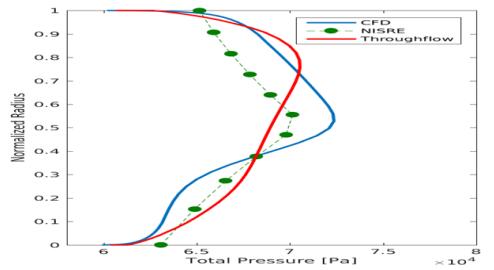


Figure 5.34: Total pressure rotor outlet

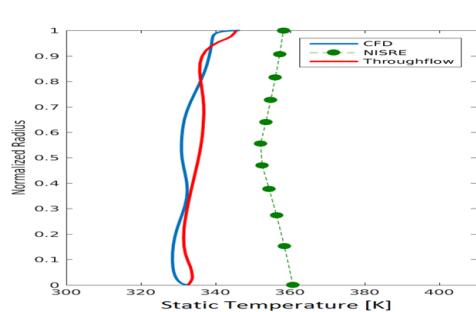


Figure 5.35: Static temperature rotor outlet

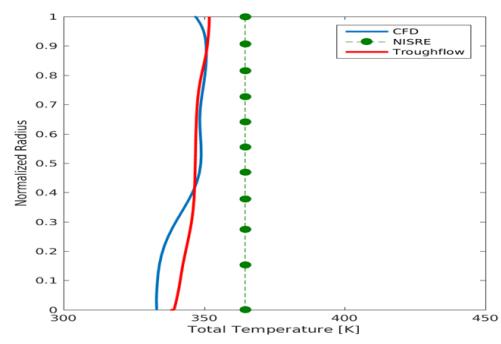


Figure 5.36: Total temperature rotor outlet

Chapter 6

Conclusions

The aim of this project has been the development of a throughflow model for the analysis of axial turbomachinery, in particular in this work we focused the efforts in studying turbine stage, however the model can be easily extended to compressor stages also. The theoretical development of the throughflow model has been developed in chapter two in three steps:

- Theoretical treatment of the equations, following the procedure proposed by Adamczyk in order to obtain a steady and axisymmetric model that is able to describe the flow in turbomachinery.
- Modelling of the various source terms arising from the average operation: blockage factor, inviscid blade forces and losses mechanism.
- Modelling of turbulence and deterministic flow features.

To model the inviscid blade forces very simple assumptions on the dynamic of the flow have been done in order to built the forces vector. In the frame of the losses mechanism we exploit the potential of the loss computation lying inside the Navier-Stokes equations, in order to capture the 3-D features of the flow we make use of losses correlation introduced by some authors and a model based of blades force modification proposed by Gallimore in order to capture the overturning at the hub and the under turning at the tip. The results of this approach give quite satisfactory result however there are still some improvements to do, in particular try to reduce the dependence of the model on previous empirical or numerical results. The last part of chapter two is dedicated to the modelling of the deterministic stresses; a reliable and computational cheap model that is able to describe the deterministic flow features is probably the key to obtain a good throughflow model. However finding a complete model that is able to describe this effect is a very demanding task. In this work the three main models have been presented and the attention was focused on the model based on the deterministic viscosity proposed by Stollenwerk. After the theoretical aspects of the throughflow model we dealt with the numerical aspects and the implementation. The implementation has been performed in Numeca FINE/Open with Openlabs. The implementation can be

easily divided into three sections: implementation of the geometrical quantities and algebraic expressions, implementation of the new transport models (entropy and deterministic viscosity) and finally implementation of the extra source terms related to blade flow interactions. The final chapter of this work is dedicated to the first validation of the model; The validation may be divided in two parts: the first one in which the inviscid features of the model has been compared to a mid-line with radial equilibrium inviscid calculation, in the second one the Navier-Stokes throughflow model has been compared with a CFD stage calculation. The comparison has been performed in the following way: a NISRE code has generated the geometry that has been imported in Numeca FINE/Turbo and in the throughflow, two calculations have been performed and compared between them and with the NISRE results. The results obtained are quite satisfactory and can be summarized as follows:

- The inviscid calculation compared to the mid-line with radial equilibrium gives a good agreement between the global quantities (massflow, power output and deflection angles) however there are some radial profiles that have the same trend but values a little different. This could be due to the two different approaches used.
- The most demanding test for the thoughtflow model has been a comparison with a CFD calculation. The comparison showed a very good agreement between the throughflow and the CFD result especially in the section downstream of the stator, the only remaining differences are at the end-walls where the empirical correlations are not able to catch completely the flow features. In the section downstream of the rotor there is still a good agreement in the mean quantities however some profiles are slightly different; this is probably due to the non optimal description of the 3-D flow features and the effects of the deterministic stresses; indeed in the rotor they are much more important than in the stator.

Finally we can state that the throughflow model gives satisfactory results compared with other analysis codes, the implementation in Openlabs is straightforward and avoids the users to deal with more technical aspects of numerical and informatic nature. About the future challenges the most pressing topics could be:

- Improve the modelling of the 3-D losses focusing our attention on the modelling of the tip flow, the model proposed by Gallimore in our case gives quite good results but they can be improved and generalized.
- More deep understanding of the role of circumferential and deterministic stresses in the model and in particular developing of a computational cheaper and more accurate model to describe their evolution.
- A step further in the validation of the model could be to insert other source terms in order to describe more detailed physical features, such as cooling injection from the case in different axial positions.

- In this work we focused our attention on the analysis mode in which the geometry of the machine is known; a very interesting future challenge could be develop the code in design mode.
- Compare with other CFD solvers and Troughflow solver of known geometries.
- In order to make the Throughflow solver more general it could be very useful parametrize stator rotor profile with parametric curves (like B-Spline or Bezier cuves).
- Automatize the process of data input and output with a more user friendly batch script.

References

- [1] . *Numeca user guide*. Numeca international, July 2008.
- [2] Craig J. Fischberg Robert M. Zacharias Aaron J. Gleixner, David A. Spear. Multistage Modelling Using Bodyforces abd deterministic Stresses. *The American Society of Mechanical Engineering*, 1995.
- [3] John J. Adamczyk. Wake mixing in axial flow compressor. *The American society of mechanical engineers*, 1996.
- [4] John D. Adamzycz. *Model Equation for Simulating Flow in Multistage Turbomachinery*. NASA, 1964.
- [5] John J. Adamczyk Allan G. van de Wall, Jaikrishnan R. Kadambi. A Transport Model for the Deterministic Stresses Associated With Turbomachinery Blade Row Interactions. *Journal of Turbomachinery*, 122, 2000.
- [6] M.Lawerenz A.Willburger. Consideration of Three-Dimensional Phenomena within Two-Dimensional Meridional Flow model. *AIAA*, 2010.
- [7] F.Lebouf D.Charbonnier. Steady Flow Simulation Of Rotor-Stator interactions With A New Unsteady Flow Model. *AIAA*, 2004.
- [8] John D. Denton. Entropy Generation in Turbomachinery Flow. 99, 1990.
- [9] roger L.Davis Douglas L.Sondak, Daniel J. Dorney. Modeling Turbomachinery Unsteadiness with Lumped Deterministic Stresses. *ASME*, 1996.
- [10] Olivier Leleonard Jean Francois Simon. Modeling of 3-D Losses and Deviations in a Throughflow Analysis Tool. *Journal of Thermal Science*, 16:1–7, 2006.
- [11] O.Leonard J.F. Simon. On the role of the deterministic and circumferential stresses in throughflow calculations. *ASME Turbo expo*, 2008.
- [12] Piort Lampart. Investigation of Endwall Flows and Losses in Axial Turbine Part 1 Formation of Endwall Flows and Losses. *Journal of Theoretical and Applied Mechanics*, 47:321–324, 2009.
- [13] Paul D. Orkwis Marshall C.Galbraith. Transpor Equation for Deterministic Stesses. *AIAA*, 2008.

- [14] E.G.Plett M.J. Kermani. Modified Entropy Correction Formula for the Roe Scheme. *AIAA*, 2001.
- [15] Josef Katz Charles Meneneau Oguz Uzol, Yi-Chih Chow. Average passage flow field and deterministic stresses in the tip and hub regions of a multi-stage turbomachine. 1:2, 2003.
- [16] J.F. Simon. *Contribuition to Throughflow Modelling for Axial Flow Turbomachines*. PhD thesis, Universit de Liege, 2007.
- [17] Francesco Bertini Simone Rosa Taddei, Francesco Larocca. A treatment of the leading edge discontinuity for time-marching throughflow analyses. *AIAA*, 2008.
- [18] Dirk Nuernberger Stefan Stollenwerk. A Model for Potential Deterministic Effects in Turbomachinery Flows. 2008.
- [19] Edmund Kugeler Stefan Stollenwerk. Deterministic stress modeling for multi-stage compressor flowfields. *ASME Turbo Expo 2013*, 2013.
- [20] Stefen Stollenwerk. *Modellierung deterministischer Spannungen in Turbomaschinenstromungen*. PhD thesis, Vrije Universiteit Brussel, 200.
- [21] Andreas Sturmayr. *Throughflow modelling*. PhD thesis, Vrije Universiteit Brussel, 2004.
- [22] S. Rosa Taddei. *Sviluppo di strumenti per l'ottimizzazione del progetto aerodinamico di turbomacchine assiali*. PhD thesis, Politecnico di Torino, 2008.

Appendix A

Appendix A

Openlabs' script

For sake of completeness in this appendix the whole Openlabs script is reported. In the chapter four the basic structure of the script has been explained. In order to make easier the reading the seven section will be presented separately. About the notation: the double backslash is used to divide the expression (if too long) in order to fit into the box and is not a Openlab's operatore while the single slash is the Openlab's operator in order to wrap around a new line.

A.0.1 CONSTANT

```
1 #geometrical properties
2 @z_Sini=0.073          #stator LE
3 @z_Sfin=0.11           #stator TE
4 @z_Rini=0.1289         #rotor LE
5 @z_Rfin=0.173          #rotor TE
6 @Rt=
7 @Rh=
8 @h=
9
10 #variance of the functions that describe the turning
11 #inside stator and rotor
12 #the functions are gaussian like y=C*exp(-((z-c)/k)^q)
13
14 @kstat=0.06
15 @krot=0.02
16
17 #exponent q for stator and rotor
18 #deflection function
19 @qs=4
20 @qr=4
21
22 #location of the max of the deflection function
23 @c=0.12
24 @cr=0.12
25
26
```

```
27 @pi=3.14159265359
28 #####
29 #blockage factor parametres
30 #the blockage factor is described by a patchwork
31 #of gaussian functions with the following parametres
32 #
33 #max value of the blockage factor
34 @a1=0.18
35 @a2=0.18
36 @a3=0.18
37 #
38 #variance of the gaussian
39 @k1=0.02
40 @k2=0.01
41 @k2bis=0.02
42 @k3=0.025
43 #
44 #location of the max value
45 #with these parametres one can model
46 #the "shape" of the blades
47 @b1=0.075
48 @b2=0.12
49 @b3=0.151
50 #
51 #####
52 #control parametres
53 #useful to model the twist
54 #it is usel to play with these parametres to reach
55 #a faster cnvergence
56 @sv=0.8
57 @rs=-0.2
58 #
59 #angular velocity
60 @OMEGA=680
61 #
62 #stator angles
63 @DS=72 #rotor deflection
64 #
65 #rotor angles
66 @DR=100 #rotor deflection
67 @R0=60 #rotor relative out angle
68 #
69 #losses parametres for Roberts correlations
70 @TC=1
71 @AR=2
72 @sigma=1
73 #
74 #
75 @kr=0.003
```

A.0.2 EQUATIONS

```

1  @NONDIFFEQ: Smoothangle
2  ->EXPRESSION: Smooth=smooth
3
4  @NONDIFFEQ: Deflectionangle
5  ->EXPRESSION: alpha=alpha0
6  ->BOUNDARY_FACE: Extrapolation
7
8  @NONDIFFEQ: GradAlpha
9  ->EXPRESSION: dalpha=dalpha0
10 ->BOUNDARY_FACE: Extrapolation
11
12 @NONDIFFEQ: TetaBladeForceSTAT
13 ->EXPRESSION: FtetaS=\\
14 Density*Vz*(GRAD(Vz,z)*tan(alpha)+Vz*dalpha)
15
16 @NONDIFFEQ: TetaBladeForceROT
17 ->EXPRESSION: FtetaR=\\
18 Smooth*Density*Vz*(GRAD(Vz,z)*tan(beta)+Vz*dbeta)
19
20 @NONDIFFEQ: Source_R_lean
21 ->EXPRESSION: Frlean=-Fteta*tan(lambda)
22
23 @NONDIFFEQ: Statorxcomponent
24 ->EXPRESSION: Fz_S=-FtetaS*tan(alpha)
25
26 @NONDIFFEQ: Rotorxcomponent
27 ->EXPRESSION: Fz_R=-FtetaR*tan(beta)
28
29 @NONDIFFEQ: Wrelteta
30 ->EXPRESSION: Wteta=Wteta0*ROT
31
32 @NONDIFFEQ: Relativeflowangle
33 ->EXPRESSION: beta=beta0
34
35 @NONDIFFEQ: defangolorel
36 ->EXPRESSION: dbeta=dbeta0
37
38 @NONDIFFEQ: totalthetaforce
39 ->EXPRESSION: Fteta=IF(zCoord<z_Sfin)(FtetaS) \
40 ELSEIF(zCoord>z_Rini AND zCoord<z_Rfin AND rCoord<Rh-TC*h)(FtetaR)\ \
41 ELSE 0.0
42
43 @NONDIFFEQ: totalaxialforce
44 ->EXPRESSION: Fz=IF(zCoord<z_Sfin)(Fz_S) \
45 ELSEIF(zCoord>z_Rini AND zCoord<z_Rfin)(Fz_R) \
46 ELSE 0.0
47
48 @NONDIFFEQ: Power
49 ->EXPRESSION: P=fabs(Fteta*Omega*rCoord*ROT)

```

```
50
51 @NONDIFFEPR : cx
52 ->EXPRESSION : Cx=-Density*((pow(0mega ,2))*rCoord+\\
53 -2*Wteta*0mega)
54
55 @NONDIFFEPR : cy
56 ->EXPRESSION : Cy=-Density*Vx*0mega
57
58 @NONDIFFEPR : Poweroutput
59 ->EXPRESSION : Ptot=INTEG3D(P)*360/10
60
61 @NONDIFFEPR : Powercorrected
62 ->EXPRESSION : Pshaft=INTEG3D(P*block)*360/10
63
64 @NONDIFFEPR : verificadeflessione
65 ->EXPRESSION : alphaver=atan(VtCylinder/Vz)
66
67 @NONDIFFEPR : blockage
68 ->EXPRESSION : block=bf
69
70 @NONDIFFEPR : dblockage
71 ->EXPRESSION : dblock=dbf
72
73 @NONDIFFEPR : Massflowinlet
74 ->EXPRESSION : M_in=INTEG2D(Density*Vz , ALLINLETS)*360/10
75
76 @NONDIFFEPR : AbsVrel
77 ->EXPRESSION : absW=sqrt(pow(Wteta ,2)+pow(Vz ,2)+\\
78 pow(Vx ,2))*ROT
79
80 @NONDIFFEPR : Relative_Enthalpy
81 ->EXPRESSION : Hrel=(Enthalpy+0.5*pow(absW ,2)+\\
82 -0.5*(rCoord*0mega ,2))*ROT
83
84 @NONDIFFEPR : statfunc
85 ->EXPRESSION : S=STAT
86
87 @NONDIFFEPR : rotfunction
88 ->EXPRESSION : R=ROT
89
90 @NONDIFFEPR : Velocitatrascinamento
91 ->EXPRESSION : U=rCoord*0mega
92
93 @NONDIFFEPR : Relativeframeangle
94 ->EXPRESSION : betaver=atan(Wteta/Vz)
95
96 ##########
97 # losses correlation implementation
98
99 #bounday layer losses stator
100
```

```
101 @NONDIFFEPR: Boundaylayerthickness
102 ->EXPRESSION:delta_s=IF(zCoord>z_Sini AND zCoord<z_Sfin)\\
103 (0.38*(zCoord-z_Sini)/pow((zCoord-z_Sini)*absU/ni,0.2)) \
104 ELSE 0.0
105
106 @NONDIFFEPR: MomentumThicknessstat
107 ->EXPRESSION:teta_s=IF(zCoord>z_Sini AND zCoord<z_Sfin)\\
108 (0.037*(zCoord-z_Sini)/pow((zCoord-z_Sini)*absU/ni,0.2)) \
109 ELSE 0.0
110
111 @NONDIFFEPR: Reynoldstetastat
112 ->EXPRESSION:Reteta_s=teta_s*absU/ni
113
114 @NONDIFFEPR: CoeffDragstat
115 ->EXPRESSION:Cdrag_s=IF(zCoord>z_Sini AND zCoord<z_Sfin)\\
116 (0.0056/pow(Reteta_s,0.16666)) \
117 ELSE 0.0
118
119 @NONDIFFEPR: EntropySourcestat
120 ->EXPRESSION:SgenStat=IF(zCoord>z_Sini AND zCoord<z_Sfin)\\
121 (0.5*(Zita+Zita_1+Zita_2)*Density*pow(absU,3)/StaticTemperature+\\
122 0.5*fstat*Density*pow(absU,3)/StaticTemperature)
123 ELSE 0.0
124
125 @NONDIFFEPR: Forceduetointropysourcestat
126 ->EXPRESSION:Fzs_s=IF(zCoord>z_Sini AND zCoord<z_Sfin)\\
127 (Density*StaticTemperature*Vz*GRAD(Sgen,z)/absU) \
128 ELSE 0.0
129
130 #boundary layer losses rotor
131
132
133 @NONDIFFEPR: SpessoreStratoLimiterot
134 ->EXPRESSION:delta_r=IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
135 (0.38*(zCoord-z_Rini)/pow((zCoord-z_Rini)*absW/ni,0.2)) \
136 ELSE 0.0
137
138 @NONDIFFEPR: MomentumThicknessrot
139 ->EXPRESSION:teta_r=IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
140 (0.037*(zCoord-z_Rini)/pow((zCoord-z_Rini)*absW/ni,0.2)) \
141 ELSE 0.0
142
143 @NONDIFFEPR: Reynoldstetarot
144 ->EXPRESSION:Reteta_r=teta_r*absW/ni
145
146 @NONDIFFEPR: CoeffDragrot
147 ->EXPRESSION:Cdrag_r=IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
148 (200*0.0056/pow(Reteta_r,0.16666)) \
149 ELSE 0.0
150
151 @NONDIFFEPR: EntropySourcerot
```

```
152 ->EXPRESSION:SgenRot=IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
153 (0.5*fr*Density*pow(absW,3)/StaticTemperature+\\
154 0.5*Cdrag_r*Density*pow(absW,3)/StaticTemperature)
155 ELSE 0.0
156
157 @NONDIFFEQ:Forceduetointropysourcerot
158 ->EXPRESSION:FzS_r=IF(zCoord>z_Rini AND zCoord<z_Rfin)
159 (Density*StaticTemperature*Vz*GRAD(Sgen,z)/absW) \
160 ELSE 0.0
161
162 ######
163 # BOUNDARYLAYER TIP AND HUB
164
165 @NONDIFFEQ:Attritiviscosi
166 ->EXPRESSION:S_attriti=phi*ni/StaticTemperature
167
168 #####
169 #GRAD TERMICI
170
171 @NONDIFFEQ:Generazione_entropia_grad_termici
172 ->EXPRESSION:S_grad_T=\\
173 (Conductivity+Cp*(EddyVisc+muD)/phys_PrandtlTurb_)/StaticTemperature*\\
174 (GRAD(GRAD(StaticTemperature,x),x)+\\
175 GRAD(GRAD(StaticTemperature,z),z))
176
177 #####
178 #ENTROPY TRASPORTATION EQUATION
179
180 @PDE:EntropyTransport
181 ->EXPRESSION:DDT(Sgen)+CONV(Sgen)=SOURCE(SgenSource_s)\\
182 +SOURCE(SgenSource_r)+SOURCE(SgenSource_attritiviscosi)\\
183 +SOURCE(Grad_termici)
184 ->Convective_Scheme: UpwindFirstOrderConservative
185 ->UseCustomBC: SgenInlet, 0
186 ->InitializeTo: SgenInit
187 ->ConstantRelaxFactor:0.5
188 ->LowerLimit:1e-20
189 ->UpperLimit:100
190
191 #####
192 #Trasport equation deterministic stresses
193
194 @PDE:DeterministicStresses
195 ->EXPRESSION:DDT(muD)+CONV(muD)=DIFF(muD_diffcoeff)\\
196 +SOURCE(muD_source)
197 ->Convective_Scheme:UpwindFirstOrderConservative
198 ->UseCustomBC:muDInlet,0,1,2
199 ->InitializeTo:muDInit
200 ->ConstantRelaxFactor:0.5
201 ->LowerLimit:1e-20
202
```

```

203 #####
204 ##### #deterministic stresses visualization
205 #####
206 @NONDIFFEPR:tauzz
207 ->EXPRESSION:tD_zz=0.66*muD*(2*GRAD(Vz,z)-GRAD(Vx,x)-Vx/rCoord)
208 #####
209 @NONDIFFEPR:taurr
210 ->EXPRESSION:tD_rr=0.66*muD*(2*GRAD(Vx,x)-GRAD(Vz,z)-Vx/rCoord)
211 #####
212 @NONDIFFEPR:tautt
213 ->EXPRESSION:tD_tt=0.66*muD*(2*Vx/rCoord-GRAD(Vz,z)-GRAD(Vx,x))
214 #####
215 @NONDIFFEPR:tauZR
216 ->EXPRESSION:tD_zr=muD*(GRAD(Vx,z)+GRAD(Vz,x))
217 #####
218 @NONDIFFEPR:tauZT
219 ->EXPRESSION:tD_zt=muD*GRAD(Vy,z)
220 #####
221 @NONDIFFEPR:taurt
222 ->EXPRESSION:tD_rt=muD*(GRAD(Vy,x)-Vy/rCoord)
223 #####
224 ##### #source terms for deterministic stresses
225 #####
226 @NONDIFFEPR:lossfunction
227 ->EXPRESSION:LS=fstat
228 #####
229 @NONDIFFEPR:losscoeff
230 ->EXPRESSION:LR=fr
231 #####
232 ##### #source terms for deterministic stresses
233 #####
234 @NONDIFFEPR:Streamlinecurvature
235 ->EXPRESSION:S_curv=GRAD(GRAD(alpha,z),z)
236 #####
237 @NONDIFFEPR:Rel_treamlinecurvature
238 ->EXPRESSION:S_curv_rot=MIN(GRAD(GRAD(beta,z),z),500000)
239 #####
240 @NONDIFFEPR:Det_Stress_Sourceterm_rot
241 ->EXPRESSION:S_muDrot=S_Drot
242 ->BOUNDARY_FACE:Extrapolation
243 #####
244 #@NONDIFFEPR:Det_Stress_Sourceterm_stat
245 #->EXPRESSION:S_muDstat=IF(zCoord>z_Sini AND zCoord<z_Sfin+\\
246 0.2*(z_Sfin-z_Sini))(S_Dstat) ELSE 0.0
247 #####
248 @NONDIFFEPR:Reynoldsnumber
249 ->EXPRESSION:Re=zCoord*absU/ni
250 #####
251 @NONDIFFEPR:Reynoldsnumber
252 ->EXPRESSION:Re=zCoord*absU/ni
253

```

```

254 @NONDIFFEPR : Mach_relative
255 -> EXPRESSION : Mach_rel = absW / SoundSpeed
256
257 @NONDIFFEPR : LeanAngle
258 -> EXPRESSION : Lean_angle = lambda

```

A.0.3 CUSTOM BOUNDARY CONDITIONS

```

1 => CUSTOM_BOUNDARY_CONDITIONS
2
3
4 @CUSTOMIZED_BOUNDARY_CONDITION : SgenInlet
5 -> EXPRESSION : 0
6
7 @CUSTOMIZED_BOUNDARY_CONDITION : muDInlet
8 -> EXPRESSION : 0.000001

```

A.0.4 ALGEBRAIC DIFFUSION COEFFICIENTS

```

1 @ALGDIFFCOEFF : muD_diffcoeff
2 -> EXPRESSION : Viscosity + EddyVisc + muD
3
4 ######
5
6 @ALGDIFFCOEFF : EffectiveViscosity
7 -> EXPRESSION (MomentumXEquation) : Viscosity + EddyVisc + muD
8 @ALGDIFFCOEFF : EffectiveConductivity
9 -> EXPRESSION (EnergyEquation) : Conductivity + \
10   Cp * (EddyVisc + muD) / phys_PrandtlTurb_

```

A.0.5 INITIAL PROFILES

```

1 @INITIAL_FIELD : SgenInit
2 -> EXPRESSION : 0
3
4 @INITIAL_FIELD : muDInit
5 -> EXPRESSION : 0

```

A.0.6 SOURCETERMS

```

1
2 ######
3 # Entropy sourceterms
4
5 @SOURCE : SgenSource_s
6 -> EXPRESSION : IF(zCoord < z_Sfin AND zCoord > z_Sini) (SgenStat)
7 ELSE 0.0
8
9 @SOURCE : SgenSource_r
10 -> EXPRESSION : IF(zCoord < z_Rfin AND zCoord > z_Rini) (SgenRot)
11 ELSE 0.0
12

```

```
13 @SOURCE: SgenSource_attritiviscosi
14 ->EXPRESSION: S_attriti
15
16 @SOURCE: Grad_termici
17 ->EXPRESSION: S_grad_T
18
19 ######
20 #Deterministic stresses source terms
21
22 @SOURCE: muD_source
23 ->EXPRESSION: IF(zCoord>z_Rini AND zCoord<z_Rfin)(S_muDrot)
24 ELSE 0.0
25
26
27 #####
28
29 @SOURCE: Bladeforceteta
30 ->EXPRESSION: a*Fteta
31 ->AddToExistingPde: MomentumYEquation
32
33 @SOURCE: noninerzx
34 ->EXPRESSION: -Density*((pow(Omega,2))*rCoord+2*Wteta*Omega)
35 ->AddToExistingPde: MomentumXEquation
36
37 @SOURCE: noninerzy
38 ->EXPRESSION: -Density*Vx*Omega
39 ->AddToExistingPde: MomentumYEquation
40
41 @SOURCE: Bladeforcezeta
42 ->EXPRESSION: a*Fz - fabs(FzS_s) -fabs(FzS_r)
43 ->AddToExistingPde: MomentumZEquation
44
45 @SOURCE: Sourceangolodilean
46 ->EXPRESSION: -Fteta*tan(lambda)
47 ->AddToExistingPde: MomentumXEquation
48
49 @SOURCE: power
50 ->EXPRESSION: -a*P
51 ->AddToExistingPde: EnergyEquation
52
53 #source term blockage factor
54
55 @SOURCE: BLF_cont
56 ->EXPRESSION: -Density*Vz/block*dblock
57 ->AddToExistingPde: ContinuityEquation
58 @SOURCE: BLF_z
59 ->EXPRESSION: (-Density*Vz*Vz/block*dblock)
60 ->AddToExistingPde: MomentumZEquation
61 @SOURCE: BLF_r
62 ->EXPRESSION: (-Density*Vz*Vx/block*dblock)
63 ->AddToExistingPde: MomentumXEquation
```

```

64 @SOURCE:BLF_t
65 ->EXPRESSION: (IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
66 (-Density*Vz*Wteta/block*dblock)
67 ELSE(-Density*Vz*Vy/block*dblock))
68 ->AddToExistingPde: MomentumYEquation
69 @SOURCE:BLF_Energy
70 ->EXPRESSION: (-Density*Vz*(AbsoluteTotalEnergy+\\
71 StaticPressure/Density)/block*dblock)
72 ->AddToExistingPde: EnergyEquation

```

A.0.7 AUXTERMS

```

1 #absolute flow angle and twist
2
3
4 @alpha0=IF(zCoord<c)\\
5 (DS*pi/180*exp(-pow(((zCoord-c)/kstat),qs))\\
6 *(1-sv*(1.71*rCoord-0.6156))) \\
7 ELSE (((DS+10)*pi/180*(exp(-pow(((zCoord-c)/(kstat*0.5)),qs)))+\\
8 -10*pi/180)*(1-sv*(1.71*rCoord-0.6156)))
9
10 @dalpha0=IF(zCoord>0 AND zCoord<c)
11 ((1/pow(cos(alpha),2))\\
12 *(-qs*(DS*pi/180)/kstat)\\
13 *pow(((zCoord-c)/kstat),qs-1)*\\
14 exp(-pow(((zCoord-c)/kstat),qs))\\
15 *(1-sv*(1.71*rCoord-0.6156)))
16 ELSE (((1/pow(cos(alpha),2))*\\
17 (-qs*((DS+5)*pi/180)/kstat)\\
18 *pow(((zCoord-c)/kstat),qs-1))\\
19 *exp(-pow(((zCoord-c)/kstat),qs)))*\\
20 (1-sv*(1.71*rCoord-0.6156)))
21
22 @Omega=IF(zCoord>z_Rini AND zCoord<z_Rfin)(OMEGA)
23 ELSE 0.0
24
25 #relative velocity
26
27 @Wteta0=IF(zCoord>0.017 AND zCoord<z_Rfin)\\
28 (Vy-Omega*rCoord)
29 ELSE 0.0
30
31 #relative flow angles
32
33 @beta0=IF(zCoord>cr AND zCoord<z_Rfin)
34 ((DR*pi/180*exp(-pow(((zCoord-cr)/krot),qr)))+\\
35 -(R0*pi/180))*(1-rs*(16.45*rCoord-5.8291))\\
36 *(-22.7*zCoord+3.927))) ELSE 0.0
37
38 @dbeta0=IF(zCoord>cr AND zCoord<z_Rfin)
39 ((1/pow(cos(beta),2))*(-qr*DR*pi/180/krot*pow(((zCoord-cr)/krot),qr-1))\\
40 *exp(-pow(((zCoord-cr)/krot),qr)))*(1-rs*(16.45*rCoord-5.8291))\\

```

```
41 * (-22.7*zCoord+3.927))) ELSE 0.0
42
43 #lean angle
44
45 @lambda=IF(zCoord>z_Rini AND zCoord<z_Rfin)\\
46 (2*pi*0.005/0.118*cos((rCoord+0.02)*2*pi/0.118))
47 ELSE 0.0
48
49 #blockagefactor
50
51 @bf=IF(zCoord>=0 AND zCoord<=b1)(1-a1*exp(-pow((zCoord-b1)/k1,2))) \
52 ELSEIF(zCoord>b1 AND zCoord<b2)(1-(a1-a2*exp(-pow((zCoord-b2)/k2bis,4))))\
53 ELSEIF(zCoord>b1 AND zCoord<b3)(1-(a1-a2*exp(-pow((zCoord-b2)/k2,2))))\
54 ELSE (1-a1*exp(-pow((zCoord-b3)/k3,2))) \
55
56
57 @dbf=IF(zCoord>=0 AND zCoord<=b1)
58 (2*a1/k1*(zCoord-b1)/k1*exp(-pow((zCoord-b1)/k1,2))) \
59 ELSEIF(zCoord>b1 AND zCoord<b2)
60 (-4*a2/k2bis*pow((zCoord-b2)/k2bis,3)*exp(-pow((zCoord-b2)/k2bis,4)))
61 ELSEIF(zCoord>b1 AND zCoord<b3)
62 (-2*a2/k2*pow((zCoord-b2)/k2,1)*exp(-pow((zCoord-b2)/k2,2))) \
63 ELSE
64 (2*a1/k3*(zCoord-b3)/k3*exp(-pow((zCoord-b3)/k3,2))) \
65
66 @ROT=IF(zCoord>=z_Rini AND zCoord<=z_Rfin)(1) \
67 ELSE 0.0
68
69 @STAT=IF(zCoord>=z_Sini AND zCoord<=z_Sfin)(1) \
70 ELSE 0.0
71
72 @absV=sqrt(pow(Vx,2)+pow(Vy,2)+pow(Vz,2))
73
74 #introduzione correlazione di perdite
75 @expo=Gamma/(Gamma-1)
76 @Machabs=absV/SoundSpeed
77 @Machrel=absW/SoundSpeed
78 @TotalPressure=StaticPressure*\\
79 pow(AbsoluteTotalEnthalpy/(1006*StaticTemperature),expo)
80 @absU=sqrt(Vx*Vx+Vy*Vy+Vz*Vz)
81
82 ##########
83 #secondary losses for stator
84
85 @Zita=4*0.02*0.02*DS*pi/180*0.008*4/3*c1/(73*pi/180)/0.1
86
87 @c1=0.25*(tan(DS*pi/180)/pow(cos(DS*pi/180),3)+3*c2)
88
89 @c2=0.5*(tan(DS*pi/180)/cos(DS*pi/180))+\\
90 0.5*log(tan(DS*pi/180)+1/cos(DS*pi/180))
```

```
91 #####
92 #Endwall losses stator
93
94 @Zita_1=4*0.2*0.008/0.1/cos(DS*pi/180)
95 @Zita_2=4*0.2*0.008/0.1*pow(cos(DS*pi/180),2)
96
97 #smoothing function
98
99 @smooth=1.25*exp(-pow((zCoord-0.151)/0.02,4))
100
101 @ni=(Viscosity+EddyVisc+muD)/Density
102 @phi=2*(GRAD(Vx,x)*GRAD(Vx,x) + Vx*Vx/rCoord/rCoord + \
103   GRAD(Vz,z)*GRAD(Vz,z)) + (GRAD(Vy,z)*GRAD(Vy,z) + \
104   (GRAD(Vz,x)+GRAD(Vx,z))*(GRAD(Vz,x)+GRAD(Vx,z)) + \
105   (rCoord*GRAD(Vy/rCoord,x))*(rCoord*GRAD(Vy/rCoord,x)))
106
107 #####
108 #####secondary losses rotor
109
110 @omega3Dtip=0.25*tanh(sqrt(1000*delta_r*TC))
111 @omega3Dhub=0.2*tanh(sqrt(15*fabs(beta)*180/pi*delta_r*delta_r/AR/sqrt(sigma)))
112
113
114 @fr2=omega3Dhub*exp(-pow((rCoord-0.09)/0.001,2))+ \
115 omega3Dtip*exp(-pow((rCoord-0.097)/0.001,2))
116
117
118 #source terms deterministic stress equation
119
120 @S_Drot=IF(S_Drot1<10000)(0.05*S_Drot1)
121 ELSE 1000
122
123 @S_Drot1=Fchi*exp(-2*pi/(5)*sqrt(1-pow(OMEGA*rCoord/SoundSpeed,2))*(zCoord-z_
124 #@S_Dstat=0.00005*fabs(S_curv)*exp(2*pi/(z_Sfin-z_Sini))\
125 *sqrt(1-pow(OMEGA*rCoord/SoundSpeed,2))*(zCoord-z_Sini))
126 #@S_Drot2=2*pow(10,2)*sin((zCoord-z_Rini)*2*pi/(z_Rfin-z_Rini))
127
128 @Fchi=Gchi*1/pi*(atan(distanceToWall/(z_Rfin-z_Rini)-0.0141)+pi/2)
129
130 @Gchi=S_curv
```

Appendix B

Impact of Deterministic Stresses

In this appendix we deal with the impact of the deterministic stresses modelling on the flowfield computed by the Throughflow model. The nature and the role of the deterministic stresses have been introduced in the chapter 2; they have the task to reintroduce in the axisymmetric model the effects of the circumferential inhomogeneities filtered by the averaging procedure developed by Adamczyk.

B.1 Impact of Deterministic Stresses

We would like to remind the approach proposed by Stollenwerk and Kugeler(2004) in order to model the extra terms coming out from the Adamczyk's procedure and close mathematically the system of equations. Stollenwerk and Kugeler made a Boussineq's like hypothesis to link the deterministic stress tensor to the deformation rate of the mean flow field by the introduction of an auxiliary scalar field called deterministic viscosity μ_d . Then they introduced a new transport equation for the evolution of the deterministic viscosity in the flowfield. The new transport equation has the deterministic viscosity as unknown and it is the following:

$$\bar{\rho} \frac{\partial \mu_d}{\partial t} + \bar{\rho} \tilde{u}_j \frac{\partial \mu_d}{\partial x_j} = \frac{\partial}{\partial x_j} (\mu_{tot} \frac{\partial \mu_d}{\partial x_j}) + S_{\mu_d} \quad (\text{B.1})$$

In order to understand the impact of the deterministic stresses we perform two different simulation:

- the first in which the deterministic stresses transport equation has been switched off, in our model there are two ways to switch off an equation: the first is to comment the line in which the equation is introduced, the second is to set to zero boundary, initial condition and source terms. We choose the second one.
- in the second simulation the effects of the deterministic stresses has been introduced thanks to the deterministic viscosity.

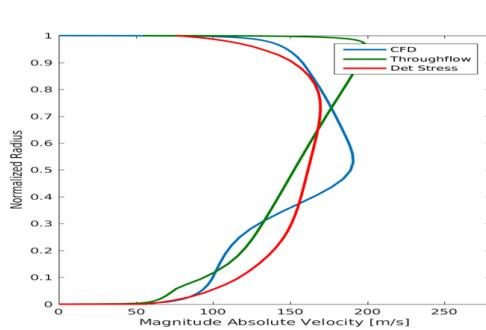


Figure B.1: Absolute Velocity

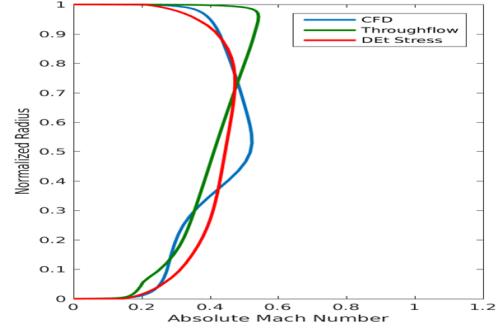


Figure B.2: Absolute Mach Number

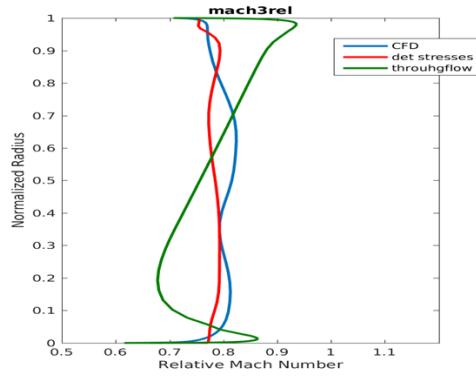


Figure B.3: Relative Mach Number

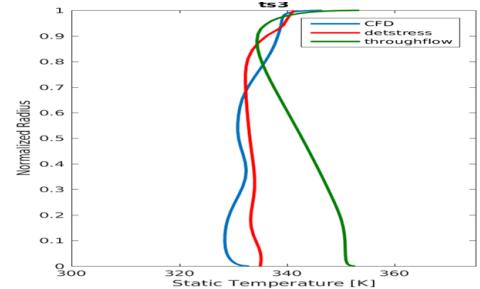


Figure B.4: Static Temperature

According to our model the deterministic stresses are due mainly to the relative rotation between the stator and rotor relative frame, for this reason they appear mainly inside and downstream of the rotor block so the effects of the deterministic stresses in the stator block are very weak (we neglect them for sake of simplicity). In the figure B1 to B6 we report the comparison between the two simulation in a radial section downstream the rotor moreover the two different throughflow simulation have been compared with the 3-D CFD simulation.

We can notice that with the introduction of the deterministic stresses the kinematic and the thermodynamic quantities show a better agreement between the throughflow caculation and the CFD calculation. Moreover, about the endwalls boundary layer we can observe that tip and hub are the regions in which the deterministic stresses are more relevant. In particular the deterministic stresses contribute with a mixing effect on the flow field that help to improve: the description of the flow field, the computation of the losses. Finally, we can conclude that the deterministic stresses are necessary to obtain a good description of the flow field inside a turbomachinery.

In order to obtain a more detailed view on the influence of the deterministic stresses in figure B7-B8 we report the radial-radial and axial-axial contribution to the momentum equations. This two component are surely the most relevant ones,

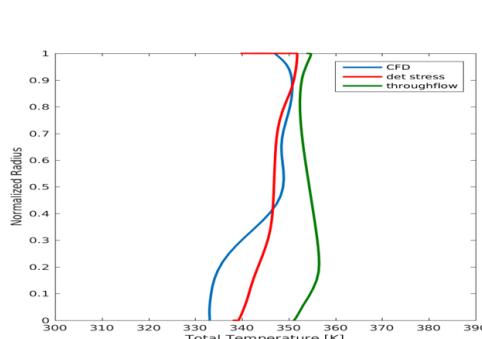


Figure B.5: Total Temperature

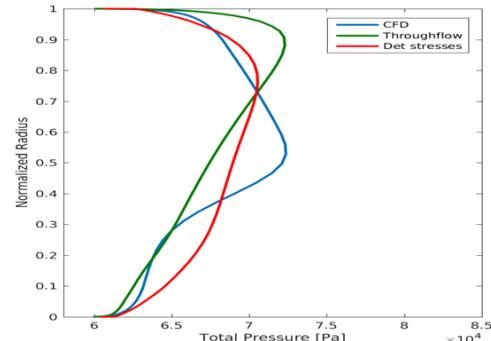


Figure B.6: Total Pressure

and they are responsible of mixing and stretching of the profile reported in figure B.1 to B.6.

$$\tau_{rr}^d = \overline{\rho u_r'' u_r''} \quad (\text{B.2})$$

$$\tau_{zz}^d = \overline{\rho u_z'' u_z''} \quad (\text{B.3})$$

From the figure B.7 B.8 we can observe that the two components of the deterministic stress tensor we are analyzing have maximum magnitude inside and at the end of rotor row. On the other hand this two component are very close to zero in the stator row; to neglect the effect of the deterministic stresses inside the stator row is an approximation necessary for a simpler analytic computation of the source terms, this will be better exploited in app.C. Looking at figure B.7 and B.8 we can notice that the absolute value of τ_{rr} and τ_{zz} in maximum inside the rotor block and in particular at the endwalls; this is physically correct from a qualitative point of view in fact these are the region in which the effect of mixing is more relevant. How these two components of the deterministic stresses' tensor affect the flowfield can be easily seen in figure B.1 B.2 in which is shown the trend of the absolute velocity and absolute Mach number. From a qualitative point of view we can notice that the presence of the deterministic stress tensor creates a radial mixing of the flow along the radial direction, figure B.7-B.8 show that τ_{rr} is negative both at the hub and tip. The effects of the deterministic stresses on the losses can be seen in figure B.5-B.6 in which a radial section of the Total Temperature and Total Pressure are presented. From figure B.6 we can observe that at the tip the presence of the deterministic stresses helps the model to reach a better agreement with the CFD calculation introducing losses that the simple throughflow model didn't catch. On the other hand we can see that at the mid-line extra losses are introduced, this is probably due to the negative value of τ_{zz} at midline (figure B.6) this could be a non physical effect due to the simplicity of our model. Finally, thanks to the figure B.3 B.4 and B.5 we can observe that the introduction of the deterministic stresses has a good influence of the relative Mach number static temperature and static pressure at the rotor outlet.

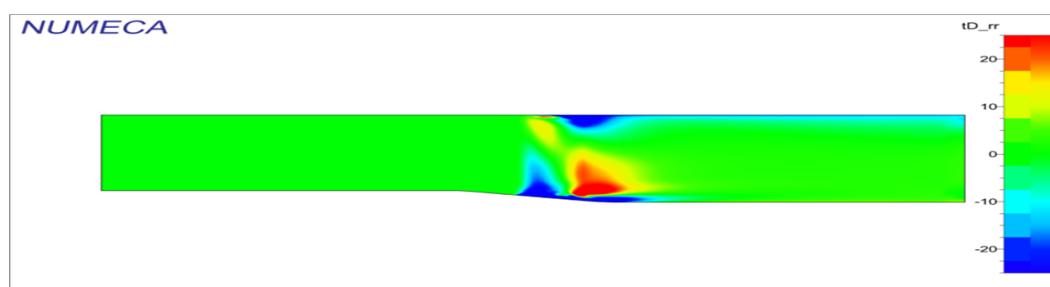


Figure B.7: Radial-Radial Stress

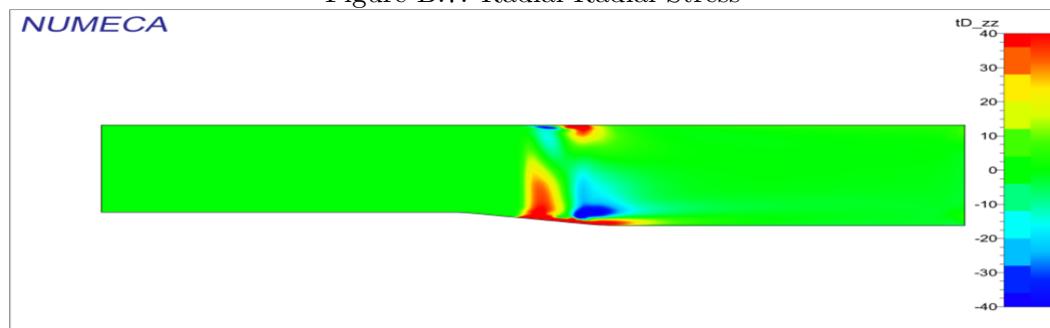


Figure B.8: Axial-Axial Stress

Appendix C

Computation of the source term in the deterministic stress transport equation

C.1 Deterministic Viscosity Closure

In this appendix we take a brief look into the idea and the theoretical aspects at the bases of the computation of the source terms appearing in the deterministic stresses transport equation. The basic idea of this work is to describe the additional terms, due to the deterministic part of the flow, thanks the introduction of the deterministic viscosity and then develop a transport model for the evolution of the deterministic viscosity itself. Let's call back the definition of deterministic viscosity as that quantities that links the deterministic stresses with the strain tensor of the mean flow:

$$\mu_d = \frac{\left(\frac{\partial x_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)}{\rho u''_i u''_j} \quad (\text{C.1})$$

and its transport equation:

$$\bar{\rho} \frac{\partial \mu_d}{\partial t} + \bar{\rho} \tilde{u}_j \frac{\partial \mu_d}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\mu_{tot} \frac{\partial \mu_d}{\partial x_j} \right) = S_{\mu_d} \quad (\text{C.2})$$

We can easily notice that at the left hand side of the equation there are the mathematical operators that describe the transport and the diffusion of the deterministic viscosity while at the right hand side there is the source of deterministic viscosity.

C.2 Source Term Computation

The source term in the equation C.2 is composed of various contribution; try to describe all of these contribution is a tremendous task and it is far beyond of the

purpose of this model. However in this work we propose a simple analytical model in order to catch the potential effect of the blade row interaction.

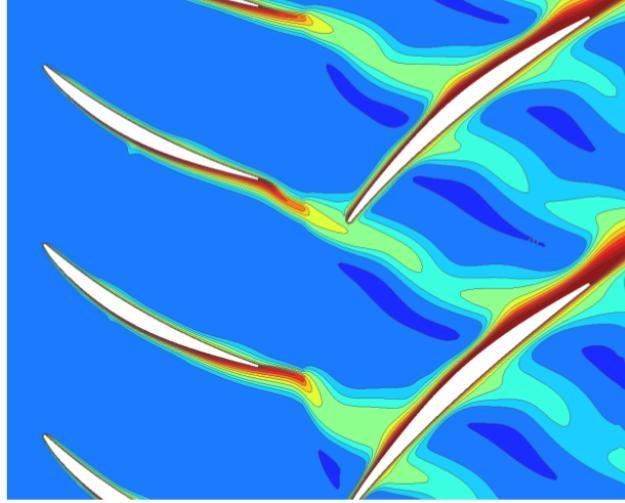


Figure C.1: Example of the blade row interaction

The basic idea of this model is to suppose that the total pressure downstream a blade row could be described by the product of a wave like function and a function of the axial coordinate:

$$\overline{p_T(x, \theta, t)} = \bar{p}_0 e^{i \frac{2\pi r}{L} (\theta - \omega t)} f(x) \quad (\text{C.3})$$

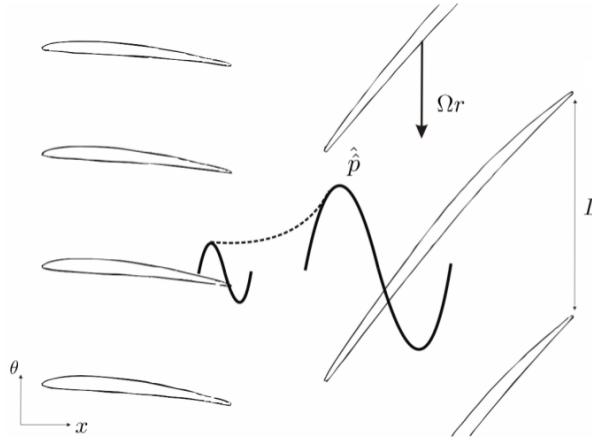


Figure C.2: Basic Idea of the model

This pressure's profile is propagated downstream according to the waves' equation:

$$(\partial_t^2 - a^2(\partial_x^2 + \frac{1}{r^2}\partial_\theta^2))\bar{p}_T(x, \theta, t) = 0 \quad (\text{C.4})$$

Taking the derivative of the expression C.3 and thanks to the properties of the exponential function

$$\partial_t^2 \bar{p}_T(x, \theta, t) = -\bar{p}_0 \left(\frac{2\pi r}{L}\right)^2 e^{i\frac{2\pi r}{L}(\theta-\omega t)} f(x) \quad (\text{C.5})$$

$$\partial_x^2 \bar{p}_T(x, \theta, t) = \bar{p}_0 e^{i\frac{2\pi r}{L}(\theta-\omega t)} f(x)'' \quad (\text{C.6})$$

$$\partial_\theta^2 \bar{p}_T(x, \theta, t) = -\bar{p}_0 \left(\frac{2\pi r}{L}\right)^2 e^{i\frac{2\pi r}{L}(\theta-\omega t)} f(x) \quad (\text{C.7})$$

So we reduce the PDE C.4 to an ODE with $f(x)$ as unknown:

$$f(x)'' - \alpha^2 f(x) = 0 \quad (\text{C.8})$$

with

$$\alpha = \left(\frac{2\pi}{L}\right)^2 \left(1 - \left(\frac{\omega r}{a}\right)^2\right) \quad (\text{C.9})$$

The general solution is :

$$f(x) = c_1 e^{\alpha x} + c_2 e^{-\alpha x} \quad (\text{C.10})$$

In order to fix the constant we have to impose two boundary conditions :

$$\begin{cases} \lim_{x \rightarrow \infty} f(x) = 0 \\ f(x = x_{LE}) = \bar{p}_{max} \end{cases} \quad (\text{C.11})$$

At the end we get the solution for $f(x)$

$$f(x) = \bar{p}_{max} e^{-\frac{2\pi}{L} \sqrt{1 - \left(\frac{\omega r}{a}\right)^2} (x - x_{LE})} \quad (\text{C.12})$$

Knowing the solution for $f(x)$ is straightforward compose the two terms and get the complete solution. This solution is rigorously valid only inside the rotor block; because the angular velocity ω is defined only inside it. For this reason the source term at the RHS of the equation C.2 is different from zero only inside the rotor, this is obviously an approximation but allow us to approximate a very complex physical phenomena in a quite simple way with acceptable agreement with more accurate CFD computation as shown in appendix B.