

**National Technical University of Athens**  
**School of Mechanical Engineering**  
**Department of Fluid Mechanics**  
**Laboratory of Thermal Turbomachinery**  
**Parallel CFD & Optimization Unit**

**Integration of the GPU-Enabled CFD Solver PUMA into  
the Workflow of a Turbomachinery Industry. Testing and  
Validation.**

Diploma Thesis

**Ilias Ioannis Tsopelas**

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, September 2018



## Acknowledgements

First and foremost, I would like to express my deep gratitude to my professor Kyr-  
iakos C. Giannakoglou for not only guiding me during the process of completing  
this diploma thesis, but also for presenting me with the unique and wonderful op-  
portunity of working on my thesis in a real industrial environment, Andritz Hydro.  
His advice and guidance helped me a lot in composing this thesis. Of course, I  
want to thank all the members of the Parallel CFD & Optimization Unit of NTUA,  
especially Dr. Varvara Asouti and Dr. Xenofon Trompoukis that helped me with  
all the questions I had when nothing seemed to be working properly.

I would also like to thank everyone working at Andritz Hydro for welcoming and  
helping me learn the ropes around the office. Specifically, I would like to thank my  
supervisor Dr. Evgenia Kontoleontos and co-worker Konstantinos Ioakeimidis for  
providing much needed counsel, but most importantly for being there when things  
got lonely. In addition, I would like to express my deep gratitude to my managers  
Peter Grafenberger and Simon Weissenberger for providing me with the opportunity  
of working in such a friendly environment.

Finally, I would like to thank my family and friends for being there for me during  
the composition of this diploma thesis, but also throughout all these long years that  
I studied at my university NTUA. Thank you for enduring all of my complaints and  
supporting me throughout my studies.



**National Technical University of Athens**  
**School of Mechanical Engineering**  
**Department of Fluid Mechanics**  
**Laboratory of Thermal Turbomachinery**  
**Parallel CFD & Optimization Unit**

## **Integration of the GPU-Enabled CFD Solver PUMA into the Workflow of a Turbomachinery Industry. Testing and Validation.**

Diploma Thesis

**Ilias Ioannis Tsopelas**

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, September 2018

### **Abstract**

This diploma thesis presents the integration and validation of the PUMA CFD solver developed by the Parallel CFD & Optimization Unit (PCOpt/NTUA) of the School of Mechanical Engineering of the National Technical University of Athens into the industrial workflow of Andritz Hydro, at the premises of which a major part of this thesis was conducted.

The PUMA CFD solver can run on clusters of GPUs, thus achieving lower computational costs. PUMA uses a vertex-centered finite volume method for solving the 3D Reynolds-Averaged Navier-Stokes equations with the use of the Spalart-Allmaras turbulence model. PUMA, also, employs the pseudocompressibility method to accelerate the convergence rate.

Various interfacing tools are programmed that enable the organic integration of the PUMA solver into the workflow. These tools include two grid converters, a pre-processor and a post-processor.

The PUMA solver is, then, compared with a commercial CFD solver. The comparison is based on various cases of hydraulic turbine configurations. With each new case the complexity of the configuration rises, thus testing the capabilities of the PUMA solver. The cases include the runner of a Francis turbine, the guide vanes and the runner of a propeller turbine and the guide vanes, the runner and the draft tube of a Kaplan turbine.

Finally, the PUMA solver is used as an evaluation tool in the framework of the design-optimization loop used by Andritz Hydro. The Evolutionary Algorithm optimization method is chosen, employed by the tool Evolutionary Algorithm SYstem (EASY), developed by PCOpt/NTUA. The shape of the blade of a Kaplan runner is optimized with the goal of maximizing the turbine's efficiency and avoiding the appearance of cavitation.



## Acronyms

NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization Unit
GPU	Graphics Processing Unit
CPU	Central Processing Unit
CFD	Computational Fluid Dynamics
PDE	Partial Differential Equation
RANS	Reynolds-Averaged Navier-Stokes
S-A	Spalart-Allmaras
SPA	Single Precision Arithmetics
DPA	Double Precision Arithmetics



# Contents

Contents	i
<b>I Thesis in English</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Hydraulic Turbines . . . . .	2
1.2 CFD-Based Analysis and Integration Process . . . . .	4
1.3 Design and Optimization of Hydraulic Turbines . . . . .	5
1.4 Thesis Outline . . . . .	6
<b>2 The GPU-enabled CFD solver</b>	<b>7</b>
2.1 Governing Equations . . . . .	7
2.2 The Spalart-Allmaras (S-A) Turbulence Model . . . . .	9
2.3 Boundary Conditions for Hydraulic Turbines . . . . .	10
2.3.1 Wall Boundary Conditions . . . . .	10
2.3.2 Inlet Boundary Conditions . . . . .	11
2.3.3 Outlet Boundary Conditions . . . . .	11
2.3.4 Periodic Boundary Conditions . . . . .	11
2.4 Wall Functions . . . . .	12
2.5 Rotor-Stator Interaction (RSI) . . . . .	13
2.6 Programming on GPUs . . . . .	15
<b>3 Interfaces for the Integration of PUMA</b>	<b>17</b>
3.1 Grid Converters . . . . .	17
3.1.1 The CGNS Format . . . . .	18
3.1.2 The LTT Format . . . . .	18
3.1.3 The Tecplot Format . . . . .	21
3.1.4 The cgnsToLtt Converter . . . . .	22
3.1.5 The pltToCgns converter . . . . .	23
3.2 Pre-processor for the PUMA Solver . . . . .	24
3.2.1 The Imposed Discharge Condition . . . . .	25
3.2.2 The Imposed Head Condition . . . . .	25
3.3 The set-up-run-gpu Shell Script . . . . .	26
3.4 Post-processor Functions . . . . .	27

3.4.1	Mass Flow Related Functions . . . . .	28
3.4.2	Torque Computation Function . . . . .	29
<b>4</b>	<b>Case Studies and Comparison</b>	<b>33</b>
4.1	Francis Runner . . . . .	36
4.2	Propeller Guide Vane - Runner . . . . .	41
4.3	Kaplan Guide Vane - Runner - Draft Tube . . . . .	47
4.4	Conclusions from the Comparison . . . . .	55
4.5	Investigation of the Artificial Compressibility Parameter $\beta$ . . . . .	56
<b>5</b>	<b>Optimization of a Kaplan Runner Blade</b>	<b>61</b>
5.1	Evolutionary Algorithms . . . . .	61
5.2	Design Variables and Blade Parameterization . . . . .	63
5.3	Objective Functions . . . . .	66
5.3.1	Efficiency . . . . .	66
5.3.2	Cavitation . . . . .	66
5.4	Constraints . . . . .	67
5.5	Optimization Results . . . . .	68
<b>6</b>	<b>Overview and Conclusions - Future Work Ideas</b>	<b>73</b>
6.1	Overview and Conclusions . . . . .	73
6.2	Proposals for Future Work . . . . .	74
<b>II</b>	<b>Thesis in Greek</b>	<b>75</b>
<b>7</b>	<b>Εκτενής Περίληψη στα Ελληνικά</b>	<b>1</b>
7.1	Εισαγωγή . . . . .	1
7.2	Ο Επιλύτης Ροής PUMA . . . . .	1
7.3	Εργαλεία Υποβοήθησης της Ένταξης του Επιλύτη PUMA . . . . .	2
7.3.1	Μετατροπείς Πλεγμάτων . . . . .	2
7.3.2	Ο Προεπεξεργαστής για τον Επιλύτη PUMA . . . . .	7
7.3.3	Το εργαλείο set-up-run-gru . . . . .	8
7.3.4	Μετεπεξεργαστικές υπορουτίνες . . . . .	8
7.4	Βιομηχανικές Εφαρμογές και Σύγκριση . . . . .	9
7.4.1	Συμπεράσματα από τη Σύγκριση . . . . .	14
7.4.2	Παραμετρική Μελέτη της Ψευδοσυμπιεστότητας $\beta$ . . . . .	17
7.5	Βελτιστοποίηση Πτερυγίου Δρομέα Υδροστροβίλου Kaplan . . . . .	19
7.5.1	Μεταβλητές Σχεδιασμού . . . . .	20
7.5.2	Αντικειμενικές Συναρτήσεις . . . . .	21
7.5.3	Περιορισμοί . . . . .	22
7.5.4	Αποτελέσματα της Βελτιστοποίησης . . . . .	22
7.6	Σύνοψη και Συμπεράσματα - Ιδέες για Μελλοντική Δουλειά . . . . .	23
7.6.1	Σύνοψη και Συμπεράσματα . . . . .	23
7.6.2	Προτάσεις για Μελλοντική Δουλειά . . . . .	24





# Part I

## Thesis in English



# Chapter 1

## Introduction

This diploma thesis is concerned with the integration and validation of the GPU-enabled flow solver, developed by the PCOpt/NTUA, into the Andritz Hydro design-optimization workflow for hydraulic turbines. The integration in the already existing toolchain of the industrial analysis workflow, requires the programming of pre- and post-processors, interfacing tools as well as extensive validation on problems-cases provided by Andritz Hydro, at the premises of which (at Linz, Austria) a major part of this work has been conducted.

The goal of this integration is to be as little intrusive as possible to the existing toolchain and achieve a unified toolchain that enables the user to interchange smoothly between various CFD solvers (although this work focuses exclusively on the flow solver programmed by PCOpt/NTUA). Furthermore, the approach to developing the various and necessary tools is always with the mind of a full automatization of the process, due to the fact that any user of this tool is not required to have specific knowledge or need to interact with the inner workings of the tool. Consequently, the final product is fully automated and user friendly and does not require any specific programming knowledge by the user.

The next step, after the integration in the unified toolchain, is to include this new toolchain in the optimization loop used in the industrial workflow. Until now, for the optimization, another software solving the Euler equations, for incompressible fluids, was used due to its low cost which is desirable in industrial applications. But, with the addition of the GPU-enabled flow solver, it was possible to accomplish comparable solving time, while simultaneously, achieving much more accurate results during the flow analysis, since the Reynolds-Averaged Navier-Stokes (RANS) equations, combined with the Spalart-Allmaras (S-A) turbulence model, are now being solved.

## 1.1 Hydraulic Turbines

Hydraulic turbines are rotary machines that convert kinetic energy and potential energy of water into mechanical work, that can then be transformed into electric power, thus providing an efficient and renewable source of energy. The hydraulic turbine harnesses the power of the water passing through its blades, which is expressed as

$$P_{in} = \rho g H Q \quad (1.1)$$

where  $\rho$  is the water's density,  $g$  is the gravitational acceleration (in  $\frac{m}{s^2}$ ),  $H$  is the hydraulic head of the hydraulic turbine (in  $m$ ), that is defined as the drop in total pressure  $p_t$  from the inlet to the outlet of the turbine, and  $Q$  (in  $\frac{m^3}{s}$ ) is the volumetric discharge. As output, the hydraulic turbine provides mechanical work for the generator, that is expressed as

$$P_{out} = T\omega \quad (1.2)$$

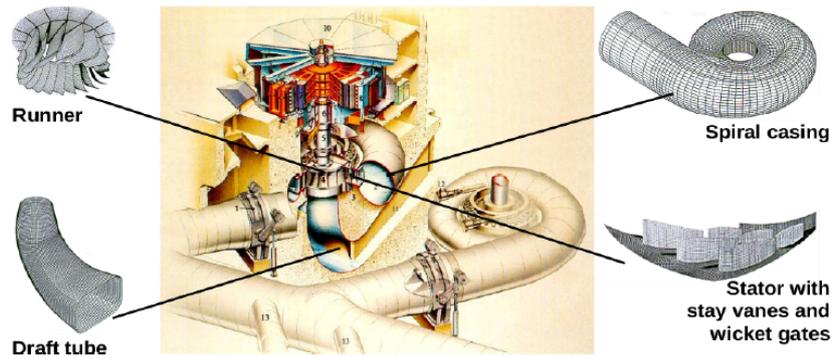
where  $T$  is the torque produced on the shaft (in  $Nm$ ) and  $\omega$  is the angular velocity of the machine (in  $\frac{rad}{s}$ ).

As a result the efficiency of the hydraulic turbine is defined as

$$\eta = \frac{P_{out}}{P_{in}} \quad (1.3)$$

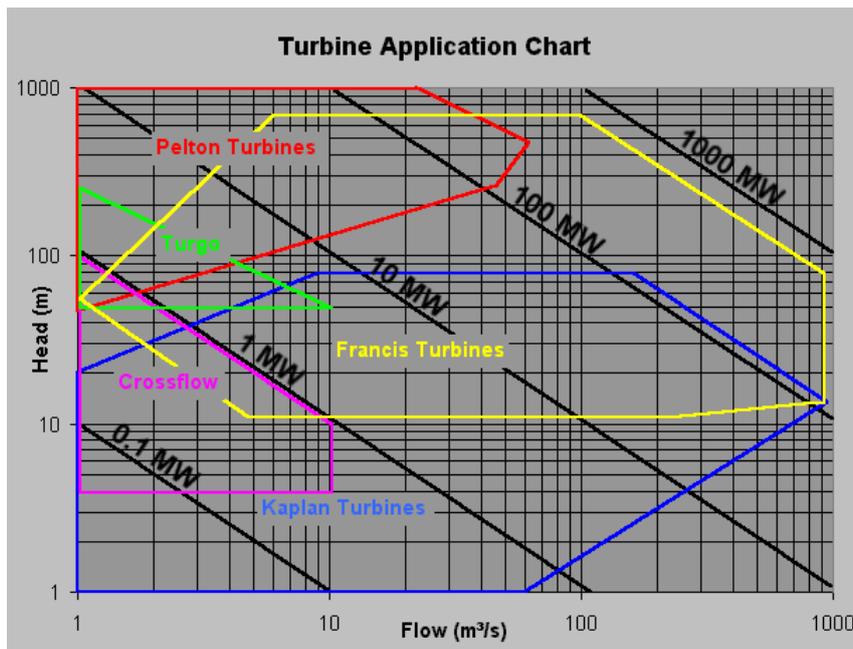
which describes how efficiently the machine converts the fluid's energy into mechanical work.

The basic components that comprise a hydraulic turbine are (a) the spiral casing that directs the flow from the turbine's inlet towards the runner, (b) the stay vanes and (c) guide vanes (or wicket gates) that are stationary blades which help guide the flow and provide an smooth entrance to the runner, (d) the runner which is the set of rotating blades that convert the water's energy into mechanical work and, finally, (e) the draft tube that is responsible for guiding the flow to the outlet of the machine.



**Figure 1.1:** *Typical configuration of hydraulic turbine's components [1].*

There are many types of hydraulic turbines depending on the hydraulic head ( $H$ ) and volumetric discharge ( $Q$ ) that are required to handle. In figure 1.2, typical values for  $H$  and  $Q$  are presented for the various types of hydraulic turbines.



**Figure 1.2:** *Chart with values of hydraulic head  $H$  and discharge  $Q$  for the various types of hydraulic turbines [2].*

The most widely used types of hydraulic turbines are the Francis and Kaplan turbines, which will also serve as the main focus of this diploma thesis. On one hand, the Francis turbines are radial turbines, meaning that the flow enters the turbine's runner radially and exits axially and can handle higher values of hydraulic head  $H$ . On the other hand the Kaplan turbines are axial turbines, meaning that the flow enters and exits the machine's runner axially and generally operate on the lower-end

of the hydraulic head spectrum of values. In figure 1.3, a typical configuration of a Kaplan and a Francis hydraulic turbine is presented.



**Figure 1.3:** *Typical designs of Kaplan (left) and Francis (right) turbine runners. The specific models shown are designed by Andritz Hydro.*

## 1.2 CFD-Based Analysis and Integration Process

For the past decades, the development in the field of CFD has been tremendous. The use of CFD for predicting fluid flows has been a powerful tool for the industry, in general. That is because it enables the faster analysis of the problem at hand, in the present case the flow inside a hydraulic turbine, and a more cost-efficient way of designing and optimizing the final product.

Of course, in order to tackle problems that are highly demanding in computational cost in an economically viable and efficient way for the industry, such as the design of a hydraulic turbine, the resources that are provided, namely the computational power available, have to be used wisely. An effective way to achieve this is to enter the field of parallel CFD computing, which means to use several processing units (namely CPUs and GPUs) to perform tasks simultaneously. As a result, the computational cost reduces extensively, thus making the use of CFD-based analysis a viable option.

When obtaining a new CFD software, the main obstacle that arises is the proper and smooth integration of said software inside the existing collaborative system of tools and processes (tool-chain). This integration must take place keeping in mind that the whole system should undergo the least possible modifications. This constraint is the result of the fact that all the users are not required to and is not necessary to know the inner workings of the software.

That being said, it is important to create and develop the necessary interfacing tools, peripheral to the main software, that enable the seamless integration of the software in the existing toolchain of the company.

In the present case, the main software is the GPU-enabled CFD solver developed by PCOpt/NTUA and used by Andritz Hydro. This CFD solver implements the RANS equations for incompressible fluid flows in their conservative form, coupled with the Spalart-Allmaras turbulence model, using a vertex-centered Finite Volume Method for discretizing them. The main advantage of this solver is the low computational cost. This advantage is the result of the use of clusters of GPUs.

The use of GPUs for performing costly computations has significantly increased over the past few years. This is because the GPUs are, in essence, shared memory parallel co-processors and, precisely, this characteristic gives them the advantage of performing very computationally demanding simulations very fast. Of course, the trade-off for this advantage is the programming skills that are required in order to develop a CFD solver that can run on GPUs efficiently.

Finally, the tools programmed in the present diploma thesis perform all the necessary tasks, during the pre- and post-processing phases of the flow analysis, in a manner that lets them run undetected, thus creating a fully automated system that complies to the rules of the overall workflow of the industry (Andritz Hydro).

### **1.3 Design and Optimization of Hydraulic Turbines**

The turbine optimization is an integral part of the industrial workflow. The optimization in Andritz Hydro is carried out by using the method of Evolutionary Algorithms, which is a stochastic population-based method. The optimization tool used is the Evolutionary Algorithm SYstem or EASY, developed by PCOpt/NTUA.

The optimization process is very important in the industrial environment because it enables the design and manufacture of much more efficient turbines. This, also, results in a much more environmentally friendly turbine design that harnesses efficiently the energy of water.

The task of designing a new hydraulic turbine encompasses some basic steps. First of all, previous knowledge and experience from existing designs is used as a starting point for the new project. Models of already designed and working turbines that operate under similar conditions provide a good basis for designing and manufacturing a new model, as opposed to starting the design from scratch.

Each component of the turbine is designed separately, in general. In some cases, components that are already designed and optimized are directly used in the new hydraulic turbine. The most important part of the turbine is the runner blades. These go through the process of optimization in order to achieve a design that best fits the conditions and requirements of the project.

Continuing, after all the necessary CFD simulations are completed, the first model

of the new turbine is manufactured. This model is tested in the lab in order to validate the predictions made from the simulations. If the results are not satisfying, then the design process is repeated until the goal is achieved.

Finally, when the lab testing process is completed, the final hydraulic turbine is manufactured and starts operating.

## 1.4 Thesis Outline

In this section, the general outline of this diploma thesis is presented.

- **Chapter 2** A brief description of the GPU-enabled CFD solver that is used in this diploma thesis. The governing equations, the various boundary conditions and the pseudocompressibility method, employed by the PUMA solver, are presented. Also, some information concerning the programming techniques for GPUs are included.
- **Chapter 3** The various interfacing tools that are programmed in this diploma thesis for the integration of the GPU-enabled CFD solver are presented extensively.
- **Chapter 4** Test cases involving various hydraulic turbine configurations are presented in this chapter. In order to validate the GPU solver, the results are compared to a commercial software. Also an investigation on the pseudocompressibility parameter  $\beta$  is included.
- **Chapter 5** The incorporation of the GPU solver in the optimization loop using Evolutionary Algorithms is presented, along with an optimization test case.

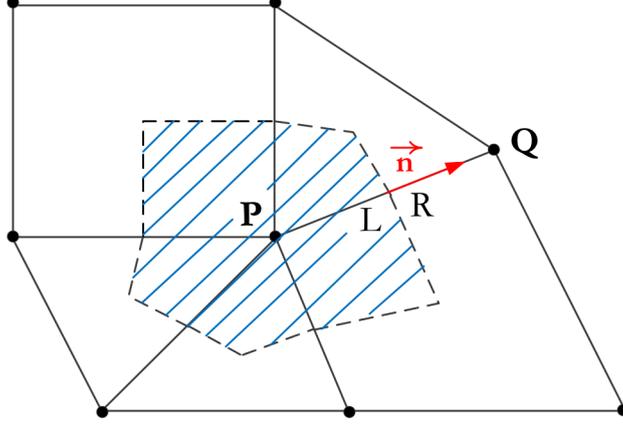
# Chapter 2

## The GPU-enabled CFD solver

For the flow analysis of the hydraulic turbines the GPU-enabled CFD solver is used, developed by PCOpt/NTUA, also known as the PUMA (Parallel Unstructured Multi-row Adjoint) solver. The PUMA solver has been the focus of already 3 PhD theses ([3], [4], [5]), and was based on an already existing in-house CFD solver running on CPUs. The PUMA solver solves numerically the Navier-Stokes equations using the finite volume method with vertex centered volumes and storage, providing second order accuracy, for incompressible fluid flows. The main advantage of this solver is the fact that it runs on clusters of GPUs, reducing the computational cost to a great extent.

### 2.1 Governing Equations

The PUMA solver employs a vertex centered finite volume method for numerically integrating the 3D RANS equations for an incompressible fluid flow, expressed in their conservative form. In the vertex-centered variant of the finite volume method, a finite volume is defined around each grid node and the flow variables are stored on the grid node, as opposed to the cell centered variation in which the finite volume coincides with the cell. Figure 2.1 presents the finite volume formed around a node in 2D grid consisting of triangular and quadrilateral elements. The finite volumes are formed by connecting the mid-edges and element barycenters surrounding each grid node.



**Figure 2.1:** Graphical representation of vertex centered finite volume.

The steady-state RANS equations in their conservative form for incompressible fluid flows, written in a non-inertial (rotating) frame of reference and using the artificial compressibility method [6], are expressed as

$$\Gamma_{nm}^{-1} \frac{\partial U_m}{\partial \tau} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} + S_n = 0 \quad (2.1)$$

where  $\tau$  is the pseudo-time step,  $U_m = [\frac{p}{\rho} \quad u_1^A \quad u_2^A \quad u_3^A]$  stands for the vector of flow variables, with  $\frac{p}{\rho}$  being the kinematic pressure (since  $\rho$  is assumed constant in incompressible fluid flows) and  $u_m^A (m = 1, 2, 3)$  the velocity components in the absolute frame of reference. The inviscid  $f_{nk}^{inv}$  and viscous  $f_{nk}^{vis}$  fluxes and the source terms  $S_n$  are defined as:

$$f_{nk}^{inv} = \begin{bmatrix} u_k^R \\ u_1^A u_k^R + p \delta_{1k} \\ u_2^A u_k^R + p \delta_{2k} \\ u_3^A u_k^R + p \delta_{3k} \end{bmatrix} \quad f_{nk}^{vis} = \begin{bmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \end{bmatrix} \quad S_n = \begin{bmatrix} 0 \\ \epsilon_{1lk} \omega_l u_k^A \\ \epsilon_{2lk} \omega_l u_k^A \\ \epsilon_{3lk} \omega_l u_k^A \end{bmatrix} \quad (2.2)$$

where  $u_m^R (m = 1, 2, 3)$  are the cartesian velocity components expressed in the relative reference frame and are linked to the absolute cartesian velocity components by the formula  $u_m^A = u_m^R + u_m^F$ , with  $u_m^F = \epsilon_{mlk} \omega_k (x_l - x_l^C)$  being the rotating/non-inertial frame velocity. The  $(k, m)$  component of the viscous and Reynolds stress tensor ( $\tau_{km}$ ) is defined as:

$$\tau_{km} = (\nu + \nu_t) \left( \frac{\partial u_k^A}{\partial x_m} + \frac{\partial u_m^A}{\partial x_k} \right) \quad (2.3)$$

where  $\nu$  and  $\nu_t$  are, respectively, the kinematic viscosity and the kinematic eddy viscosity.

The source term  $S_n$  corresponds to the Coriolis force, that arises from the formulation of eqs 2.1 in a non-inertial (rotating) frame of reference. Specifically,  $\omega_k$  are the components of the rotational speed of the turbine and the symbol  $\epsilon_{mlk}$  is the Levi Civita symbol that is defined as

$$\epsilon_{mlk} = \begin{cases} +1 & , \text{if } (m, l, k) \text{ is } (1, 2, 3) \text{ or } (2, 3, 1) \text{ or } (3, 1, 2) \\ -1 & , \text{if } (m, l, k) \text{ is } (3, 2, 1) \text{ or } (1, 3, 2) \text{ or } (2, 1, 3) \\ 0 & , \text{if } m = l \text{ or } l = k \text{ or } k = m \end{cases}$$

Finally, the matrix  $\Gamma_{nm}^{-1}$ , as introduced by Turkel [7], is defined as:

$$\Gamma_{nm}^{-1} = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 \\ \frac{u_1^A + \alpha u_1^R}{\beta^2} & 1 & 0 & 0 \\ \frac{u_2^A + \alpha u_2^R}{\beta^2} & 0 & 1 & 0 \\ \frac{u_3^A + \alpha u_3^R}{\beta^2} & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

where  $\alpha$  is a predefined parameter, while  $\beta$  is equivalent to an artificial speed of sound. The pseudo-compressibility method, introduced by Chorin [6], enables the use of time-marching techniques for solving the PDEs. Practically,  $\Gamma_{nm}^{-1}$  acts as a preconditioning matrix for the generalized artificial compressibility equations 2.1. To ensure a fast convergence rate and robustness, the  $\beta$  parameter must be selected carefully, so as to get a condition number as close to unity as possible, thus leading to a stable system of PDEs.

## 2.2 The Spalart-Allmaras (S-A) Turbulence Model

For the modeling of turbulence, the GPU solver employs the S-A turbulence model.

In the S-A model [8], the kinematic eddy viscosity  $\nu_t$  is given by:

$$\nu_t = \tilde{\nu} f_{v1} \quad (2.5)$$

after solving an additional PDE for the turbulence variable  $\tilde{\nu}$ , which is the following:

$$\frac{\partial \tilde{\nu}}{\partial \tau} + \frac{\partial (u_k^R \tilde{\nu})}{\partial x_k} - \frac{1}{\sigma} \left\{ \frac{\partial}{\partial x_k} \left[ (\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_k} \right] + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \right\} - c_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} + \left( c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t1} \right) \left( \frac{\tilde{\nu}}{\Delta} \right)^2 = 0 \quad (2.6)$$

where  $\Delta$  is the distance of the internal nodes from the closest wall boundary. Eq. 2.6 is closed by the following relations and constants:

$$\begin{aligned}\chi &= \frac{\tilde{\nu}}{\nu}, & f_{v_1} &= \frac{\chi^3}{\chi^3 + c_{v_1}^3}, & f_{v_2} &= 1 - \frac{\chi}{1 + \chi f_{v_1}}, & S &= \sqrt{\epsilon_{klm} \epsilon_{kqr} \frac{\partial u_m^A}{\partial x_l} \frac{\partial u_r^A}{\partial x_q}}, \\ \tilde{S} &= S + \frac{\tilde{\nu} f_{v_2}}{\kappa^2 \Delta^2}, & f_w &= g \left( \frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6} \right)^{\frac{1}{6}}, & g &= r + c_{w_2} (r^6 - r), \\ r &= \min \left( 10, \frac{\tilde{\nu}}{\tilde{S} \kappa^2 \Delta^2} \right), & \mu_t &= \rho \tilde{\nu} f_{v_1}, & \tilde{\mu} &= \rho \tilde{\nu}, & f_{t_2} &= c_{t_3} e^{-c_{t_4} \chi^2}, \\ c_{v_1} &= 7.1, & c_{b_1} &= 0.1355, & c_{b_2} &= 0.622, & c_{w_1} &= \frac{c_{b_1}}{\kappa^2} + \frac{1 + c_{b_2}}{\sigma}, \\ c_{w_2} &= 0.3, & c_{w_3} &= 0.3, & \sigma &= \frac{2}{3}, & \kappa &= 0.41, & c_{t_3} &= 1.2, & c_{t_4} &= 0.5\end{aligned}$$

## 2.3 Boundary Conditions for Hydraulic Turbines

In order to define properly the flow problem, equations 2.1 must be accompanied by a set of appropriate boundary conditions. These conditions vary, based on the physical boundary they are being imposed on. Mainly, there are four kinds of boundary conditions, one set for the solid walls, one set for the inlet, one for the outlet of a hydraulic turbine and one set for periodic boundaries.

### 2.3.1 Wall Boundary Conditions

The wall boundary conditions can further be categorized in two categories based on the type of flow, inviscid or viscous. For inviscid flows, the no-penetration condition applies, where the normal component of the velocity relative to the wall is set equal to the normal component of the wall velocity. This is expressed as

$$u_k^A n_k = u_k^W n_k \quad (2.7)$$

where  $u_k^W$ , ( $k = 1, 2, 3$ ) is the wall velocity. For viscous flows, the no-sip condition applies, where the absolute velocity is set equal to the wall velocity, namely

$$u_k^A = u_k^W, k = 1, 2, 3 \quad (2.8)$$

In addition, the turbulence variable  $\tilde{\nu}$  is set to zero, i.e.  $\tilde{\nu} = 0$ .

### 2.3.2 Inlet Boundary Conditions

At the inlet boundary of a 3D domain, three quantities must be defined while one is extrapolated from the flow domain. The two of the three quantities are the two angles  $\theta_1, \theta_2$  that link the vectors of the cylindrical velocity components and the velocity magnitude  $|u_l^A|$ . This link is expressed as

$$\begin{aligned}u_1^A &= |u_l^A| \sin \theta_1 \\u_2^A &= |u_l^A| \cos \theta_1 \sin \theta_2 \\u_3^A &= |u_l^A| \cos \theta_1 \cos \theta_2\end{aligned}\tag{2.9}$$

The third quantity can either be the absolute velocity magnitude at the inlet  $|u_l^A|$  or the total pressure  $p_t$ . The quantity that is extrapolated from the flow domain is the static pressure. In the case of specifying the total pressure at the inlet, the velocity magnitude is then computed based on the formula

$$p_t = p + \frac{1}{2} u_l^A u_l^A\tag{2.10}$$

As for the turbulence, the ratio  $\frac{\nu_t}{\nu}^{IN}$  is set at the boundary (in the absence of exact data), with typical values of  $10 \div 20$ .

### 2.3.3 Outlet Boundary Conditions

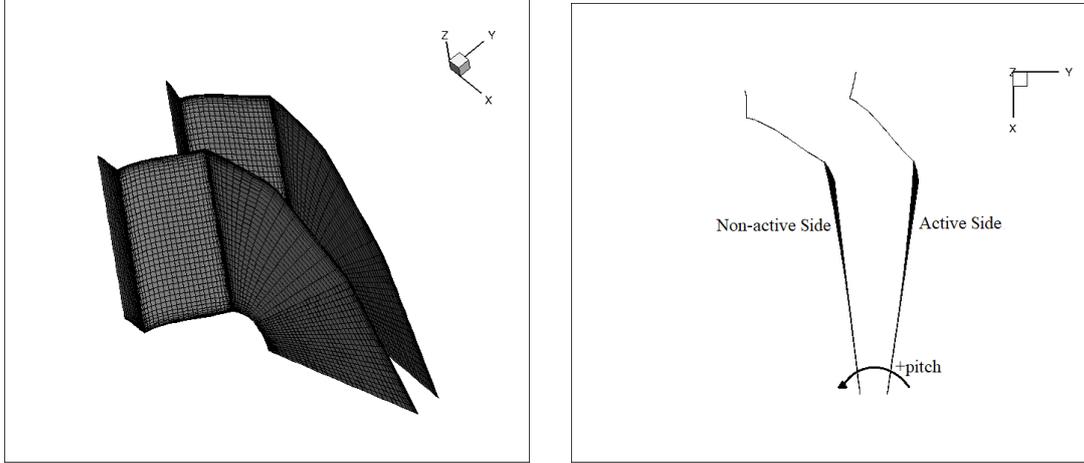
Conversely, at the outlet boundary one flow quantity must be defined and three are extrapolated from the flow domain. The imposed quantity usually is a fixed value for the static pressure  $p_s$ , although there are other options available, such as imposing a fixed value for the average static pressure  $\overline{p_s}$  over the whole outlet or, alternatively, imposing the flow rate.

It is important to state that care must be taken when combining the types of boundary conditions imposed at the inlet and outlet boundaries. For example, when the velocity magnitude and the flow angles are imposed at the inlet, then the flow rate cannot be imposed at the outlet.

### 2.3.4 Periodic Boundary Conditions

In the cases of hydraulic turbines examined in this thesis, there are mainly two sets of periodic boundaries, one for the guide vanes and one for the runner blade. For each set of periodic boundaries two sides are identified, the *active* and the *non – active*. The *active* side is the one that when rotated in the positive pitch-wise direction (for

less than  $180^\circ$ ) matches perfectly with the other one. This is best understood by examining figure 2.2.



**Figure 2.2:** Graphical representation of the two periodic boundaries.

The result of the perfect match between the two sides is a list of all the corresponding nodes that belong to each periodic boundary.

In order to form the finite volume around each boundary node of the *active* periodic side, the neighbouring nodes of both the node on the *active* side and its corresponding node on the *non – active* side are used. Of course, the neighbouring nodes that belong to the *non – active* side are rotated appropriately to create the finite volume. After the formation of the finite volume the solution is updated in each boundary node of the *active* periodic side, and then directly copied to the matching boundary node of the *non – active* side.

## 2.4 Wall Functions

For the treatment of the boundary layer the PUMA solver provides the option of using wall functions. Specifically, the use of the formula for describing the velocity's behaviour inside the boundary layer proposed by Spalding [9], is employed. The formula is expressed as

$$y^+ = u^+ + 0.1108 \left\{ e^{0.4u^+} - 1 - 0.4u^+ - \frac{(0.4u^+)^2}{2!} - \frac{(0.4u^+)^3}{3!} - \frac{(0.4u^+)^4}{4!} - \frac{(0.4u^+)^5}{5!} \right\} \quad (2.11)$$

where  $y^+$  and  $u^+$  are the characteristic dimensionless expressions for distance and velocity, respectively, in the boundary layer and are defined, in relation to the friction

velocity  $u_\tau$ , as

$$y^+ = \frac{yu_\tau\rho}{\mu} \quad (2.12)$$

$$u^+ = \frac{u}{u_\tau} \quad (2.13)$$

Practically, formula 2.11 manages to describe both the viscous sublayer and the *log-law* region of the boundary layer with a single, unified formula. As stated by D. B. Spalding this is achieved by establishing a formula which:

- passes through the point:  $y^+ = 0, u^+ = 0$ .
- is tangential, at this point, to:  $u^+ = y^+$ .
- is asymptotic at large values of  $y^+$  to  $u^+ = 2.5\ln y^+ + 5.5$ .
- fits the experimental data, produced from a flat-plate flow experiment, at intermediate values of  $y^+$ .

## 2.5 Rotor-Stator Interaction (RSI)

The relative motion between successive components of a hydraulic turbine (i.e. guide vanes, runner, draft tube etc.) is handled using the mixing plane method.

This method is based on a pitch-wise averaging at the interface of the two components that are in relative motion. The interaction is carried out by exchanging circumferentially averaged flow quantities (such as total pressure, flow angles and static pressure). The averaging takes place over the radial zones that are formed at the two sides of the interface, as seen in figure 2.3.



**Figure 2.3:** *Graphical representation of the radial zones that are formed at a single boundary of the interface.*

## 2.6 Programming on GPUs

As discussed earlier, the PUMA solver had its starting point from an in-house CFD solver that runs on clusters of CPUs ([10], [11], [12]). The transition to a GPU-enabled solver presented some difficulties, due to the architecture and logic behind the GPUs themselves. GPUs are, in effect, shared memory parallel co-processors and this characteristic alone prohibits the programmer from using classic scatter-add techniques, that are widely used in CPU-based CFD solvers. As a result, the original CPU code had to be restructured and modified scatter-add techniques had to be employed (for further reading, refer to [4]).

The GPU solver uses the delta-formulation for solving eqs. 2.1 and 2.6 in their discretized form. In fact, the unknown variable is the update ( $\Delta U_n$ ) in the flow variables from one pseudo-time iteration ( $\kappa$ ) to the next ( $\kappa + 1$ ) as presented here.

$$\begin{aligned} \frac{\partial R_n}{\partial U_m} \Delta U_m &= -R_n \\ U_n^{k+1} &= U_n^k + \Delta U_n \end{aligned} \quad (2.14)$$

where  $\frac{\partial R_n}{\partial U_m}$  is the Jacobian matrix and is referred to as the left hand side (l.h.s.), while  $R_n$  is the residuals and is referred to as the right hand side (r.h.s.).

To further enhance the PUMA solver's efficiency, the Mixed Precision Arithmetics (MPA) is used [13]. It employs a combination of single and double precision arithmetics for the computations and data storage. Specifically, the residuals  $R_n$  are computed and stored as double precision numbers, due to the need for high accuracy. On the contrary, the l.h.s. coefficients ( $\frac{\partial R_n}{\partial U_m}$ ) are computed with Double Precision Arithmetics (DPA) but are stored in Single Precision Arithmetics (SPA).

The MPA technique, compared to the DPA technique, provides the same accuracy in the results, while reducing the memory requirements and the number of total memory accesses. There is only a slight difference in the convergence rate between the two techniques.



# Chapter 3

## Interfaces for the Integration of PUMA

In the process of implementing the GPU-enabled CFD solver, what becomes apparent, is the need for the development of various peripheral tools that will enable the smooth integration of the in-house solver inside the overall toolchain. These tools mainly include the two grid converters, the modification of the post-processing routines and the overall management script, that coordinates the whole process.

### 3.1 Grid Converters

First of all, the tools that had to be developed are the two grid converters to convert the 3D grids, generated by the meshing tool, developed and used in Andritz Hydro, from one format (CGNS) to another (LTT), and vice versa.

The first tool, converts the .cgns binary file, containing the desired geometry, to the LTT format needed by the PUMA solver. On the other hand, the second converter takes the output file from the solver, which is a Tecplot (.plt) file, containing the flow solution stored on the grid nodes, and transforms it back to a .cgns binary file that can be used for the post processing. This new file contains now both the geometry and the flow solution of the hydraulic turbine.

But first, in order to be able to develop said tools one must first understand in depth the three formats that are mentioned. In the following sections, the three formats (CGNS, LTT, Tecplot) are comprehensively described, so that the reader can understand the basic terminology pertaining to each format.

### 3.1.1 The CGNS Format

A CFD General Notation System (CGNS) file is an entity that is organized, inside the file itself, into a set of "nodes" in a structure resembling a tree. The base "node" is referred to as the "root node", containing general information for the file itself (i.e. the file size, the encoding used etc.). Each new "node" that is branched out of the "root node" is defined by a name and contains data [14].

Generally, there are two distinct methods of organizing CFD information in a hierarchy, *topologically based* and *data – type based*. Inside a CGNS file, a topologically based hierarchy is implemented. In this method, the information pertaining to each zone of the grid is grouped together, whereas in a data-type based hierarchy all the information pertaining to each flow-field variable is grouped together. This is best explained in figure 3.1. For example, the boundary patches of the grid play the role of the zones, with each one containing the flow solution and the grid element connectivity. Another zone is the volume grid of the main flow domain that contains all the grid coordinates and the flow data of the internal grid nodes [15].

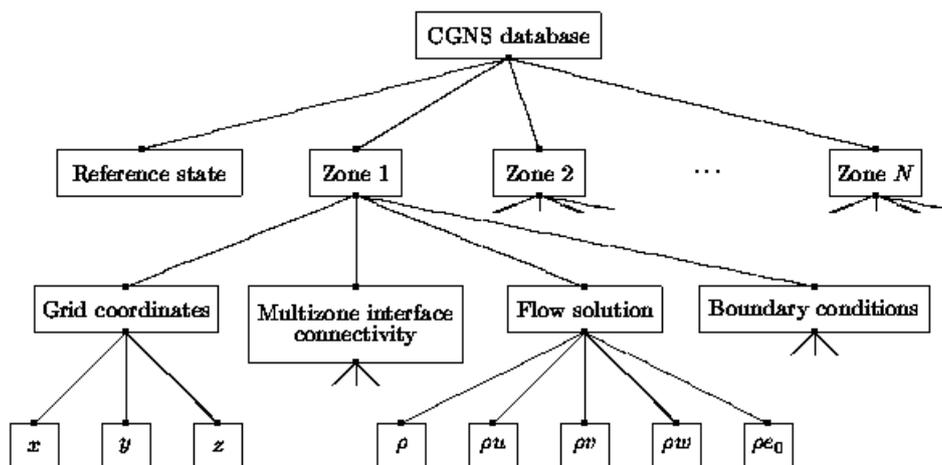


Figure 3.1: Graphical representation of the CGNS tree-like hierarchy.

### 3.1.2 The LTT Format

The computational grid is described in a set of three files, all of which consist the LTT format. These files are:

- The .nod file, containing the x, y, z coordinates of all the nodes.
- The .hyb file, containing the element connectivity of the grid.
- The .patch file, containing the information on the boundary patches of the mesh.

## The .nod file

The first file is the .nod file containing the x, y and z coordinates of the grid nodes. In the first line of the file the total number of nodes is written, while in the next three lines the x, y and z coordinates of each node are written, respectively.

```
nbNodes
nodeX1 nodeX2 nodeX3 ...
nodeY1 nodeY2 nodeY3 ...
nodeZ1 nodeZ2 nodeZ3 ...
```

**Figure 3.2:** Representation of the .nod file structure where  $nodeX^i$ ,  $nodeY^i$  and  $nodeZ^i$  are the x, y and z coordinates, respectively, of the  $i$ -th node.

## The .hyb file

The second file is the .hyb file containing the element connectivity of the grid. In further analysis, in the first line of the file the total numbers of tetrahedra ( $nbTet$ ), pyramids ( $nbPyr$ ), prisms ( $nbPri$ ) and hexahedra ( $nbHex$ ) is written. In the following lines, the nodal IDs of each element are written as shown in figure 3.3, where  $iHex_1^1$   $iHex_1^2$   $iHex_1^3$   $iHex_1^4$   $iHex_1^5$   $iHex_1^6$   $iHex_1^7$   $iHex_1^8$ , for example, are the nodal IDs of the nodes of the first hexahedron. The nodal IDs of each element type are written in a separate line

```
nbTet nbPyr nbPri nbHex
iTet11 iTet12 iTet13 iTet14 ...
iPyr11 iPyr12 iPyr13 iPyr14 iPyr15 ...
iPri11 iPri12 iPri13 iPri14 iPri15 iPri16 ...
iHex11 iHex12 iHex13 iHex14 iHex15 iHex16 iHex17 iHex18 ...
```

**Figure 3.3:** Representation of the .hyb file structure.

## The .patch file

The third and last file completing the three-file structure of the LTT format is the .patch file. This file contains all the necessary information concerning the grid boundaries (i.e. the patches). In the first line of the file, the total number of patches is written. In the following lines, sections with information per boundary patch are written. Each section contains, the name of the patch, the patch information and, finally, the list of nodal IDs that belong to each patch. Everything, except the patch name, is enclosed in curly brackets, as shown in figure 3.4.

```
nbPatches
patchName_1
{
    BCType boundaryConditionType | geometricType
    ConnectionId 1
    :
    Nodes nbNodes
    list_of_patch_node_ids
}
patchName_2
{
    BCType boundaryConditionType | geometricType
    ConnectionId 1
    :
    Nodes nbNodes
    list_of_patch_node_ids
}
...
```

**Figure 3.4:** Representation of the .patch file structure.

The keywords *boundaryConditionType* and *geometricType* can have multiple values, that are presented extensively in the table 3.1:

**Table 3.1:** Valid values for the keywords *boundaryConditionType* and the corresponding values for keyword *geometricType*.

<i>boundaryConditionType</i>	<i>geometricType</i>		
<i>InviscidStationaryWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>ViscousStationaryWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>InviscidRotatingWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>ViscousRotatingWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>MatchingConnection</i>	<i>PeriodicSideX</i>	<i>PeriodicSideY</i>	
<i>Inlet</i>			—
<i>Outlet</i>			—
<i>InletRSI</i>			—
<i>OutletRSI</i>			—

Finally, in a case which involves a Rotor-Stator Interaction (RSI), the keyword *ConnectionId* defines the corresponding "outlet" patch of the upstream component and "inlet" patch of the downstream component of the RSI interface.

### 3.1.3 The Tecplot Format

The PUMA solver provides the option of printing the resulting flow field in (a) LTT format ( a single .res file in which the computed pressure and velocity fields are written in separate lines), (b) VTU format (.vtk file) and/or (c) Tecplot format (.plt file). In the current diploma thesis the Tecplot output file is converted into CGNS format. The following paragraph describes the Tecplot format. For the Tecplot format option, a single output file (.plt file) is written. Every .plt file starts with a header containing general information about the grid and the variables that are presented inside the file, such as the total number of nodes and elements or the name and type of the variables included in the file. In figure 3.5, an example of a header is presented.

```

title = filename
variables = var_1, var_2, var_3 ...
zonet = zone_name
Nodes = nbNodes
Elements = nbElements
ZoneType = {FeBrick, FePoint}
DataPacking = {Block, Brick}
DT = (var_1_DataType, var_2_DataType, var_3_DataType ...)
Results

```

**Figure 3.5:** Example of the .plt file header.

After the header, the values of each variable are written. There are two ways of writing the flow field data. The first is the *point – wise* format, where there are as many columns as the variables to be written and each row of data corresponds to a single node. The second way is the *block – wise* format, where all the nodal values of each variable are written in a single block, with each block stacked below the previous one. Both are described in figures 3.6 and 3.7.

$$\begin{array}{l} var\_1\_Node^1, var\_2\_Node^1, var\_3\_Node^1 \quad \dots \\ var\_1\_Node^2, var\_2\_Node^2, var\_3\_Node^2 \quad \dots \\ var\_1\_Node^3, var\_2\_Node^3, var\_3\_Node^3 \quad \dots \end{array}$$

**Figure 3.6:** Example of the point-wise format, where  $var\_i\_Node^j$  is the value of the  $i$ -th variable of the  $j$ -th node.

$$\begin{array}{l} var\_1\_Node^1, var\_1\_Node^2, var\_1\_Node^3 \quad \dots \\ var\_2\_Node^1, var\_2\_Node^2, var\_2\_Node^3 \quad \dots \\ var\_3\_Node^1, var\_3\_Node^2, var\_3\_Node^3 \quad \dots \end{array}$$

**Figure 3.7:** Example of the block-wise format, where  $var\_i\_Node^j$  is the value of the  $i$ -th variable of the  $j$ -th node.

In the last section of the .plt file, the element connectivity is written, namely the 8 nodal IDs that define each hexahedral element.

### 3.1.4 The cgnsToLtt Converter

The first conversion tool programmed in this thesis is the cgnsToLtt, which converts the input .cgns file, containing the geometry, to the desired three-file structure of the LTT format. This is achieved through the use of the Cgns Mid-Level Library, which provides access to the binary .cgns file for reading all the necessary data, such as the grid size and coordinates, the element type and connectivity and information related to the boundary patches.

The tool is programmed to be user-friendly and the only input arguments that are needed is the name of the .cgns file, that is to be converted, and the name that the user desires to have on the output files.

In further detail, the first step is to read the cartesian coordinates of the mesh nodes from the .cgns file and create the .nod file. The next file to be written is the .hyb file that contains the element connectivity of the grid. In order to achieve this, the type (hexahedral, tetrahedral etc.) and element connectivity is read and, then, the .hyb file is written. Finally, the .patch file is written, which contains the necessary data

on the boundary patches, not only the nodes comprising these patches, but also the way that the boundary conditions will be imposed on each and every one of them.

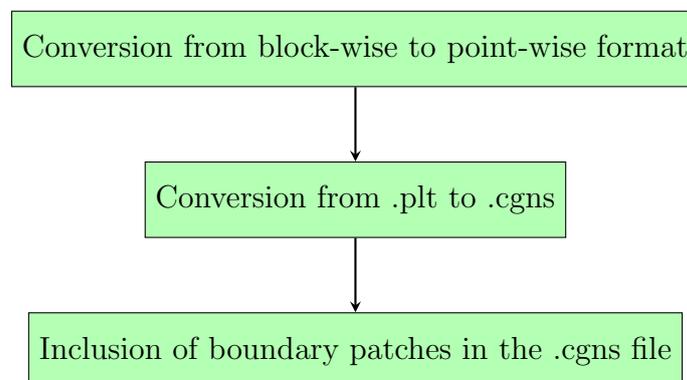
This final process is a combination of two steps. First of all, the nodes comprising each boundary patch is listed with a generic header above containing trivial information for the patch. Then, this header is appropriately modified for each boundary patch of the grid, so as to inform the PUMA solver of the way that boundary conditions need to be imposed. The information contained in the header are comprehensively presented in the section defining the LTT format.

### 3.1.5 The `pltToCgns` converter

The second converter that is programmed, is used to enable the compatibility between the solver and the rest of the post-processing. This necessity arose from the fact that the PUMA solver produces a `.plt` text file containing the flow solution and the geometry of the whole computational domain but the post-processing needed again a `.cgns` binary file.

As a result, the programmed converter transforms the `.plt` output text file into a single `.cgns` binary file that contains not only the flow solution on the internal nodes but also the overall geometry and the boundary surfaces separately, after first converting from the *block – wise* to the *point – wise* format.

The tool requires little input from the user, namely the number of computational domains (especially in the RSI cases) and the names of the input `.plt` file and the output `.cgns` file. Moreover, the tool requires no specific knowledge of programming by the user and keeps its inner workings hidden, so as to simplify its usage.



**Figure 3.8:** Graphical representation of the process of the `pltToCgns` converter.

## 3.2 Pre-processor for the PUMA Solver

The pre-processor is a code written in Fortran, that performs the necessary mathematical operations to prepare the final stages of the run of the PUMA solver. This step was required because the PUMA solver is a general purpose CFD solver that was not designed specifically for the analysis of turbomachinery. As a matter of fact, the user desires to give input values that have physical meaning for each case, such as the volumetric discharge  $Q_{tot}$  and hydraulic head  $H_{net}$  on the design point, but the solver requires more generalized input values, such as the fixed values of the velocity magnitude and the total pressure. Consequently, some additional mathematical operations need to be performed to turn the turbomachinery-specific values (discharge, head) into more general values (velocity, pressure).

The set of general operations that were necessary are presented here, while some additional operations that are case dependent are explained later.

- The conversion of the rotational speed from  $rpm$  to  $\frac{rad}{s}$ :

$$\omega = \frac{2\pi n}{60} \quad (3.1)$$

where:

- $\omega$  is the rotational speed in  $\frac{rad}{s}$ .
- $n$  is the rotational speed in  $rpm$ .
- The computation of the inlet absolute velocity magnitude based on the volumetric discharge  $Q$  of a single passage of the turbine:

$$V_{in} = \frac{Q}{A_{in} \sin(\alpha)} \quad (3.2)$$

where:

- $A_{in}$  is the area of a single-passage inlet patch (in  $m^2$ ).
- $\alpha$  is the inflow angle in degrees ( $^\circ$ ).
- The computation of the static pressure at the outlet  $p_{s,out}$ :

$$p_{s,out} = p_{atm} + \rho g(z_{ref} - z_{out} - Z_s) - \Delta p_{s,out} \quad (3.3)$$

where:

- $p_{atm}$  is the atmospheric pressure (in  $Pa$ ).
- $z_{ref}$  is the reference level (in  $m$ ).
- $z_{out}$  is the level of the outlet of the turbine (in  $m$ ).

- $Z_s$  is the suction head (in  $m$ ), and is calculated by equation 3.4.

$$Z_s = -\sigma H_{net} - \frac{p_{atm} - p_{vap}}{\rho g} - \frac{V_{dt,out}^2}{2g} \quad (3.4)$$

where  $V_{dt,out} = \frac{Q_{tot}}{\frac{D_{ref}^2}{\pi \cdot 4} f_{A,dt}}$  is the estimated velocity magnitude at the draft tube outlet, while  $f_{A,dt}$  is the ratio between the area of the runner outlet of a single-passage and the area of the draft tube outlet. Also  $\sigma$  is the cavitation coefficient.

- $\Delta p_{s,out}$  is the estimated pressure drop in the draft tube computed by the formula  $\Delta p_{s,out} = \frac{1}{2} \rho \eta_{dt} (1 - \frac{1}{f_{A,dt}^2}) (\frac{Q_{tot}}{A_{outnbl}})^2$ , where  $\eta_{dt}$  is the draft tube's efficiency.

This code was programmed with the aim of tackling the different boundary conditions that need to be imposed depending on each case and keeping in mind that these boundary conditions need to be imposed in the same manner as they were being imposed before the introduction of the PUMA solver in the general toolchain.

In effect, in this step what is being done is the "translation" of the hydraulic turbine-specific boundary conditions to a set of more generalised boundary conditions.

### 3.2.1 The Imposed Discharge Condition

The first boundary condition, is the imposed discharge at the runner inlet. This boundary condition, generally, is used in a runner-alone case.

Moreover, when imposing this boundary condition, the inlet flow angle ( $\alpha$  coincides with angle  $\theta_1$  mentioned in eqs. 2.9) is computed from the following equation

$$\alpha = \arctan\left(\frac{\omega R Q}{A_{in} g H_{net}}\right) \quad (3.5)$$

Although the assumption that the flow exits the runner purely in the axial direction may seem a little unwarranted, the comparative results between the PUMA solver and the commercial solver are almost the same. Hence, equation 3.5 is proved to be a valid method for computing the inflow angle  $\alpha$ .

### 3.2.2 The Imposed Head Condition

The other boundary condition that is used in guide vane - runner cases or guide vane - runner - draft tube cases is the imposed head boundary condition.

This boundary condition is translated into imposing the total pressure at the inlet and the static pressure at the outlet. The steps that are needed to compute the necessary values are described below.

First of all, the static pressure at the outlet is evaluated based on equation 3.3.

Then, based on the average velocity magnitude at the outlet patch  $\overline{V_{out}}$ , the dynamic pressure is computed  $p_{dyn,out} = \frac{1}{2}\rho\overline{V_{out}}^2$ , thus the total pressure at the outlet becomes available. Finally, the total pressure at the inlet is computed by adding the hydraulic head of the turbine. The complete formula is presented in equation 7.5.

$$p_{tot,in} = \rho g H_{net} + p_{s,out} + p_{dyn,out} \quad (3.6)$$

A feature that needed to be included in the PUMA solver is the ability to update the average velocity magnitude at the outlet within each iteration.

To that effect, some changes were programmed in the PUMA solver, in order to enable it to impose the boundary condition in the same way that it was already being imposed, meaning that now the total pressure at the inlet can be updated in each iteration based on the mass flow averaged velocity magnitude at the outlet  $\overline{V_{out}}$ .

### 3.3 The set-up-run-gpu Shell Script

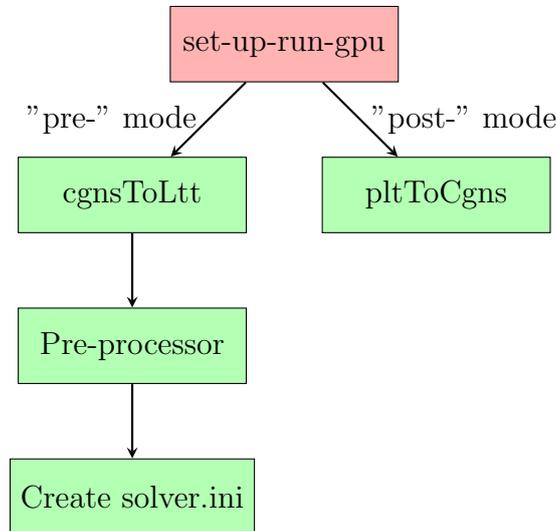
The final tool that was programmed is the "set-up-run-gpu" shell script. The basic purpose of this script is to manage the smooth run of the various tools described above (i.e. the cgnsToLtt, the pre-processor and the pltToCgns) and to perform some additional tasks to combine and coordinate the different tools, such as some additional manipulation of the .patch file and creating the desired input file for the PUMA solver (i.e. solver.ini file).

The steps that are executed here are the following:

1. Reading the general input file, containing the data concerning each hydraulic turbine case, such as discharge, head, number of blades, etc.
2. Converting the .cgns file to the LTT format described above.
3. Performing the necessary pre-processing, namely converting the discharge and hydraulic head to velocity magnitude and total pressure at the inlet etc.
4. Converting the .plt output file back to a .cgns file containing now the flow solution.

This script was programmed to support two options of execution. The first option is "pre-", and actually performs steps 1-3, while the second option is "post-" and

performs step number 4, meaning the second conversion from a .plt file to a .cgns file. All these are best represented in figure 3.9



**Figure 3.9:** Graphical representation of the processes handled by the "set-up-run-gpu" shell script.

## 3.4 Post-processor Functions

The last step in enabling the smooth integration of the PUMA solver in the unified toolchain was to modify the necessary files (.sub) in the already existing post-processing.

The main problem with these files was the input they were being given. In the existing post-processing files, a binary file was given as input, which contained additional pieces of information, aside from the flow field data. These additional information could not be included in the .cgns file that is produced after the final conversion.

As a result, some of the key functions being used, could not anymore, as they relied on these additional information. The list of functions that did not support the .cgns format include the mass flow related functions (e.g. mass flow averaging, computation of the mass flow etc.) and the computation of the torque that is being produced from the turbine.

To overcome these obstacles, it was necessary to create alternative functions that perform the same computations. In other words, this new set of functions performs the computations step-by-step that the previous set of functions did automatically. To further explain the procedure, this is the equivalent set of functions that was used instead of the hardcoded functions.

### 3.4.1 Mass Flow Related Functions

The mass flow related functions (or subroutines) include the mass flow computation, averaging and integrating. These three hardcoded functions, relied heavily on the data already existing in the input file and could not support the .cgns file format.

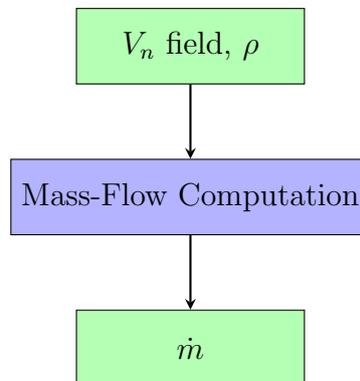
Consequently, they had to be replaced with new functions that performed the actual analytical calculations that are implied by the name of these functions. These calculations are presented here:

- Mass flow computation:

$$\dot{m} = \iint_A \rho V_n dA \quad (3.7)$$

where:

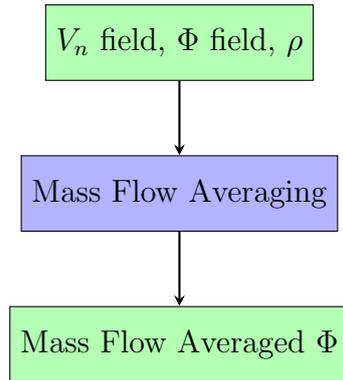
- $V_n$  is the velocity vector normal to the surface where the mass flow is computed.



**Figure 3.10:** Graphical representation of the mass-flow calculation function.

- Mass-flow averaging of a flow variable  $\Phi$ :

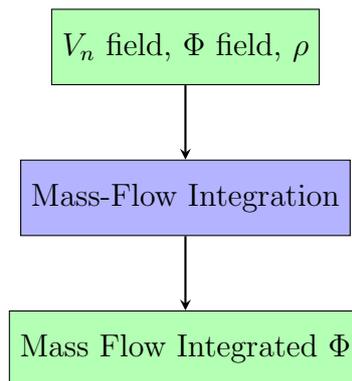
$$\bar{\Phi}_{mass} = \frac{\iint_A \rho V_n \Phi dA}{\iint_A \rho V_n dA} \quad (3.8)$$



**Figure 3.11:** Graphical representation of the mass flow averaging function.

- Mass-flow integrating a flow variable  $\Phi$ :

$$\iint_A \rho V_n \Phi dA \quad (3.9)$$



**Figure 3.12:** Graphical representation of the mass-flow integration function.

### 3.4.2 Torque Computation Function

The same problem concerns the function that calculates the torque that is produced from the turbine. Although, this function was a bit more complicated to replace, as it required a few more computations to be performed in order to obtain the final result. Generally, the torque produced from the hydraulic turbine is the combination of two terms, as follows

$$M_{tot} = M_{p,z} + M_{v,z} \quad (3.10)$$

The first term ( $M_{p,z}$ ) is the result of the pressure force exerted on the blade surface, while the second term ( $M_{v,z}$ ) is produced by the shear stresses on the blade surface.

The pressure term ( $M_{p,z}$ ) is computed as the surface integral of the pressure multiplied by the local radius of the blade, as follows

$$M_{p,z} = \iint_{A_{bl,x}} xp_{abs} dA - \iint_{A_{bl,y}} yp_{abs} dA \quad (3.11)$$

where:

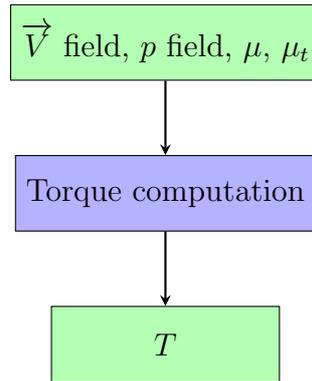
- $A_{bl,x}$  is the surface area of the blade projected on the  $yz$  plane.
- $A_{bl,y}$  is the surface area of the blade projected on the  $xz$  plane.
- $p_{abs}$  is the absolute pressure.

Similarly, the shear stress term ( $M_{v,z}$ ) is computed again as a surface integral of the shear forces on the blade multiplied by the radius.

$$M_{v,z} = \iint_{A_{bl}} x\tau_{w,y} dA - \iint_{A_{bl}} y\tau_{w,x} dA \quad (3.12)$$

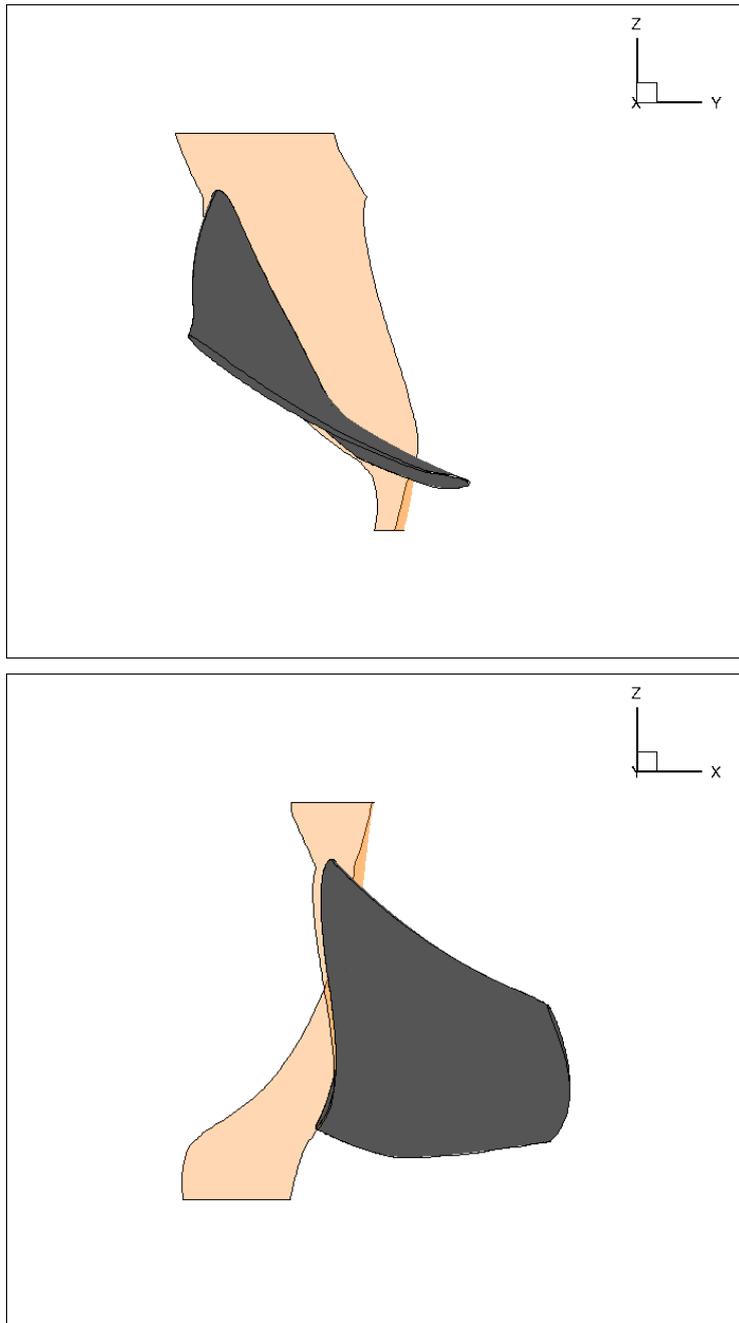
where:

- $\tau_{w,x} = (\mu + \mu_t)(2n_x \frac{\partial u}{\partial x} + n_y(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) + n_z(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}))$ .
- $\tau_{w,y} = (\mu + \mu_t)(n_x(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) + 2n_y \frac{\partial v}{\partial y} + n_z(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}))$ .



**Figure 3.13:** Graphical representation of the torque computation function.

In figure 3.14, the runner blade is shown including the frame of reference, as well as the two projected areas  $A_{bl,x}$  and  $A_{bl,y}$ .



**Figure 3.14:** *The runner blade (grey) and the hub (pink) of a hydraulic turbine's single passage. Also, the two projected areas  $A_{bl,x}$  (top) and  $A_{bl,y}$  (bottom).*



# Chapter 4

## Case Studies and Comparison

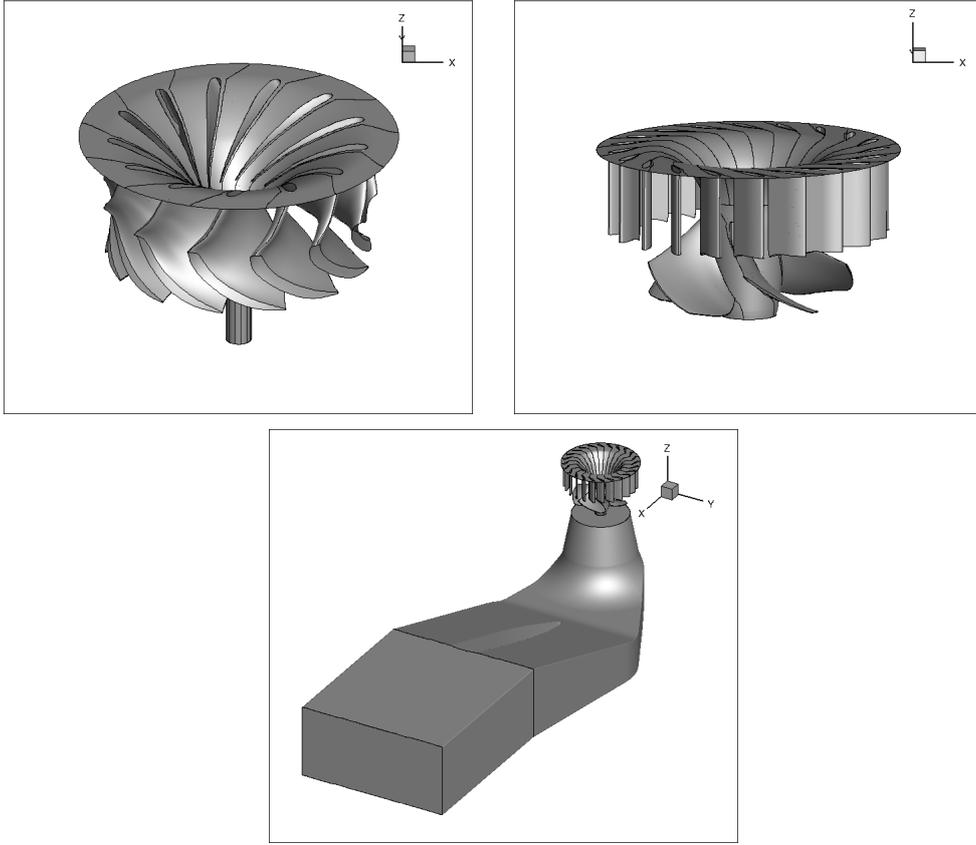
After the development of the necessary tools for the proper integration of the PUMA solver, the next step is to test and validate the solver. This was achieved by running different cases of turbine configurations and then comparing the results with a commercial CFD software provided by Andritz Hydro.

The cases that are tested include the runner of a Francis turbine, the guide vanes and runner of a propeller turbine and the combination of the guide vanes, runner and draft tube of a Kaplan turbine. The respective turbine configurations are presented in figure 4.1.

To ensure fairness in comparing the results of the PUMA solver and the results of the commercial CFD solver, it is important to clearly state that the post processing is exactly the same in both solvers. Thus, the only differences that arise are due to the solvers themselves. Also, it is noteworthy to state that both solvers employ a vertex-centered finite volume method for numerically solving the RANS equations. Conversely, there are some key differences between the two solvers. First, the method for numerically solving the RANS equations is different. The PUMA solver uses the pseudo-compressibility method, whereas the commercial solver uses a pressure-based method. Second, the PUMA solver employs the S-A turbulence model, while the commercial solver employs the  $k - \omega$  SST turbulence model. Third, the wall functions used for the boundary layer treatment are different in the two solvers.

The results that are presented for each case are values and graphs that best describe the operation of the turbine. Mainly, the graphs that are presented are those of the pressure coefficient  $C_p$  profiles in three key positions (hub, mid-span, shroud) and those of the dimensionless absolute cylindrical velocity profiles at the outlet of the runner. The latter are important, because they define to a great extent the efficiency of the draft tube that follows the runner.

The pressure coefficient  $C_p$  is computed, during the post processing phase over the whole runner blade. The pressure coefficient  $C_p$  at each point on the blade surface



**Figure 4.1:** *The three turbine configurations that are tested. The Francis runner (top-left), the propeller turbine guide vanes and runner (top-right) and the Kaplan guide vanes and runner including the draft tube (bottom).*

is defined as

$$C_p = \frac{p - p_{vap}}{\rho g H_{net}} \quad (4.1)$$

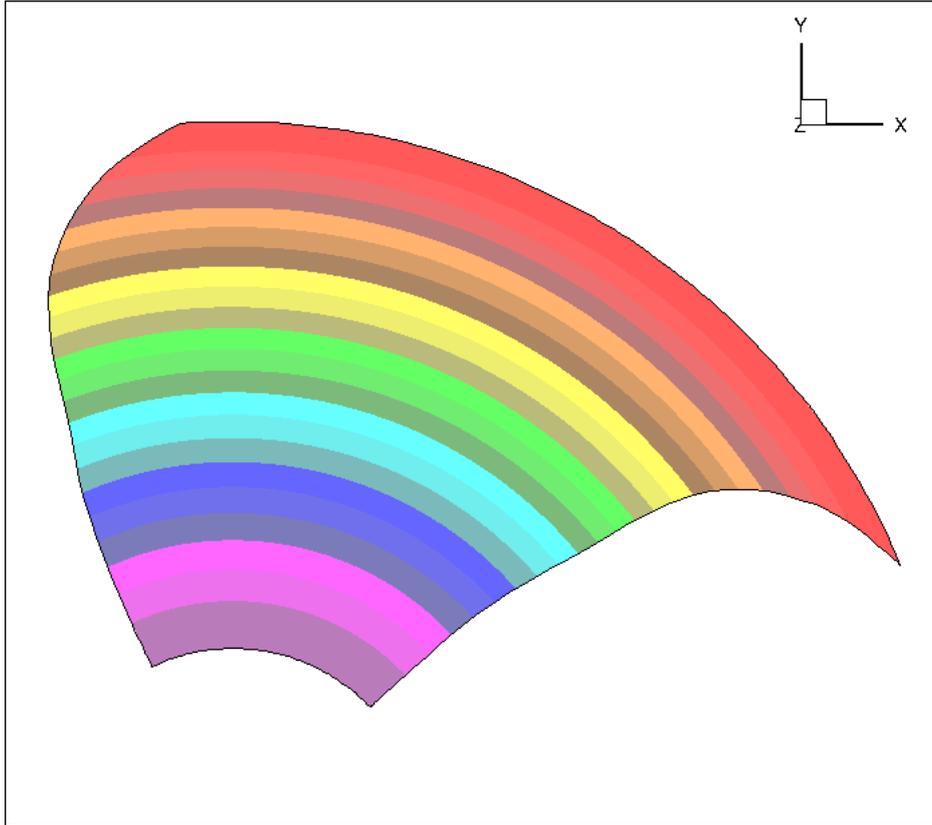
where  $p_{vap}$  is the evaporation pressure of water and  $H_{net}$  is the hydraulic head given as input.

Also, the three dimensionless velocity profiles that are presented are the profiles of the peripheral  $c_u^*$ , radial  $c_r^*$  and meridional  $c_m^*$  absolute velocity components at the outlet of the runner. They are expressed as

$$\begin{aligned} c_u^* &= \frac{c_u}{\sqrt{2gH_{net}}} \\ c_r^* &= \frac{c_r}{\sqrt{2gH_{net}}} \\ c_m^* &= \frac{c_m}{\sqrt{2gH_{net}}} \end{aligned} \quad (4.2)$$

where  $c_u$  is the magnitude of the dimensional peripheral absolute velocity,  $c_r$  is the magnitude of the dimensional radial absolute velocity and  $c_m = \sqrt{c_r^2 + c_a^2}$  is the magnitude of the dimensional meridional absolute velocity.

For these quantities to be computed, first, the outlet of the runner is divided into radial zones as seen in fig. 4.2 and, then, for each zone the average values of eqs. 4.2 are computed.



**Figure 4.2:** *The radial zones created for averaging the three velocity components over the outlet patch.*

Finally, for each case, some characteristic quantities are presented, that further define the turbine's operation. Those include the hydraulic head, the volumetric discharge, the torque that is produced and the efficiency that can be achieved.

The hydraulic head (in  $m$ ) is defined as the drop in the mass-averaged values of the total pressure from the inlet to the outlet of the runner as computed from the solver, while also the volumetric discharge  $Q$  (in  $\frac{m^3}{s}$ ) is computed directly from the PUMA solver. The computation of  $H$  is expressed as

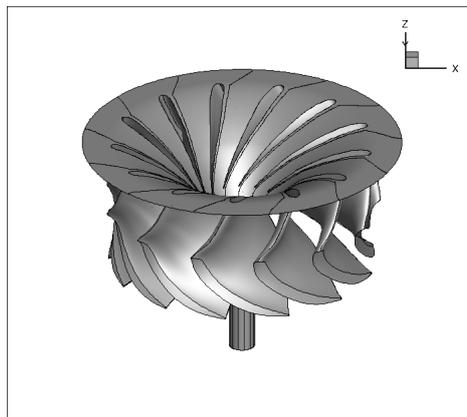
$$H_{calc} = \frac{p_{t,in} - p_{t,out}}{\rho g} \quad (4.3)$$

In addition, the torque  $T$  (in Nm) is computed from equation 3.10 for each of the rotating parts of the turbine (the runner blade and the hub). Finally, efficiency  $\eta$  is computed based on the formula

$$\eta = \frac{T\omega}{\rho g H_{calc} Q} \quad (4.4)$$

## 4.1 Francis Runner

The first case is that of a Francis runner blade. This case concerns only the runner of the turbine, which has 13 blades, and is a typical case for applying the discharge boundary condition. Generally, in the Francis cases, it is typical to study only the runner. The configuration is presented figure 4.3:



**Figure 4.3:** *The Francis hydraulic turbine blade configuration.*

A single passage of the turbine is studied and its grid consists of 220416 nodes and comprises hexahedral elements. The size of the grid is generally small, especially compared to later cases that include multiple turbine components. As a result, the wall-clock time is extremely small, almost 5 minutes on 4 GPUs (GeForce GTX 1080 Ti).

This case is run for the best-efficiency operating point on the Hill chart. This is described in table 4.1.

**Table 4.1:** *CFD analysis of the Francis runner using PUMA. Operating point of the turbine.*

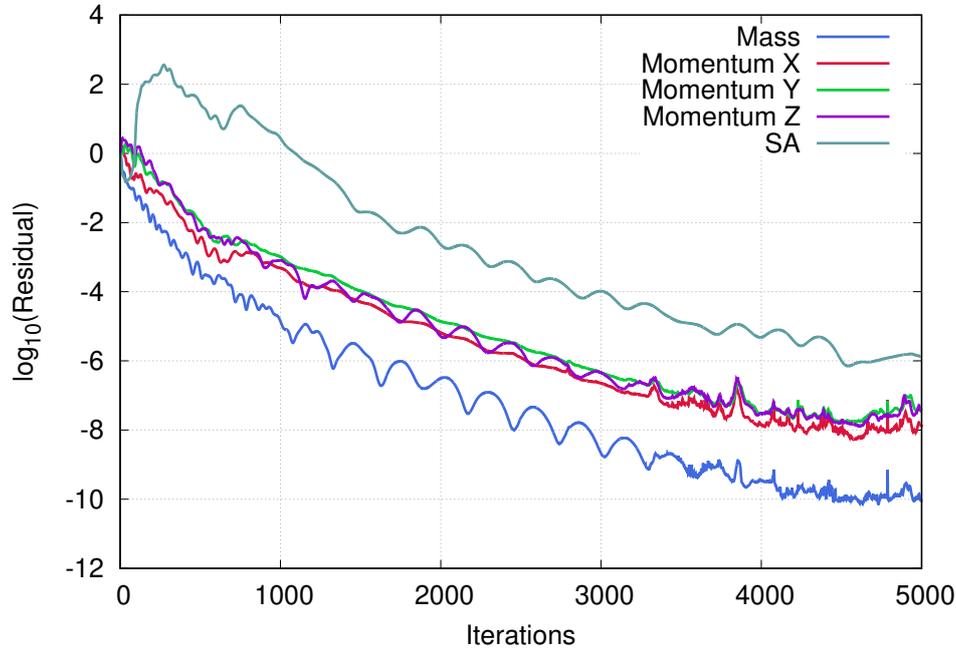
Variable	Value
H (m)	34.59
Q ( $\frac{m^3}{s}$ )	0.57
N (rpm)	1021.04

For this case, a constant value for the absolute velocity magnitude accompanied by the two flow angles are imposed at the inlet and a constant value for the static pressure is imposed at the outlet. The specific values for all the necessary quantities on the boundaries are presented in table 4.2.

**Table 4.2:** *CFD analysis of the Francis runner using PUMA. Imposed values of the flow field variables. The  $\theta_1$  flow angle results from the computation  $\theta_1 = 180 + \alpha$ , where  $\alpha$  is computed by eq. 3.5, because the turbine revolves in the counter-clockwise direction.*

Variable	Value
$V_{in}$ ( $\frac{m}{s}$ )	15.49
$\theta_1$ (deg)	197.895
$\theta_2$ (deg)	90.0
$\frac{p_{s,out}}{\rho}$ ( $\frac{Pa}{kg/m^3}$ )	32.35
$\frac{\nu_{in}}{\nu}$	20.0

In figure 4.4, the residual convergence rate is presented of each of the PDEs that are being solved, which are the continuity equation, the three components of the momentum equation and the Spalart-Allmaras equation.



**Figure 4.4:** *CFD analysis of the Francis runner using PUMA. Residual convergence of the five equations (continuity, three components of the momentum and S-A model).*

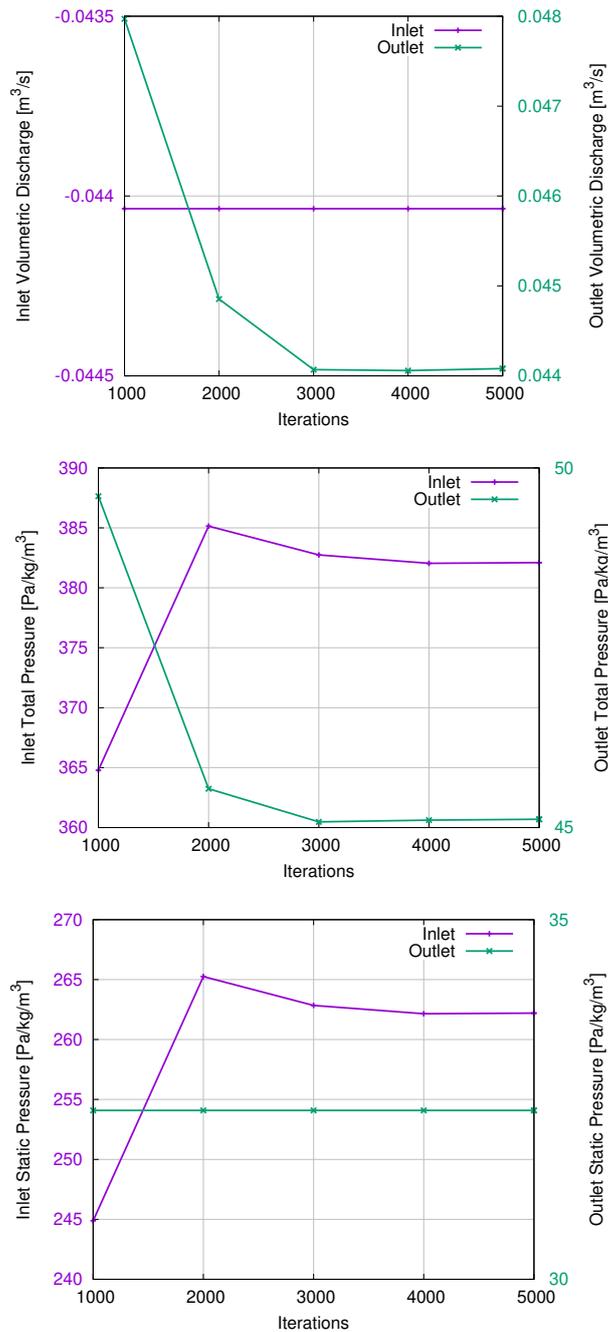
In figure 4.4, it is evident that the residual magnitude order decreases significantly (from 6 to 10 orders depending on the PDE) and after 4000 iterations the residual magnitude order seems to maintain a relatively stable value.

Furthermore, the convergence rate of certain integral quantities is presented in figure 4.5, namely the static and total pressure and the volumetric discharge at the inlet and outlet of the flow domain. As it can be seen from figure 4.5, the values of these quantities all seem to stabilize after the first 3000 iterations.

Based on the previous two figures (4.4, 4.5), it is safe to assume that the case is converged before the 5000 iterations. This means that the computation could have been stopped at an earlier point, thus reducing even more the computational cost.

The computed pressure coefficient  $C_p$  profiles are presented in figure 4.6. There are three profiles, one at the hub, one at the mid-span and one at the shroud. From figure 4.6 it can be inferred that the deviation between the results of the two solvers are negligible, providing a first step towards the validation of the PUMA solver.

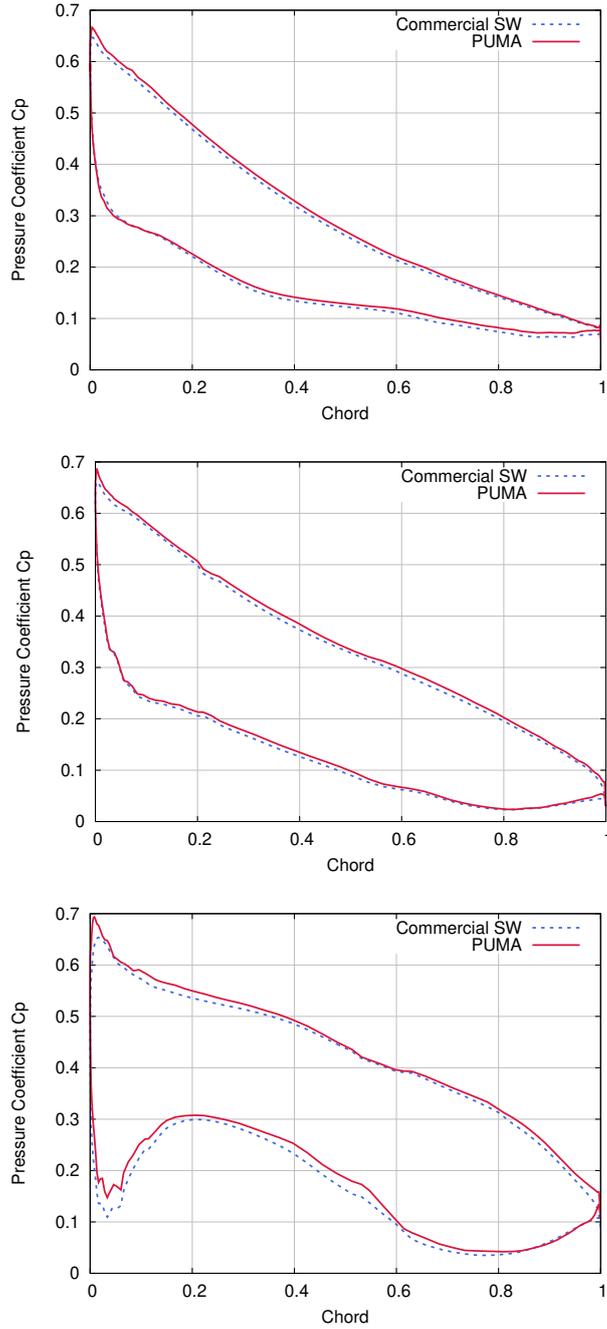
In figure 4.7 the dimensionless profiles of the peripheral  $c_u^*$ , radial  $c_r^*$  and meridional  $c_m^*$  velocity components at the outlet of the runner are presented. As it can be seen in figure 4.7, the profiles of the absolute cylindrical velocity components produced



**Figure 4.5:** CFD analysis of the Francis runner using PUMA. Convergence of integral quantities, such as mass flow (top), total (middle) and static (bottom) pressure at the inlet and outlet boundaries.

from the two solvers seem to be almost the same, indicating further the equivalence of the two softwares.

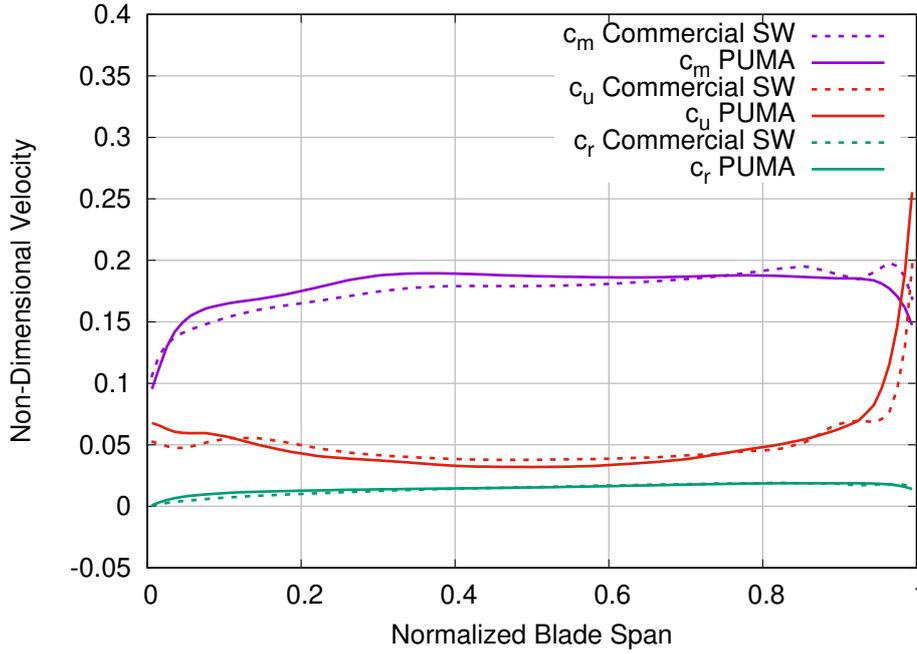
In table 4.3 some characteristic quantities are presented, describing the turbine



**Figure 4.6:** CFD analysis of the Francis runner using PUMA. Pressure coefficient ( $C_p$ ) over the blade section at the hub (top), mid-span (middle) and shroud (bottom) positions.

operation.

Based on table 4.3, the results from the comparison are satisfying. The only quantity that presents a large deviation (2.4 %) is that of the efficiency  $\eta$ .



**Figure 4.7:** *CFD analysis of Francis runner using PUMA. Dimensionless profiles of the radial, peripheral and meridional velocity components at the outlet.*

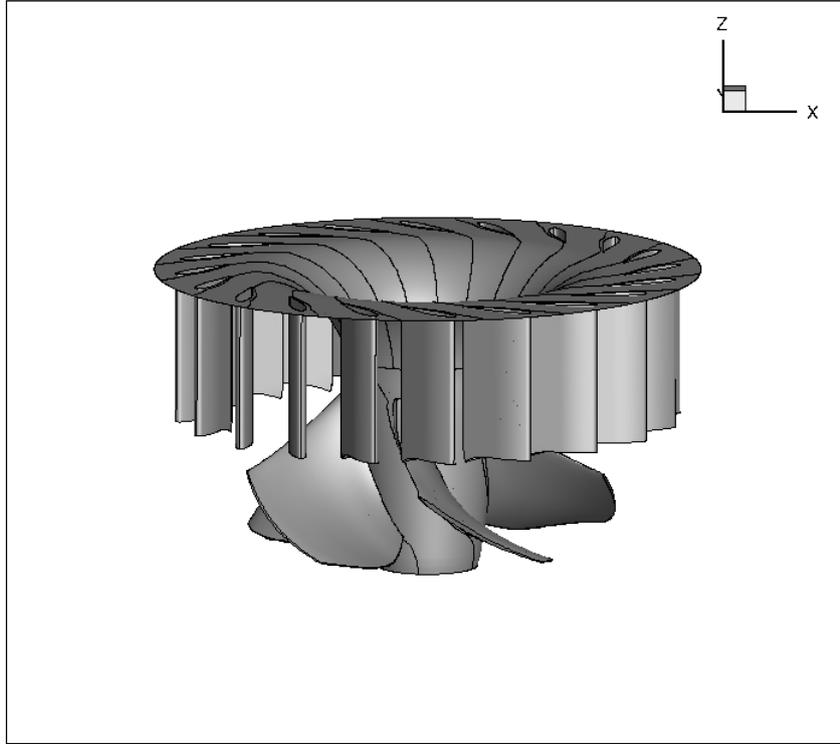
**Table 4.3:** *CFD analysis of Francis runner using PUMA. Computed characteristic quantities of the hydraulic turbine operation for the best-efficiency operating point. As far as the efficiency  $\eta$ , the percentile deviation from the result produced by the commercial solver is presented.*

Quantity	PUMA	Commercial SW
H (m)	34.37	33.92
Q ( $\frac{m^3}{s}$ )	0.573	0.571
$\delta\eta(\%)$	+2.4	-
T (Nm)	1792.21	1720.75

## 4.2 Propeller Guide Vane - Runner

The second case that is tested is that of a propeller turbine. Here we have a combination of two components, the guide vanes (20 blades) and the runner (5 blades). The configuration is presented in figure 4.8.

The grid of the two passages (one passage for the guide vanes and one passage for the runner) has 1061221 nodes, more specifically 606375 nodes for the guide vanes and 454846 nodes for the runner, and consists, again, only of hexahedral elements. A slightly finer grid is used than before, so the wall-clock time on 4 GPUs (GeForce GTX 1080 Ti) is now almost 10 minutes.



**Figure 4.8:** *The propeller hydraulic turbine configuration.*

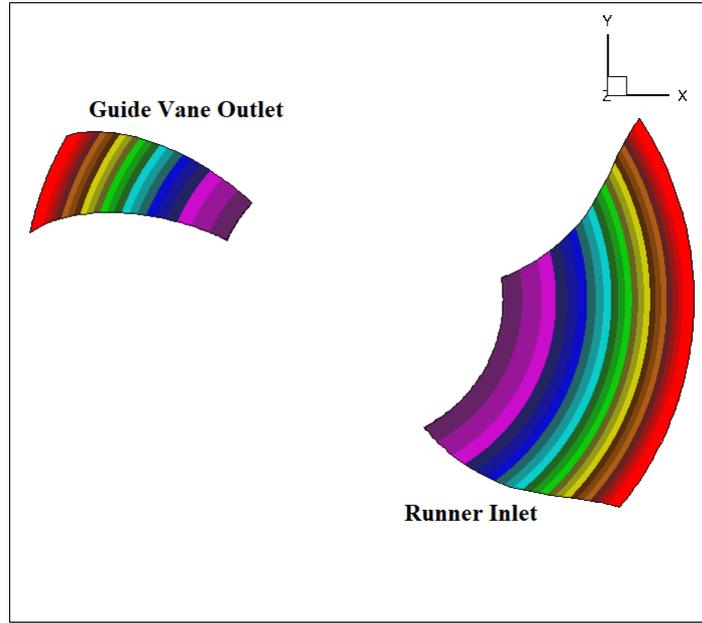
The operating point for which case is described in the following table 4.4.

**Table 4.4:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. Operating point of the turbine.*

Variable	Value
H (m)	13.22
Q ( $\frac{m^3}{s}$ )	0.51
N (rpm)	1651.87

The mixing plane method was used for the RSI connection between the guide vanes and the runner. For the mixing plane method, 21 radial zones are formed on both the outlet boundary of the guide vane passage and the inlet boundary of the runner passage. In each zone of the upstream component (guide vane outlet) the flow variables are averaged circumferentially and are passed on to the downstream component (runner inlet). The radial zones created for the mixing plane method are represented in figure 4.9.

For this case, a constant value for the total pressure along with the two angles are imposed at the inlet and a constant static pressure at the outlet. The specific values for all the necessary quantities imposed on the boundaries are presented in table 4.5.



**Figure 4.9:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. The radial zones formed for the mixing plane method at the guide vane outlet of single passage (left) and the runner inlet of a single passage (right). The flow quantities are averaged over each zone of the guide vane outlet and each average value is transferred to the runner inlet radial zone of the corresponding colour.*

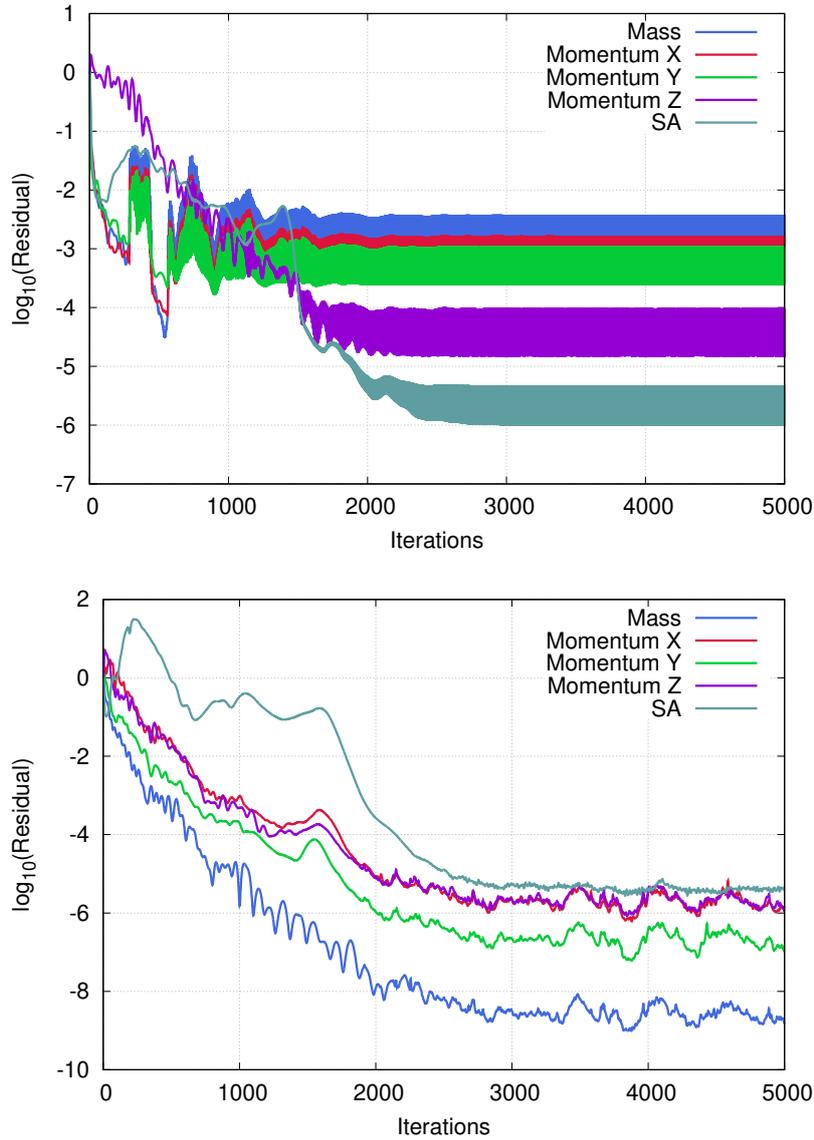
**Table 4.5:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. Imposed values of the flow field variables. The  $\theta_1$  flow angle results from the computation  $\theta_1 = 180 + \alpha$ , where  $\alpha$  is the known guide vane angle ( $45^\circ$ ), because the turbine revolves in the counter-clockwise direction.*

Variable	Value
$p_{t,in} \left( \frac{m}{s} \right)$	261.11
$\theta_1 \text{ (deg)}$	225.0
$\theta_2 \text{ (deg)}$	90.0
$\frac{p_{s,out}}{\rho} \left( \frac{Pa}{kg/m^3} \right)$	91.98
$\frac{\nu_{t,in}}{\nu}$	20.0

First of all, the convergence rate of the residuals of the governing equations is presented below in figure 4.10, for both components, the guide vanes and the runner blade.

Furthermore, the convergence rate of the static and total pressure and the volumetric discharge at the inlet and outlet of the turbine is presented in figure 4.11.

By examining both figures 4.10 and 4.11, it is clear that the reduction of the residual magnitude order is noticeable and specifically after 2500 iterations the residuals seem to stabilize. The latter is particularly evident in the convergence rate figure of the guide vane passage. Also, the integral quantities of figure 4.11 appear to be stable

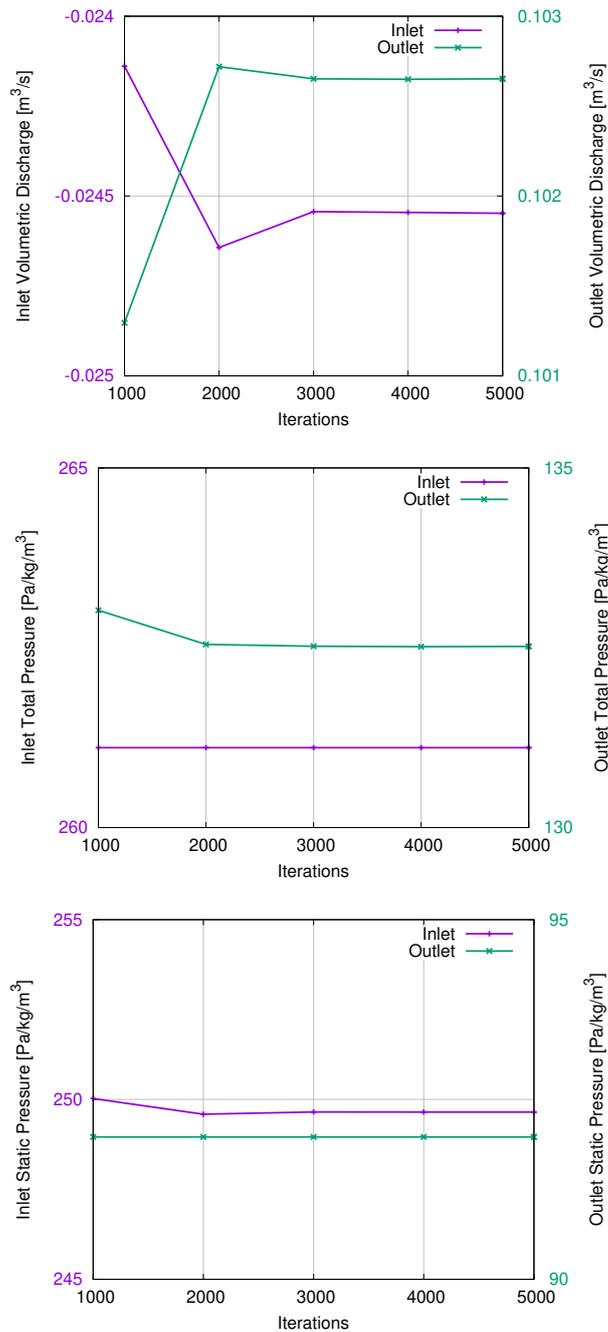


**Figure 4.10:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. Residual convergence of the five equations (continuity, three components of momentum and S-A turbulence model) for the guide vanes (top) and the runner blade (bottom).*

after 3000 iterations.

The resulting pressure coefficient  $C_p$  profiles are presented for the runner blade in figure 4.12. The three profiles, presented for the hub the mid-span of the blade and the shroud, again appear to be almost identical.

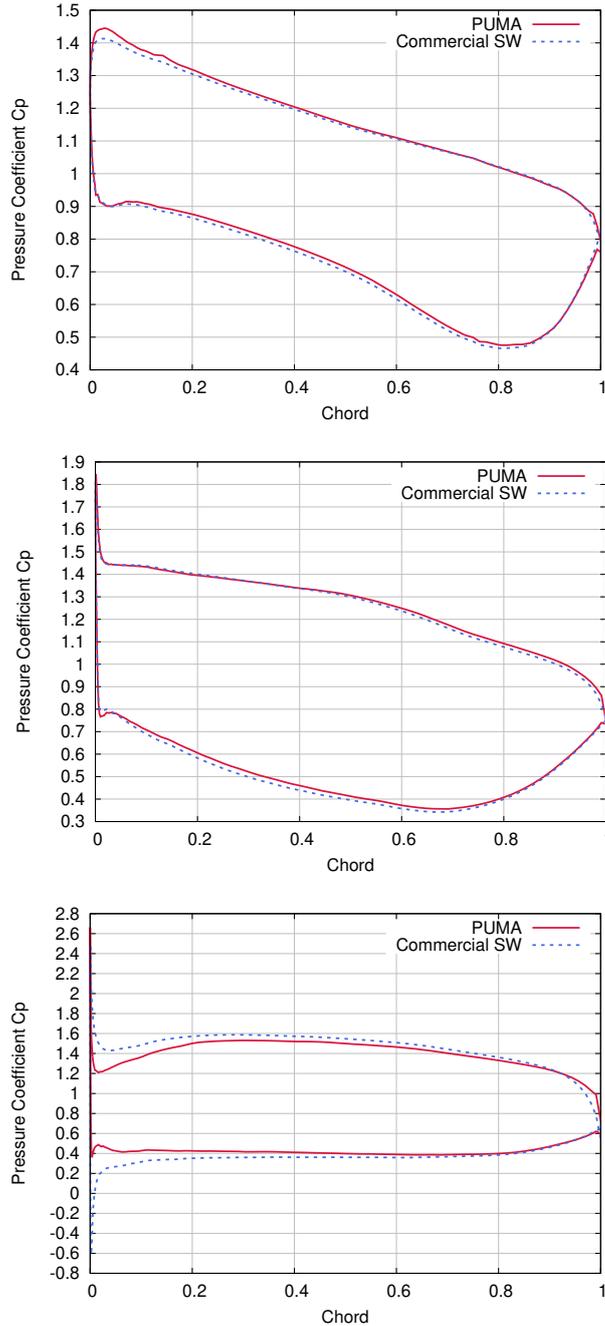
In figure 4.13, the dimensionless profiles of the peripheral  $c_u^*$ , radial  $c_r^*$  and meridional  $c_m^*$  absolute velocity components at the outlet of the runner are presented. The comparison between the profiles produced from the solvers, further proves the



**Figure 4.11:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. Convergence of integral quantities, such as mass flow (top), total (middle) and static (bottom) pressure over the inlet and outlet patches.*

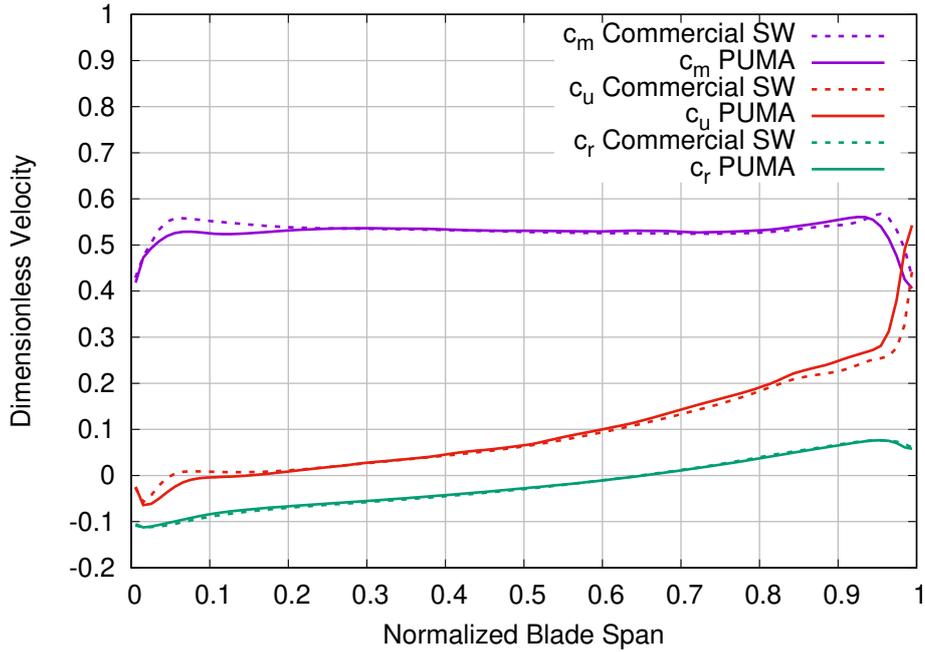
validity of the results of the PUMA solver, as the profiles match almost perfectly.

In table 4.6, some characteristic quantities, describing the turbine operation, are presented:



**Figure 4.12:** *CFD analysis of the propeller turbine guide vanes and runner using PUMA. Pressure coefficient ( $C_p$ ) over the blade section at the hub (top), mid-span (middle) and shroud (bottom) positions.*

In this case, the deviation between the results of the two solvers is smaller than the Francis case, even in the computation of the efficiency  $\eta$ . Such a good comparison between the two software packages shows that the PUMA can safely be used instead of the commercial software into the optimization loop.



**Figure 4.13:** CFD analysis of the propeller turbine guide vanes and runner. Dimensionless profiles at the outlet of the radial, peripheral and meridional velocity components.

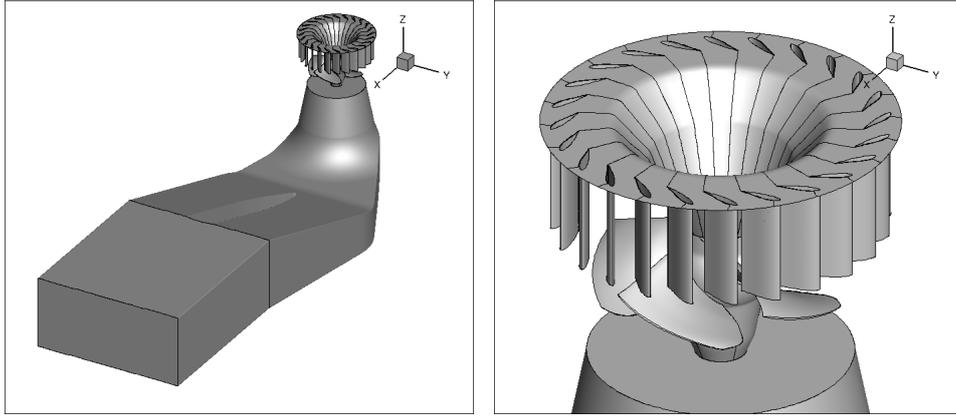
**Table 4.6:** CFD analysis of the propeller turbine guide vanes and runner using PUMA. Overall results describing the hydraulic turbine function. As far as the efficiency  $\eta$ , the percentile deviation from the result produced by the commercial solver is presented.

Quantity	PUMA	Commercial SW
H (m)	13.20	13.18
Q ( $\frac{m^3}{s}$ )	0.513	0.517
$\delta\eta(\%)$	+1.4	-
T (Nm)	375.86	372.76

### 4.3 Kaplan Guide Vane - Runner - Draft Tube

The third case that is tested was that of a Kaplan turbine (guide vanes and runner) coupled with a draft tube. Here, the turbine's configuration combines three components, the guide vanes (24 blades), the runner (5 blades) and the draft tube, making this case the most computationally demanding. The configuration is presented in figure 4.14. For each of the guide vane and runner components, a single passage of the axisymmetric flow domain is used, while for the draft tube the complete flow domain is used.

It is noteworthy to state that predicting the flow inside the draft tube is considered



**Figure 4.14:** *The Kaplan hydraulic turbine configuration, including the draft tube. The whole configuration of the guide vanes, runner blades and draft tube (left) and a close-up view of the guide vanes and runner (right).*

a very demanding task and even the commercial solver cannot reproduce the results yielded in the lab, which is certainly not the case for the other components of the turbine.

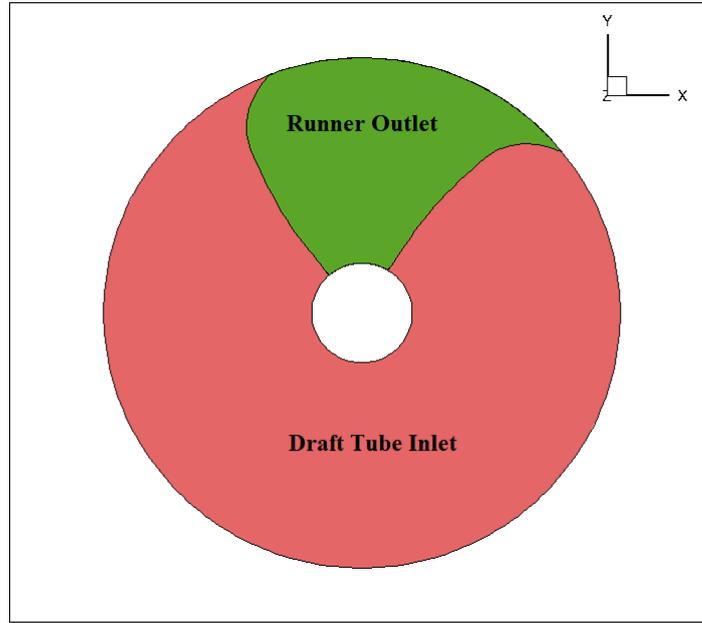
An interesting difference between this new case and the previous one including an RSI, is the interface that is formed between the runner and the draft tube. As it can be seen in figure 4.15, the outlet of the runner corresponds to a single passage of the turbine, while the inlet of the draft tube corresponds to the complete draft tube. Conversely, the interface between the guide vanes and the runner is formed from the outlet of the guide vanes and the inlet of the runner, which both correspond to a single passage of the turbine. Both RSI interfaces are tackled with the mixing plane method, so there is no need to include the full 360° configuration of the guide vanes and the runner.

The grid of this case has 3690349 nodes, more specifically 148320 nodes for the single passage of the guide vanes, 356478 nodes for the single passage of the runner and 3185349 for the complete draft tube, and consists, again, only of hexahedral elements. The largest grid of the cases presented and generally the most demanding, so now the wall-clock time on 4 GPUs (GeForce GTX 1080 Ti) is almost 54 minutes.

This case was run for the operating point that is described, based on the turbine's Hill chart, in table 4.7.

**Table 4.7:** *CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Operating point of the turbine.*

Variable	Value
H (m)	6.43
Q ( $\frac{m^3}{s}$ )	0.42
N (rpm)	1160.67



**Figure 4.15:** *CFD analysis of Kaplan guide vanes, runner and draft tube using PUMA. In the RSI connection between the runner and the draft tube the outlet boundary of the runner (green) corresponds to a single passage of the turbine, while the inlet boundary of the draft tube (red) corresponds to the complete component.*

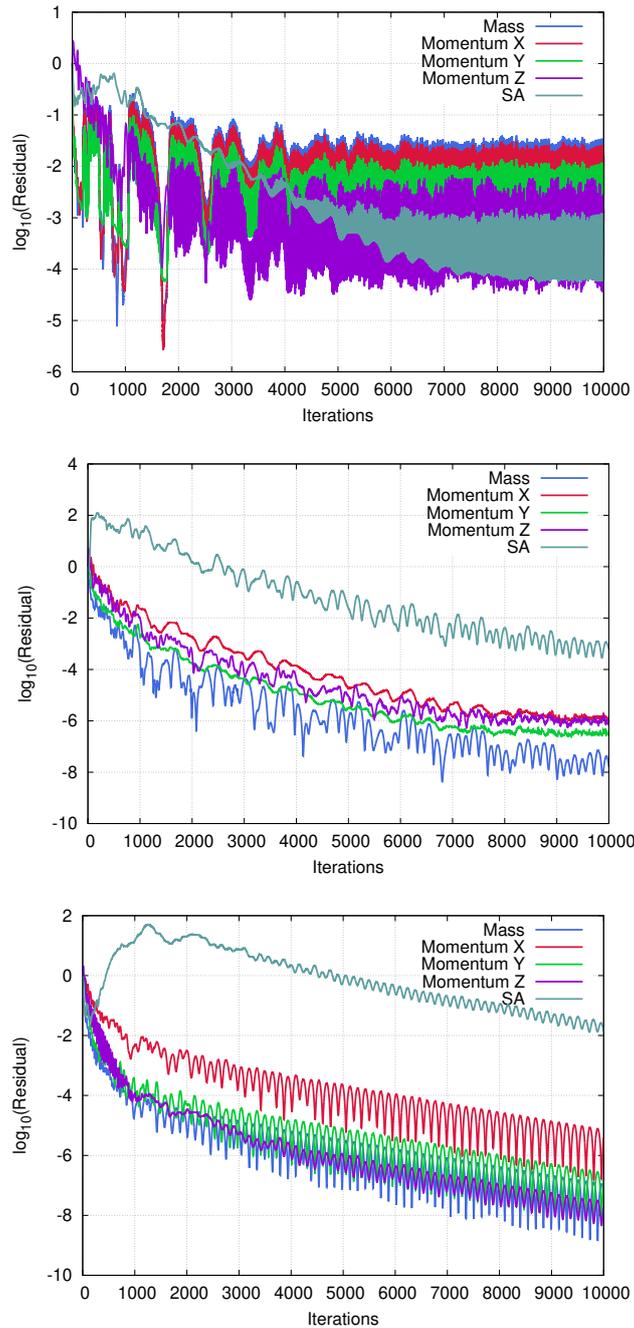
**Table 4.8:** *CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Imposed values of the flow field variables. The  $\theta_1$  flow angle results from the computation  $\theta_1 = 180 + \alpha$ , where  $\alpha$  is the known guide vane angle ( $30^\circ$ ), because the turbine revolves in counter-clockwise direction.*

Variable	Value
$p_{t,in} \left( \frac{m}{s} \right)$	117.67
$\theta_1 \text{ (deg)}$	210.0
$\theta_2 \text{ (deg)}$	90.0
$\frac{p_{s,out}}{\rho} \left( \frac{Pa}{kg/m^3} \right)$	54.67
$\frac{v_{t,in}}{\nu}$	20.0

As with the previous case, constant values for the total pressure along with the two flow angles are imposed at the inlet and a constant value for the static pressure at the outlet. The specific values for all the imposed quantities along the boundaries are presented in table 4.8.

The convergence rate of the residuals is presented below, for all three components. In each figure, the residual convergence rate is presented of each the PDEs that are being solved.

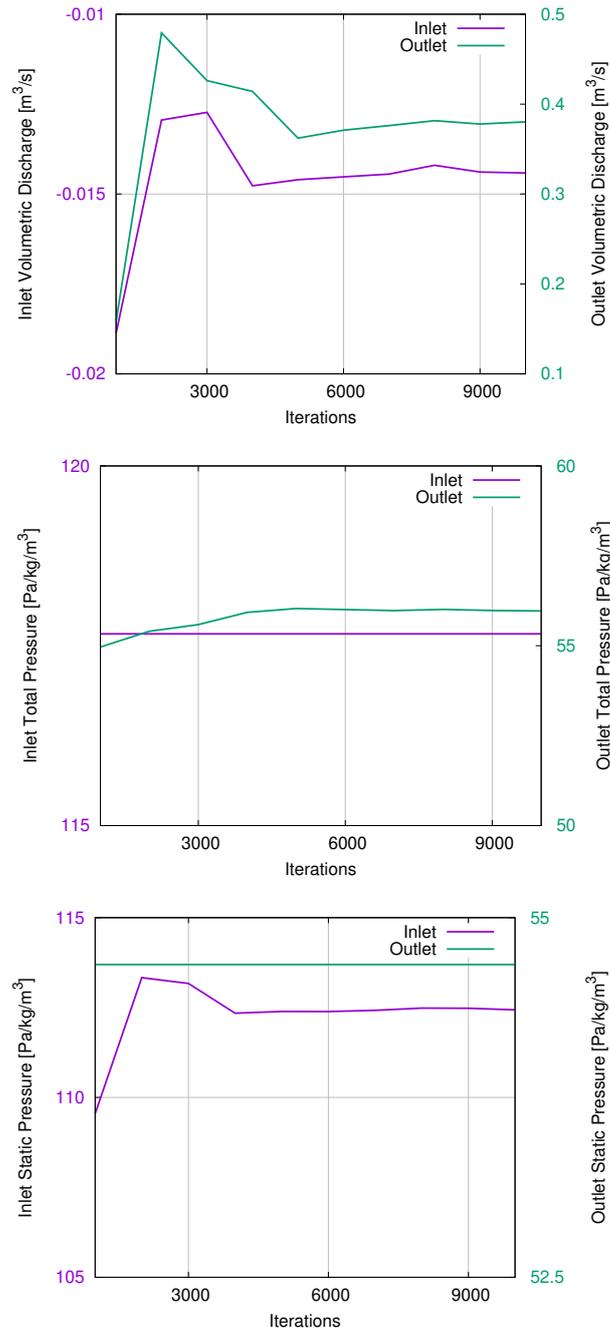
Furthermore, the convergence of certain integral quantities is presented here, namely the static and total pressure and the volumetric discharge at the inlet and outlet of



**Figure 4.16:** *CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Residual convergence of the five equations (continuity, three components of momentum and S-A model) for the guide vanes (top), the runner blade (middle) and the draft tube (bottom).*

the turbine:

Based on figures 4.16 and 4.17, it is clear that due to the complexity of the turbine

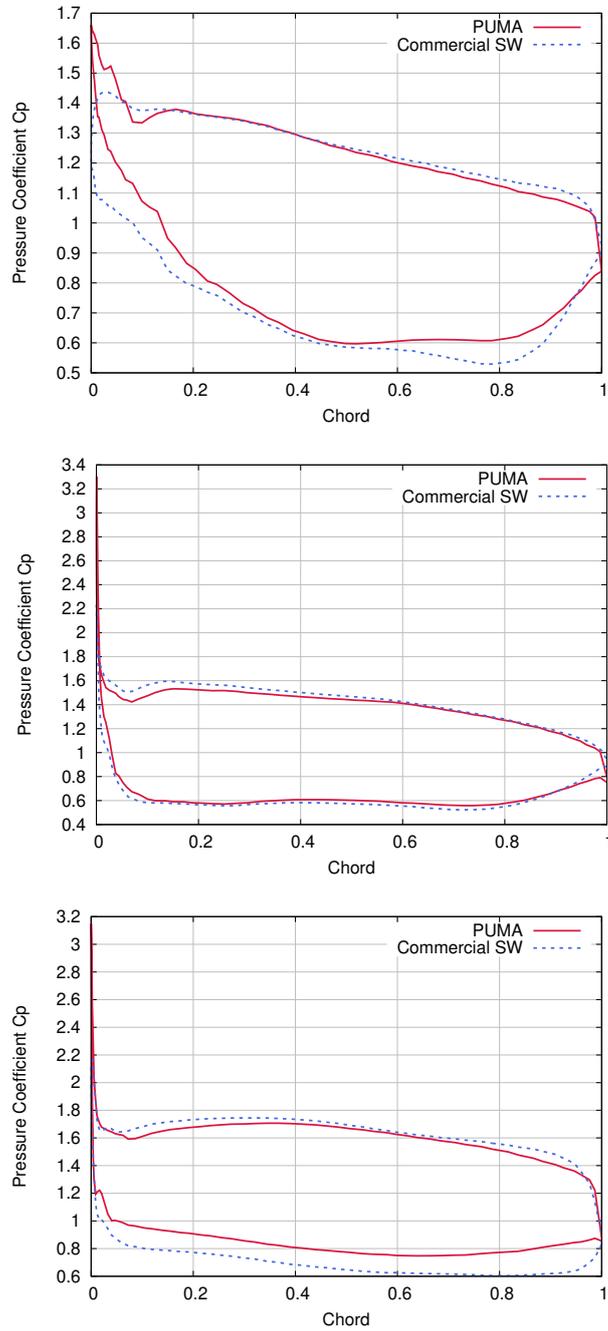


**Figure 4.17:** CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Convergence of integral quantities, such as mass flow (top), total (middle) and static (bottom) pressure over the inlet and outlet patches.

configuration, the required iterations to achieve a satisfying reduction in the residual magnitude order was raised to 10000 iterations (from the 5000 previously used).

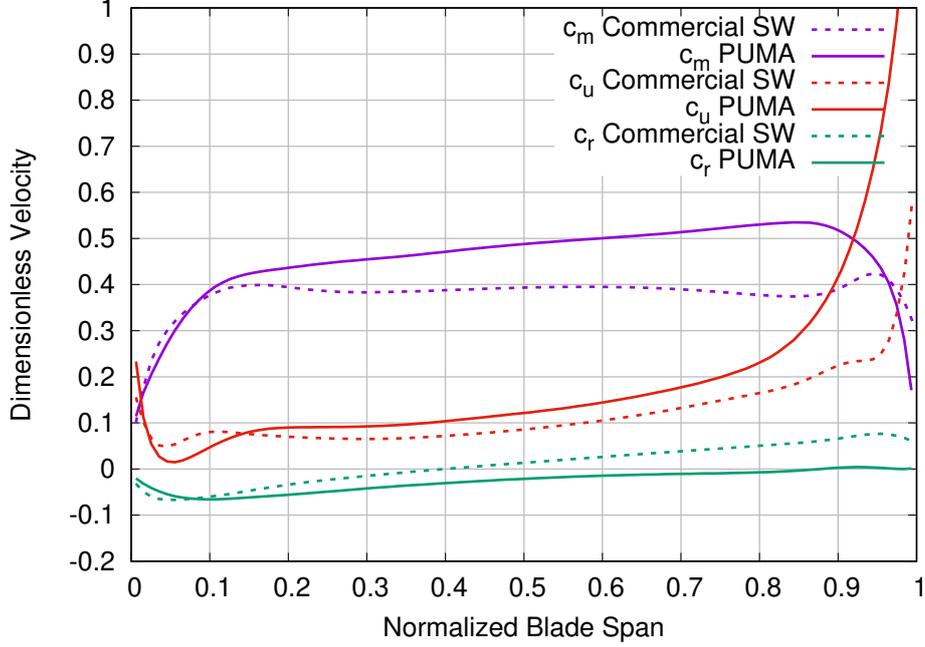
The  $C_p$  profiles are presented in figure 4.18. There are three profiles, one at the hub,

one at the midspan and one at the shroud of the runner's blade.



**Figure 4.18:** CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Pressure coefficient ( $C_p$ ) over the blade section at the hub (top), mid-span (middle) and shroud (bottom) positions.

The dimensionless profiles of the peripheral  $c_u^*$ , radial  $c_r^*$  and meridional  $c_m^*$  velocity components at the outlet of the runner are presented in figure 4.19.



**Figure 4.19:** *CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Dimensionless profiles at the outlet of the radial, peripheral and meridional velocity components.*

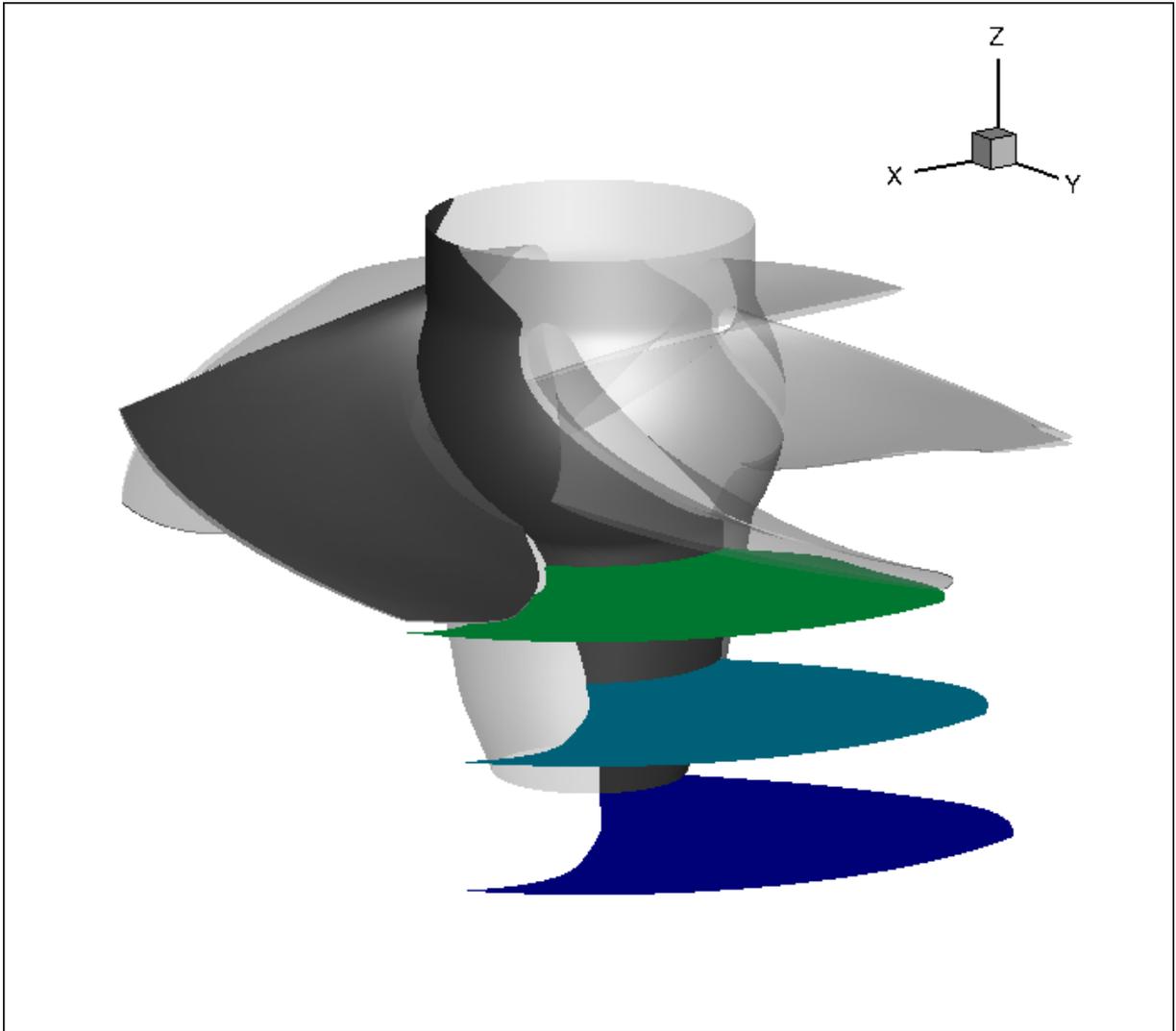
To further examine the effect that the presence of the draft tube has on the resulting flow field, in figure 4.21 the contours of the dimensionless absolute cylindrical velocity components ( $c_u^*$ ,  $c_r^*$  and  $c_m^*$ ) are presented. The contours are computed on three iso-z surfaces, one close to the TE of the blade, one at the interface of the runner passage and the draft tube and one intermediate position. The positions of the three surfaces are shown in figure 4.20.

Based on the comparative contours presented in fig. 4.21, it is apparent that as we approach the outlet of the runner passage the deviation between the results of the two solvers becomes more pronounced. As a result, the deviation presented in figure 4.19 is further justified.

Moreover, the percentile losses at the draft tube are computed based on equation

$$\delta p_{dt} = \frac{p_{tot,in} - p_{tot,out}}{p_{tot,in}} * 100 \quad (4.5)$$

where  $p_{tot,in}$  and  $p_{tot,out}$  are the area averaged values for the total pressure at the inlet and outlet of the draft tube. The losses computed by the PUMA solver are 14.3 %, while the losses computed by the commercial solver are 1.5 %. The glaring deviation between the two results further enhances the belief that the flow inside

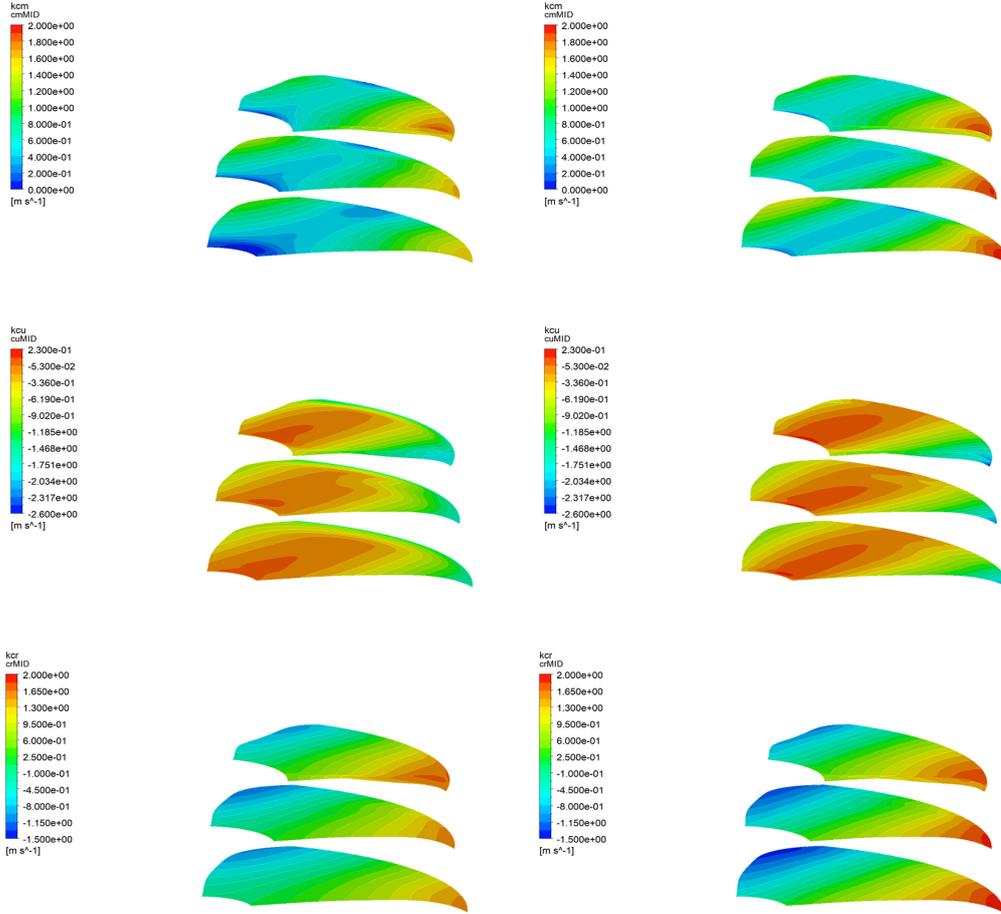


**Figure 4.20:** *CFD analysis of Kaplan guide vanes, runner and draft tube using PUMA. Positions of the three iso-z surfaces used for the contours.*

the draft tube of the turbine is difficult to predict for any solver.

Lastly, in table 4.9 some characteristic quantities, describing the turbine operation, are presented:

By examining figures 4.18 and 4.19 and 4.9, the deviation between the results produced from the two solvers is more pronounced than the previous two cases. This can be, mainly, attributed to the presence of the draft tube that raises the complexity of the configuration.



**Figure 4.21:** *CFD analysis of the Kaplan guide vanes, runner and draft tube using PUMA. Contours of the dimensionless absolute cylindrical velocity components:  $c_u^*$  (top),  $c_r^*$  (middle) and  $c_m^*$  (bottom) computed by the PUMA solver (left) and the commercial solver (right).*

**Table 4.9:** *CFD analysis of Kaplan guide vanes, runner and draft tube using PUMA. Overall results describing the hydraulic turbine function. As far as the efficiency  $\eta$ , the percentile deviation from the result produced by the commercial solver is presented.*

Quantity	PUMA	Commercial SW
H (m)	6.11	6.41
Q ( $\frac{m^3}{s}$ )	0.379	0.398
$\delta\eta(\%)$	-0.9	-
T (Nm)	173.67	193.41

## 4.4 Conclusions from the Comparison

In this section, some conclusions based on the comparative results of the cases that were tested between the PUMA and the commercial solver, are presented.

The convergence rate of both the residuals of the governing equations and the integral quantities for the three cases indicate that convergence is achieved well before the 5000 iterations. This fact enables the reduction of the computational cost by lowering the maximum number of iterations needed for each case.

In addition, the deviation observed between the  $C_p$  and dimensionless velocity profiles produced by the two solvers (figures 4.6, 4.7, 4.12, 4.13, 4.18, 4.19) is considered negligible. The only case that presents more pronounced deviation is that of the second Kaplan turbine including the draft tube. Although this case is the most computationally demanding, in terms of both grid size and turbine configuration, the results produced from the PUMA solver are almost the same as the ones produced by the commercial solver.

Concerning the characteristic quantities, it is fair to say that the computed values of the hydraulic head and the volumetric discharge are almost identical, while the computed values of the torque and the efficiency present a larger deviation. The deviation in the efficiency is directly linked to the corresponding deviation in the torque. As far the torque, the deviation can be attributed to the different way the wall boundary is treated in the solvers or even the turbulence model that is used.

In conclusion, it is safe to assume that the validation of the PUMA solver is successful, based on the comparative results presented.

## 4.5 Investigation of the Artificial Compressibility Parameter $\beta$

In this section, an investigation is carried out concerning the effect  $\beta$  has on the convergence rate of the PDEs and the results produced.

As stated in **chapter 2**, the PUMA solver employs the method of artificial compressibility to solve equations 2.1. In this method, proposed by Chorin [6] and then expanded by Turkel [7], the RANS equations formulated for incompressible fluid flows, are solved using a matrix  $\Gamma_{nm}^{-1}$  used for preconditioning. This matrix is defined as

$$\Gamma_{nm}^{-1} = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 \\ \frac{u_1^A + \alpha u_1^R}{\beta^2} & 1 & 0 & 0 \\ \frac{u_2^A + \alpha u_2^R}{\beta^2} & 0 & 1 & 0 \\ \frac{u_3^A + \alpha u_3^R}{\beta^2} & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

where  $u_k^A$  and  $u_k^R$  are the cartesian components of the absolute and relative velocities, respectively, and  $\alpha$  is a predefined parameter. The  $\beta$  parameter that is used inside

this matrix represents an artificial speed of sound. As a result, the effect of pseudo-compressibility is introduced. The method of artificial compressibility is employed in time-marching techniques for solving the RANS equations, because it enables a faster convergence rate.

In order to control the convergence rate, but also the stability of the system of PDEs that is formulated, the  $\beta$  parameter must be chosen as such that the condition number is as close to unity as possible.

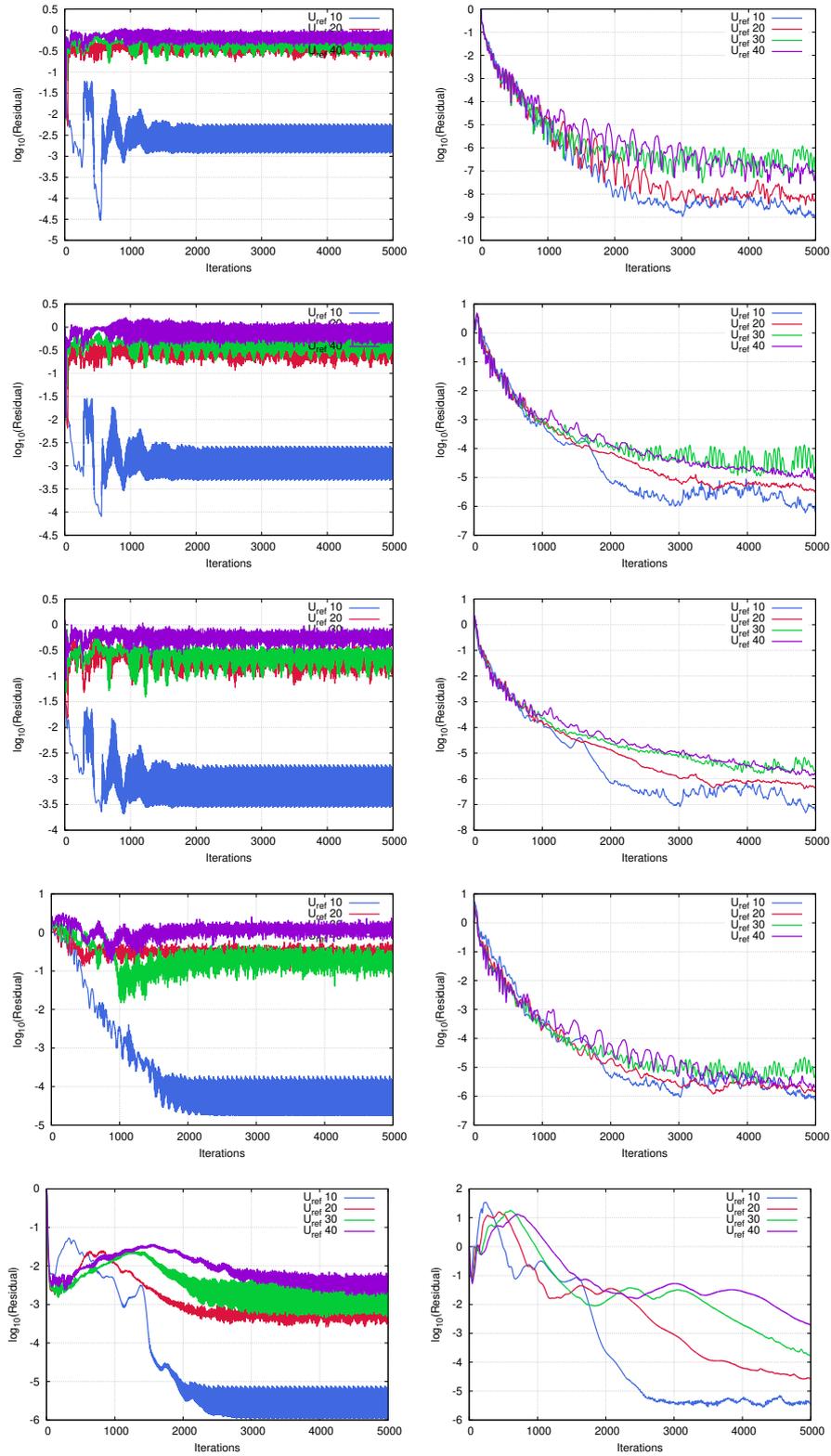
The technique, provided by PUMA, for controlling the value of  $\beta$  is by scaling an initial value of  $\beta$  with a scaling factor called reference velocity ( $U_{ref}$ ). The initial value of  $\beta$  is 3.33 and is scaled by multiplying it with the value of reference velocity ( $U_{ref}$ ) chosen by the user.

In order to investigate the effect of the different values of  $\beta$  on the convergence rate, the propeller turbine with the guide vanes and the runner is run for four different values of  $U_{ref}$  (10, 20, 30 and 40  $\frac{m}{s}$ ). The first value for  $U_{ref}$  is chosen to be close to a characteristic value of the absolute velocity magnitude of the turbine. This characteristic value is the result of the division  $\frac{Q}{A_{in}}$ , where  $Q$  is the flow rate and  $A_{in}$  is the inlet area of a single passage of the turbine. The division  $\frac{Q}{A_{in}}$  results in a characteristic absolute velocity magnitude of 4.89  $\frac{m}{s}$ . Thus the first value for  $U_{ref}$  is chosen to be 10  $\frac{m}{s}$ . The remaining three values for  $U_{ref}$  are chosen as multiples of 10 and this results to the three values being 20, 30 and 40  $\frac{m}{s}$ . In figure 4.22, the convergence rate of the residuals for both components for the four different values of  $U_{ref}$  is presented.

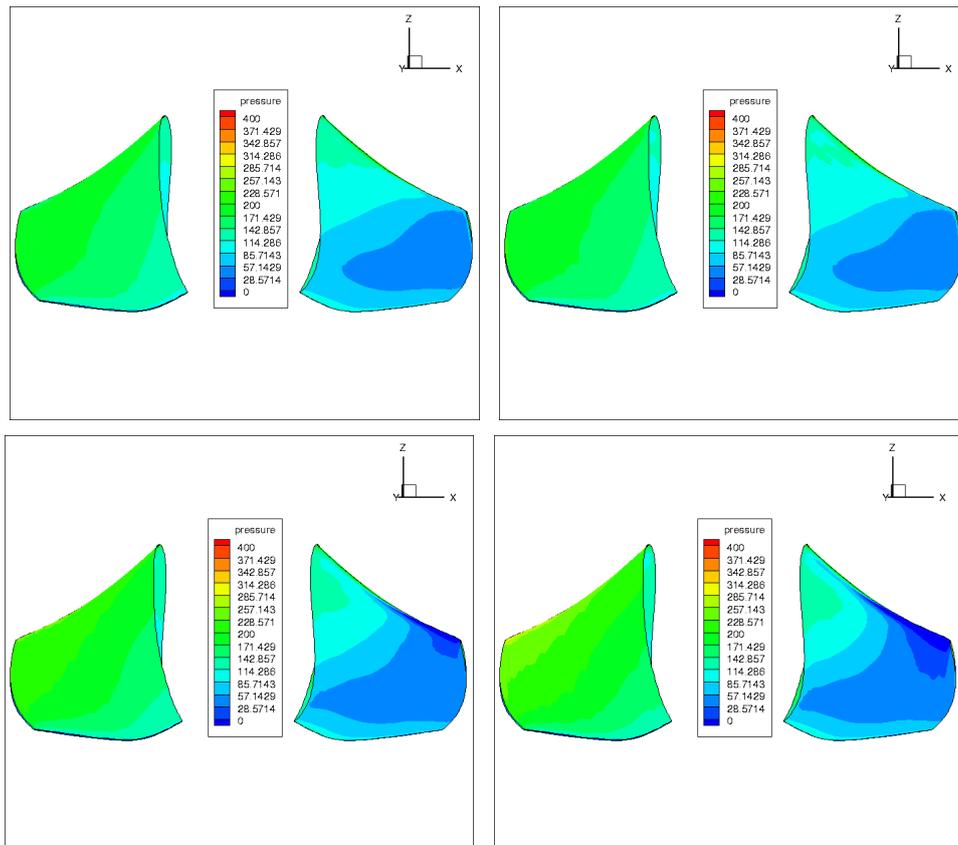
From figure 4.22, it can be inferred that as the value of  $U_{ref}$  increases, the reduction of the order of the residuals decreases and the convergence rate drops significantly. It is noteworthy to state that the same case with a reference velocity  $U_{ref}$  of 5  $\frac{m}{s}$  could not converge at all.

Of course, the resulting flow fields for all the values of  $U_{ref}$  must be compared in order to prove the independence of the results from the chosen value of  $U_{ref}$ . In order to do so, in figure 4.23, the resulting pressure fields are presented for the four different values of  $U_{ref}$ .

Based on figure 4.23, it seems to be a minor change in the resulting pressure fields, although in theory there shouldn't be any. Thus, the results produced are not entirely independent from the selection of  $\beta$ . Generally, it is good practice to set the value of  $U_{ref}$  as close as possible to a value of a characteristic absolute velocity magnitude, in order to achieve faster convergence rate.



**Figure 4.22:** Parametric investigation of the propeller turbine guide vanes and runner using PUMA. Convergence rate of the residuals for the guide vanes (left) and the runner (right) for different values of  $U_{ref}$  (10, 20, 30, 40  $\frac{m}{s}$ ).



**Figure 4.23:** Parametric investigation of the propeller turbine guide vanes and runner using PUMA. The resulting pressure fields of the runner for different values of  $U_{ref}$  (top-left to bottom-right, 10, 20, 30 and 40  $\frac{m}{s}$ ).



# Chapter 5

## Optimization of a Kaplan Runner Blade

The last step of this diploma thesis was to integrate the newly created unified toolchain in the optimization loop routinely used in Andritz Hydro. As of now, in the optimization loop a CFD solver for the Euler equations predicting inviscid flows for incompressible fluids was used.

This choice was made due to the low cost of solving the Euler equations, although the results produced are slightly worse concerning their accuracy. The solving speed of the flow is of crucial importance, since the optimization method that is used is the Evolutionary Algorithms. For the Evolutionary Algorithms, there is the necessity of evaluating a large set of candidate solutions, and for each solution the CFD solver needs to be computationally inexpensive.

For the present diploma thesis, the optimization tool EASY (Evolutionary Algorithm SYstem) was used for the optimization of a Kaplan runner blade. This tool, that was developed in PCOpt/NTUA.

### 5.1 Evolutionary Algorithms

The method used for the optimization of the turbine was the Evolutionary Algorithms (EAs). Evolutionary Algorithms are a stochastic population-based optimization method, meaning that depend on probabilistic methods for determining the optimal solution. One advantage of EAs is their ability to avoid local minima when searching the design space, precisely due to their probabilistic nature. The biggest drawback of EAs is the high wall-clock time.

An EA evaluates a population of candidate solutions. Specifically, a population of  $\mu$  candidate solutions or parents evolves into a population of  $\lambda$  new candidate solutions

or offspring. From this population of offspring a new population of  $\mu$  parents is chosen in order to create the next generation  $g$ . The best candidate solutions of each generation are called the elites, meaning that these solutions satisfy best the objective function. Generally, there are three sets in each generation  $g$  that an EA handles, the parents ( $S^{g,\mu}$ ), the offspring ( $S^{g,\lambda}$ ) and the elites ( $S^{g,e}$ ).

During the process of an EA, there are two processes that occur in order to produce the next generation. These are the mutation and the crossover. After elites are chosen in each generation, the two processes will take place in order to produce the next generation ( $g + 1$ ). Crossover is the procedure where the characteristics of the parents of the previous generation are combined in order to produce an offspring that presumably is even better than its parents, while mutation is the procedure where the characteristics the new offspring are randomly changed in order to explore other possible candidate solutions of the design space. With the combination of the two procedures two goals are achieved. With the crossover it is ensured that the so called good qualities of a candidate solution are passed on to the next generation giving a direction towards the optimal solution, while the mutation ensures that all the possibilities will be explored, avoiding any local minima.

In further analysis, the whole process that takes place in an EA is explained step by step here:

1. Some basic parameters are chosen, such as the population sizes for the parents ( $\mu$  and the offspring ( $\lambda$ ) or the mutation probability. Also, the members of the first generation (also called the zeroth generation) are chosen randomly in order to initialize the EA.
2. The  $\lambda$  members of the set  $S^{g,\lambda}$  are evaluated. Basically, in this step the CFD solver is called  $\lambda$  times, making this part of the algorithm the one with the highest computational cost.
3. The set of the elites  $S^{g,e}$  is updated with every member of  $S^{g,\lambda}$  that qualifies to be included, based on its value of the objective function, that the evaluation process produced.
4. From the elites set  $S^{g,e}$ , some members are randomly selected that replace members of  $S^{g,\lambda}$ . The members of  $S^{g,\lambda}$  that are being replaced, usually, are the ones that have a worse value for the objective function, thus ensuring that the next generation will yield better performance.
5. The set of parents of the next generation  $S^{g+1,\mu}$  is chosen by combining members of both the sets of the existing parents  $S^{g,\mu}$  and offspring  $S^{g,\lambda}$ .
6. Crossover and mutation take place, combining and changing characteristics of the parents of  $S^{g+1,\mu}$ , creating the  $\lambda$  offspring of the next generation  $S^{g+1,\lambda}$ .
7. The convergence criterion is applied, based on the total number of generations or the change in the current optimal solution. If convergence is not achieved,

steps 2 through 6 are repeated.

## 5.2 Design Variables and Blade Parameterization

In each optimization process, the design variables need to be defined first and foremost. These are the variables that are changed in each candidate solution of each generation of the EA and these are the variables that control the shape of the hydraulic turbine's runner blade. These variables come from the process known as blade parameterization.

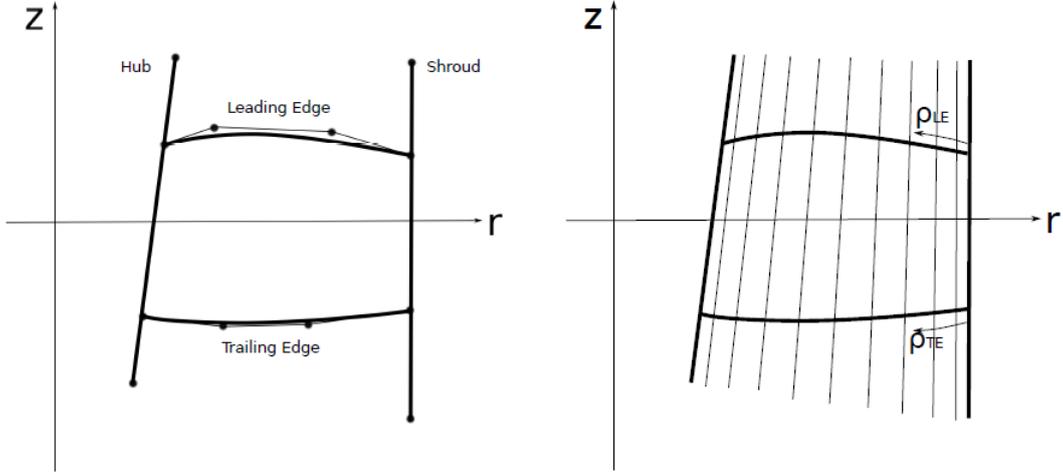
Blade parameterization is the process of describing the blade of the runner with a set of variables or parameters. For the optimization case presented in this chapter the method for parameterizing the blade is explained in short in this section, in order to understand the design variables used in the optimization.

The blade parameterization includes two steps. The first step is the parameterization of the 3D mean camber surface and the second step is the thickness distribution around the mean camber surface [1], [16].

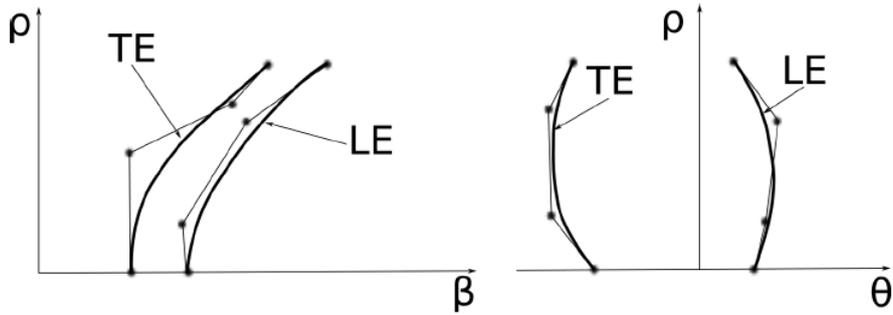
In order to parameterize the mean camber surface, first, its meridional projection has to be defined. The meridional projection is the resulting surface from projecting the 3D mean camber surface on the meridional plane, that is the plane that passes through the turbine's axis (i.e. the z-axis). This projected surface is constrained by 4 curves, namely the hub and shroud curves and the LE and TE curves. On this surface, a set of 2D iso-span lines are generated between the hub and shroud curves, as seen in figure 5.1. The LE, TE and iso-span curves are parameterized with the use of *Bézier* curves. The 3D shape of the 2D iso-span lines is defined by a set of parameters, namely  $\rho, \theta, \beta, \zeta$  and  $\mu$ . The new 3D lines that are generated comprise the main skeleton of the mean camber surface of the blade.

The five parameters mentioned are further explained:

- $\rho$  is defined along both the LE and TE as the normalized meridional projection of the arc-length of the edge, and takes values between 0, at the shroud, and 1, at the hub.
- $\theta$  is the wraparound angle and defines the angular positions of the LE and TE. The  $\theta$  distributions along the LE and TE are defined by *Bézier* curves, as seen in figure 5.2 (right).
- $\beta$  is defined as the angle between the local tangent to the mean-camber surface and the tangent to the z-centered circle through this point. Therefore, the  $\beta$  distributions define the mean-camber surface metal angles across the previously defined LE and TE. The  $\beta$  distributions along the LE and TE are defined by *Bézier* curves, as seen in figure 5.2 (left).



**Figure 5.1:** Left: Meridional projection of the mean camber surface of the turbine blade. Right: The 2D iso-span lines distributed between shroud and hub. Definition of  $\rho$  for the LE and TE [1].



**Figure 5.2:** Left:  $\beta(\rho)$  distribution. Right:  $\theta(\rho)$  [1].

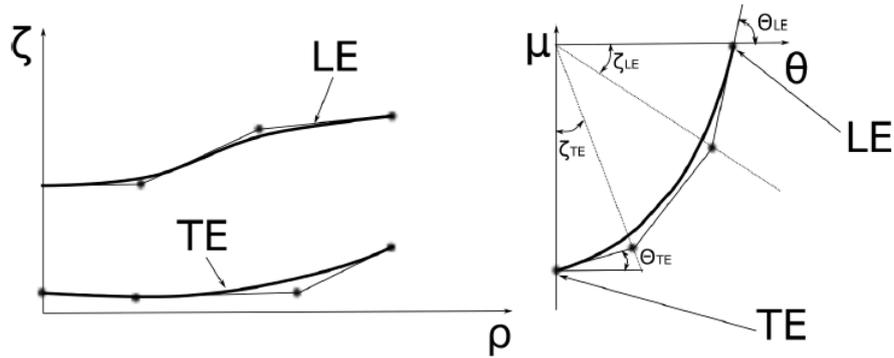
- $\mu$  is defined through the conformal mapping  $\Phi$ , as follows

$$\Phi : (r, z) \rightarrow \mu, \quad \mu = \int \frac{1}{r} dl \quad (5.1)$$

where  $l$  denotes the arc-length of the meridional projection of the iso-span line. This conformal mapping performs the angle-preserving transformation of the iso-span lines from the cylindrical  $(r, z, \theta)$  to the  $(\mu, \theta)$  coordinate system. This process is presented in figure 5.3

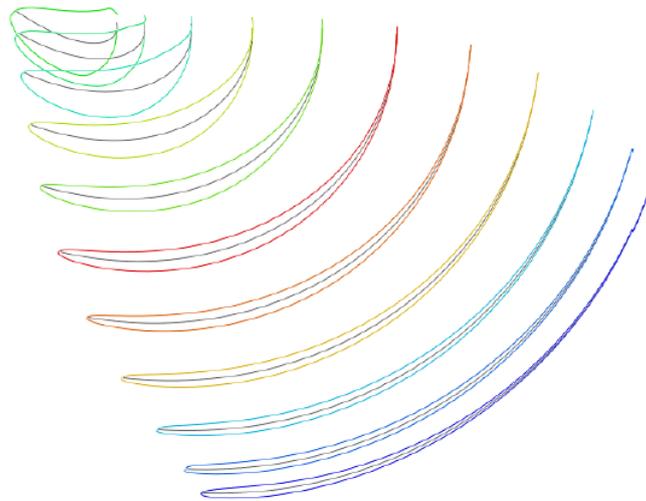
- $\zeta$  is defined for both the LE and TE and helps define the two internal control points for all the iso-span lines. In other words, the  $\zeta$  parameter acts as control system for the curvature of the mean camber surface.

The last step of the blade parameterization is the definition of the thickness distribution around the mean camber surface. The thickness profiles are defined again with the use of *Bézier* curves and are superimposed onto the mean camber surface.



**Figure 5.3:** Left:  $\zeta(\rho)$  distributions for the LE and TE. Right: Iso-span line expressed in the  $(\mu, \theta)$  coordinate system [1].

The final 3D iso-span lines with the thickness profiles superimposed is seen in figure 5.4.



**Figure 5.4:** The 3D iso-span lines with the thickness profiles superimposed on them [1].

From the five parameters described above (i.e.  $\rho$ ,  $\theta$ ,  $\beta$ ,  $\zeta$  and  $\mu$ ), the  $\beta$  distributions for the metal angle, which are described with a set of 30 control points (15 control points for the LE Bézier curve and 15 control points for the TE Bézier curve), are chosen. This set of 30 values act as the design variables for the optimization. The choice, generally, varies from case to case and is left to the designer's discretion.

## 5.3 Objective Functions

For the optimization loop of a Kaplan turbine runner there are two objective functions that, generally, are applied, thus performing multi-objective optimization (MOO). The two objective functions are the maximization of the turbine's efficiency and the maximization of the minimum static pressure to avoid the onset of cavitation.

### 5.3.1 Efficiency

The first and most obvious objective function is the one concerning the maximization of the hydraulic turbine's efficiency, as computed by eq. 4.4. This objective function is expressed as

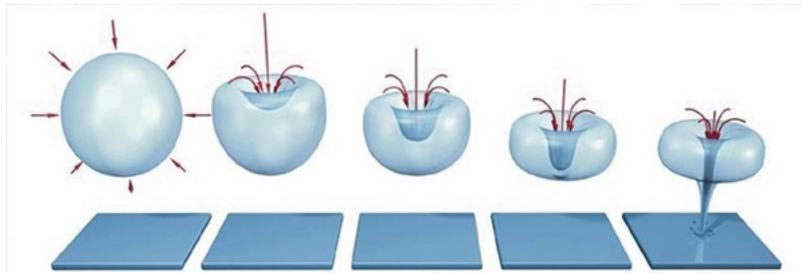
$$F_1 = \min(-\eta) \quad (5.2)$$

Of course, this objective function is the most important of the two, because the hydraulic turbine's efficiency defines its performance and is directly connected with the economic viability of its construction.

Equation 5.2 is expressed in this form because the EASY software handles only minimizations of objective functions.

### 5.3.2 Cavitation

The second objective function concerns the cavitation that may occur on the runner's surface. Cavitation is a phenomenon that occurs when the static pressure  $p_s$  drops below a certain value ( $p_{cav}$ ), at which point the liquid fluid (i.e. water) evaporates creating little bubbles on the blade's surface. Those bubbles, then, implode and create an intense shock wave that damages the metal surface of the blade. In figure 5.5, this phenomenon is presented graphically.



**Figure 5.5:** Graphical representation of the process of cavitation [17].

Cavitation results in huge reductions of the turbine's efficiency, due to change in the blade's shape, as seen in figure 5.6.



**Figure 5.6:** *The metal surface of a turbine's runner after extensive damage from cavitation [18].*

In order to prevent this, the goal is to maximize the value of the minimum pressure of the blade's surface. This is also expressed as

$$F_2 = \min(-p_{s,min}) \quad (5.3)$$

## 5.4 Constraints

For the optimization case a constraint concerning the outlet velocity profiles for the dimensionless peripheral  $c_u^*$  and meridional  $c_m^*$  velocity components is imposed. The constraint is expressed as the deviation of the resulting velocity profiles from a set of target velocity profiles. These target profiles are defined at the inlet of the draft tube and ensure that the draft tube produces minimum losses.

Mathematically, this constraint is expressed as

$$C_m \leq 0.04 \quad (5.4)$$

$$C_u \leq 0.04 \quad (5.5)$$

where

$$C_m = \int_{A_{out}} (c_m^* - c_{m,tar}^*)^2 dr \quad (5.6)$$

$$C_u = \int_{A_{out}} (c_u^* - c_{u,tar}^*)^2 dr \quad (5.7)$$

The constraint is expressed in this form because EASY handles constraints of the type "less or equal to" ( $\leq$ ).

## 5.5 Optimization Results

The optimization case presented is that of a Kaplan hydraulic turbine, including the guide vanes and the runner blade. For the optimization, the design variables that are chosen are the  $\beta$  angle distributions at the LE and TE. For the  $\beta$  distributions 30 control points are used in total, 15 for the LE and 15 for the TE. The minimum and maximum values for each design variable are presented in table 5.1

**Table 5.1:** *Optimization of a Kaplan runner using EASY and PUMA. Minimum and maximum values for each of the 30 design variables, where  $\beta_i$  is the  $i$ -th control point of the  $\beta$  distribution. The minimum and maximum values of each variable result from a percentile offset of  $\pm 5\%$  applied on the  $\beta_i$  values of the initial geometry.*

Variable	Min. Value	Max. Value	Variable	Min. Value	Max. Value
$\beta_1$	65.6812	72.595	$\beta_{16}$	45.2388	50.0008
$\beta_2$	51.8671	57.3267	$\beta_{17}$	40.5096	44.7738
$\beta_3$	41.1341	45.4641	$\beta_{18}$	30.7535	33.9907
$\beta_4$	28.6676	31.6852	$\beta_{19}$	22.0966	24.4226
$\beta_5$	23.5894	26.0726	$\beta_{20}$	22.5649	24.9401
$\beta_6$	55.3698	61.1982	$\beta_{21}$	42.6302	47.1176
$\beta_7$	46.7945	51.7203	$\beta_{22}$	35.7783	39.5445
$\beta_8$	35.1295	38.8273	$\beta_{23}$	28.7235	31.7471
$\beta_9$	27.0282	29.8732	$\beta_{24}$	21.8111	24.1071
$\beta_{10}$	22.9273	25.3407	$\beta_{25}$	22.335	24.686
$\beta_{11}$	45.8818	50.7114	$\beta_{26}$	35.0848	38.778
$\beta_{12}$	40.3341	44.5797	$\beta_{27}$	30.366	33.5624
$\beta_{13}$	33.17	36.6616	$\beta_{28}$	24.4629	27.0379
$\beta_{14}$	23.9992	26.5254	$\beta_{29}$	19.1359	21.1503
$\beta_{15}$	22.3561	24.7093	$\beta_{30}$	21.5761	23.8473

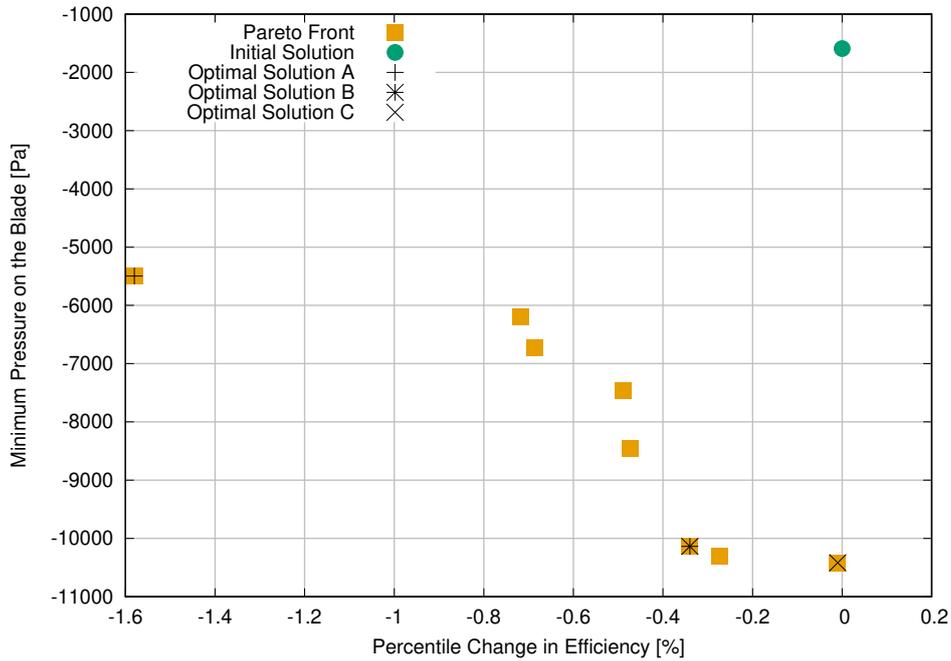
Moreover, some additional parameters for the EASY set-up are presented in table 5.2

The EA was run for 21 generations and in figure 5.7 the Pareto front for the two-objective optimization case is presented.

From figure 5.7 three optimal solutions are chosen for comparing to the initial solution, so as to best describe the results of a multi-objective optimization. The two extreme solutions were chosen and one at the middle, in order to comprehend how each solution depends on its values of the objective functions. Of course, from all of the computed optimal solutions the designer is called to choose which one will be

**Table 5.2:** Optimization of a Kaplan runner using EASY and PUMA. Basic set-up parameters for the EASY software and their corresponding values.

Parameter	Symbol	Value
Parent Population Size	$\mu$	15
Offspring Population Size	$\lambda$	30
Elite Population Size	$e$	20
Crossover Probability	-	0.9
Maximum Number of Evaluations	-	12000
Variable Coding	-	Real



**Figure 5.7:** Optimization of a Kaplan runner using EASY and PUMA. The Pareto front of the two-objective optimization. Each square point represents an optimal solution with its corresponding values for the two objective functions ( $F_1$  and  $F_2$ ). Also, in the upper-right corner is the initial solution (filled green circle).

further developed or even manufactured. It is useful to remind that, as the objective functions are defined counter-intuitively, the values on the axes of the Pareto front become more desirable as they decrease.

In figure 5.8, the pressure coefficient  $C_p$  is presented for the three optimal solutions that are chosen compared to initial solution. It is apparent that there is a shift upwards for the three optimal solutions relative to the initial solution's  $C_p$  distribution, meaning that the Kaplan runner was optimized with respect to that objective.

In table 5.3, the percentile increase in the efficiency of each optimal solution compared to the initial solution is presented.

**Table 5.3:** Optimization of a Kaplan runner using EASY and PUMA. Percentile increase in efficiency for each optimal solution compared to the initial.

Optimal Solution ID	Percentile change in Efficiency (%)
Optimal Solution A	+1.58
Optimal Solution B	+0.34
Optimal Solution C	+0.01

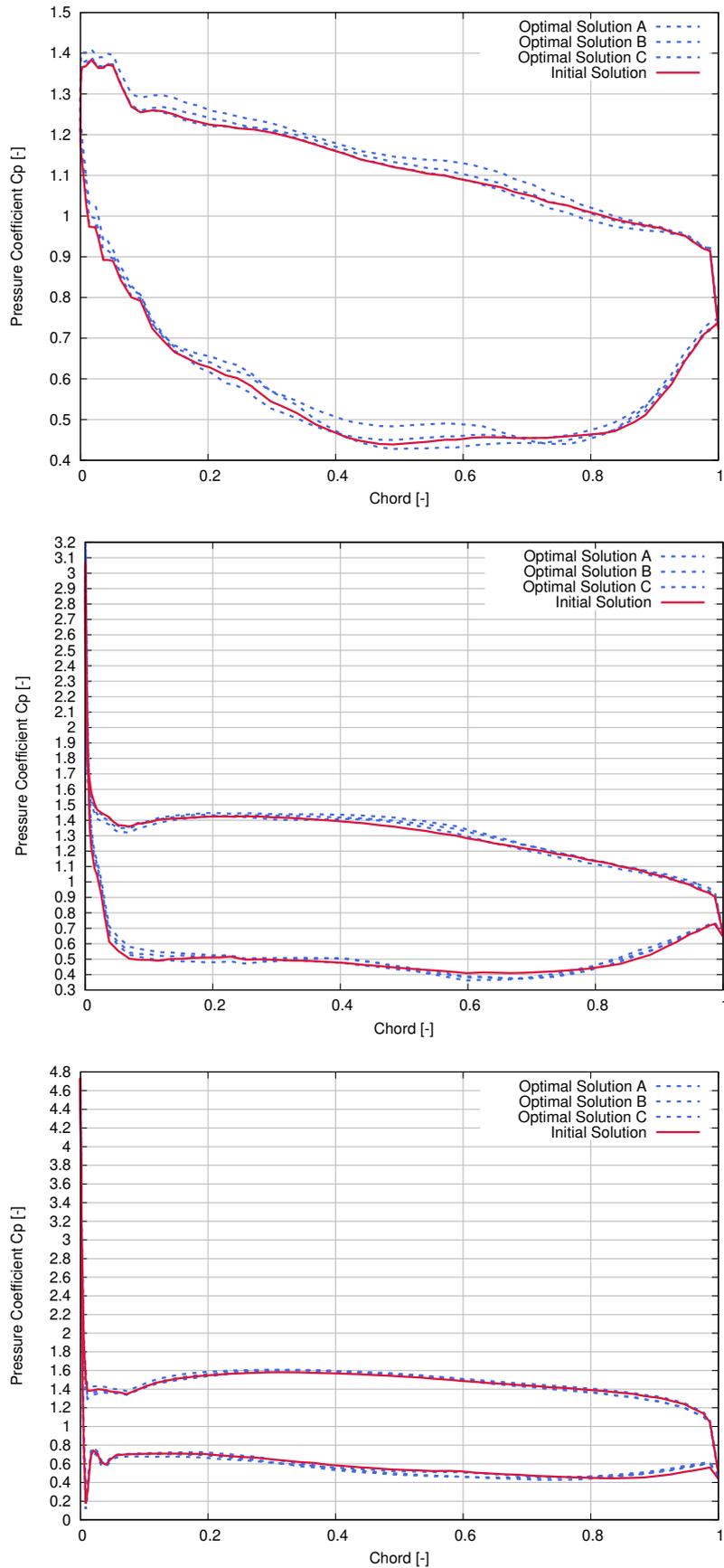
Here, it is clear that, depending on the optimal solution that is chosen, one can enjoy minor (+0.2%) to major (+1.6%) increases in efficiency.

Finally, in figure 5.9 the resulting velocity profiles of the dimensionless peripheral ( $c_u^*$ ) and meridional ( $c_m^*$ ) velocity components are presented, compared to the target velocity profiles of the constraint, while in table 5.4 the value of the constraint as it is calculated from eq. 5.4 is presented for each one of the optimal solutions.

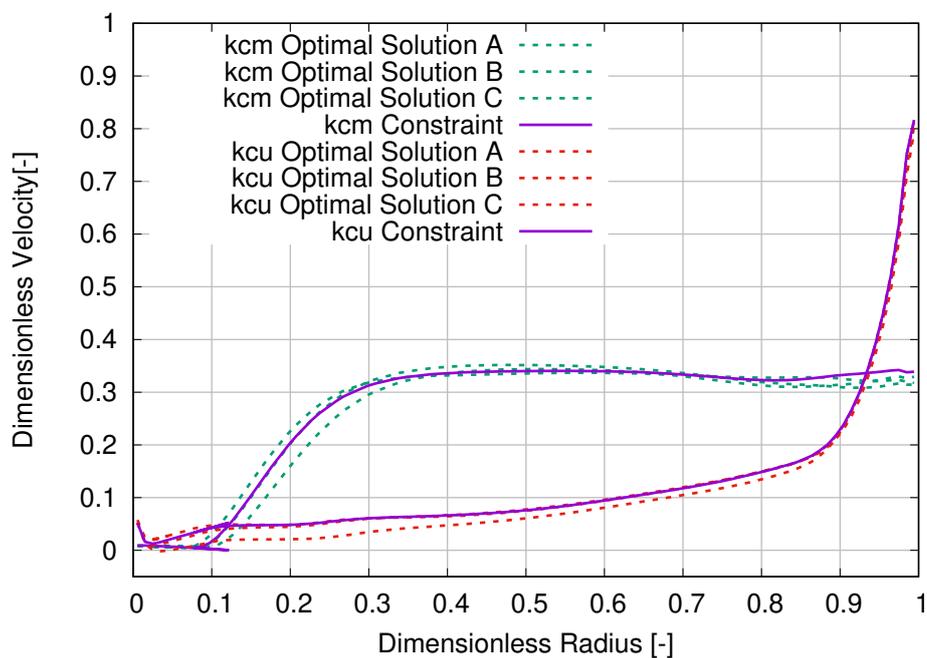
**Table 5.4:** Optimization of a Kaplan runner using EASY and PUMA. The resulting constraint's value as it is calculated from eq. 5.4 for each optimal solution.

Optimal Solution ID	Value of the $c_m^*$ constraint	Value of the $c_u^*$ constraint
Optimal Solution A	0.01134	0.01688
Optimal Solution B	0.00881	0.00287
Optimal Solution C	0.00852	0.00365

As it is clear from the results presented the optimization case that is tested produces acceptable results, while simultaneously respecting the constraints that are imposed.



**Figure 5.8:** Optimization of a Kaplan runner using EASY and PUMA. The  $C_p$  distribution for the three optimal solutions and the initial geometry on three separate positions, the hub (top), the mid-span (middle) and the shroud (bottom) of the blade.



**Figure 5.9:** Optimization of a Kaplan runner using *EASY* and *PUMA*. The resulting dimensionless velocity profiles of the peripheral ( $c_u^*$ ) and meridional ( $c_m^*$ ) velocity components at the runner outlet compared to their respective target profiles imposed from the constraint. All the results are presented for the three optimal solutions.

# Chapter 6

## Overview and Conclusions - Future Work Ideas

### 6.1 Overview and Conclusions

The work presented in this diploma thesis concerned the validation and integration of the PUMA CFD solver, developed by PCOpt/NTUA, into the industrial workflow used by Andritz Hydro for analysis and optimization.

After organically incorporating the PUMA solver in the industrial workflow of Andritz Hydro and testing the validity of the results produced by PUMA, the following conclusions can be drawn:

- The smooth integration of the PUMA solver into the industrial workflow, using the various interfacing tools that were programmed, is successful. The result of the work is a unified toolchain that is user-friendly and is being used in present time by the designers in Andritz Hydro.
- The comparison of the PUMA solver with a commercial solver is also deemed a success, as the deviation between the results of the two solvers was negligible. This encourages many more designers, working in Andritz Hydro, to try running their simulations using the PUMA solver.
- Achieving small wall-clock time for the various cases. Specifically, the value of the wall-clock time required for every 1000 grid nodes ranges from 0.01 to  $0.02 \frac{\text{min}}{1000 \text{ nodes}}$ , on four GPUs (GeForce GTX 1080 Ti).
- The use of the PUMA solver as an evaluation tool for EASY seems to be a viable option, as the results of the optimization are satisfying and the computational cost is relatively low. This opens up the possibility of using the optimization to an even greater extent than it was previously being used.

## 6.2 Proposals for Future Work

As far as ideas for future work are concerned, the following are proposed:

- Further development of the grid converter tools to include the ability to handle grids with different type of elements, apart from hexahedral.
- The integration of the grid converter tools inside the PUMA solver.
- Further testing of the PUMA solver by examining even more complex turbine configurations, such as the inclusion of a spiral casing, in order to explore the capabilities of PUMA.
- The simulation and testing of truly unsteady incompressible fluid flows inside the hydraulic turbines.

**Part II**  
**Thesis in Greek**



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Μηχανολόγων Μηχανικών  
Τομέας Ρευστών  
Εργαστήριο Θερμικών Στροβιλομηχανών  
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής  
& Βελτιστοποίησης

Ένταξη του GPU Επιλύτη Ροής PUMA σε  
Ολοκληρωμένο Εργαλείο Ανάλυσης και  
Βελτιστοποίησης Στροβιλομηχανών της Βιομηχανίας.  
Δοκιμές και Πιστοποίηση.

Διπλωματική Εργασία

Ηλίας Ιωάννης Τσόπελας

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2018



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Μηχανολόγων Μηχανικών  
Τομέας Ρευστών  
Εργαστήριο Θερμικών Στροβιλομηχανών  
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής  
& Βελτιστοποίησης

**Ένταξη του GPU Επιλύτη Ροής PUMA σε  
Ολοκληρωμένο Εργαλείο Ανάλυσης και  
Βελτιστοποίησης Στροβιλομηχανών της Βιομηχανίας.  
Δοκιμές και Πιστοποίηση.**

Διπλωματική Εργασία

**Ηλίας Ιωάννης Τσόπελας**

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2018

**Περίληψη**

Η διπλωματική εργασία παρουσιάζει την ένταξη και πιστοποίηση του επιλύτη ροής PUMA, που αναπτύχθηκε από τη Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης (ΜΠΥΡ&Β) της Σχολής Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, στο βιομηχανικό εργαλείο ανάλυσης και βελτιστοποίησης στροβιλομηχανών της Andritz Hydro, όπου και διεξάχθηκε μεγάλο μέρος της διπλωματικής εργασίας.

Ο επιλύτης PUMA τρέχει σε συστοιχίες από GPUs και χρησιμοποιεί την κεντροκομβική μέθοδο των πεπερασμένων όγκων για τη επίλυση των εξισώσεων Navier-Stokes σε συνδυασμό με το μοντέλο τύρβης των Spalart-Allmaras. Επίσης, εφαρμόζεται η μέθοδος της τεχνητής συμπίεστότητας με την οποία εξασφαλίζεται ταχύτερη σύγκλιση.

Προγραμματίστηκαν διάφορα εργαλεία με σκοπό την οργανική ένταξη του PUMA επιλύτη στο βιομηχανικό εργαλείο ανάλυσης και βελτιστοποίησης της Andritz Hydro. Αυτά τα εργαλεία περιλαμβάνουν δυο μετατροπείς πλεγμάτων, έναν προεπεξεργαστή και έναν μετεπεξεργαστή.

Στη συνέχεια, ο επιλύτης PUMA συγκρίνεται με έναν εμπορικό επιλύτη για να γίνει η πιστοποίησή του. Η σύγκριση βασίζεται σε τρεις περιπτώσεις υδροστροβίλων, με κάθε περίπτωση να παρουσιάζει διαφορετική πολυπλοκότητα. Οι τρεις περιπτώσεις περιλαμβάνουν τον δρομέα ενός στροβίλου Francis, τα οδηγία πτερύγια και τον δρομέα ενός στροβίλου τύπου προπέλα και, τέλος, το συνδυασμό οδηγών πτερυγίων, δρομέα και αγωγού εξόδου ενός στροβίλου Kaplan.

Τέλος, ο PUMA επιλύτης χρησιμοποιείται ως εργαλείο αξιολόγησης μέσα στο βρόχο βελτιστοποίησης της Andritz Hydro. Η μέθοδος βελτιστοποίησης των Εξελικτικών Αλγορίθμων χρησιμοποιείται, όπως εφαρμόζεται από το εργαλείο EASY, το οποίο αναπτύχθηκε από τη ΜΠΥΡ&Β/ΕΜΠ. Το σχήμα του πτερυγίου ενός δρομέα Kaplan βελτιστοποιείται με στόχο τη μεγιστοποίηση του βαθμού απόδοσης του υδροστροβίλου και την αποφυγή εμφάνισης σπηλαίωσης.



## Ακρωνύμια

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΜΠΥΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
ΜΔΕ	Μερικές Διαφορικές Εξισώσεις
GPU	Graphics Processing Unit



# Κεφάλαιο 7

## Εκτενής Περίληψη στα Ελληνικά

### 7.1 Εισαγωγή

Η διπλωματική εργασία ασχολείται με την ένταξη και την πιστοποίηση του GPU επιλύτη ροής PUMA, που αναπτύχθηκε από τη ΜΠΥΡ&B/ΕΜΠ, στο βιομηχανικό εργαλείο ανάλυσης και βελτιστοποίησης στροβιλομηχανών της Andritz Hydro. Η ένταξη στο προϋπάρχον σύστημα από εργαλεία της βιομηχανίας, απαιτεί τον προγραμματισμό προεπεξεργαστή και μετεπεξεργαστή, εργαλείων υποβοήθησης της ένταξης, καθώς επίσης και εκτενή πιστοποίηση των αποτελεσμάτων πάνω σε περιπτώσεις-προβλήματα, τα οποία παρέχονται από την Andritz Hydro, όπου μεγάλο μέρος της διπλωματικής εργασίας έλαβε χώρα. Τέλος, η καινούρια ενοποιημένη σειρά από εργαλεία χρησιμοποιείται για την βελτιστοποίηση ενός δρομέα Kaplan.

### 7.2 Ο Επιλύτης Ροής PUMA

Για την ανάλυση της ροής μέσα στους υδροστροβίλους χρησιμοποιείται ο GPU επιλύτης PUMA, που αναπτύχθηκε από τη ΜΠΥΡ&B/ΕΜΠ. Το βασικό προτέρημα αυτού του επιλύτη είναι η δυνατότητά του να τρέχει σε συστοιχίες από κάρτες γραφικών (GPUs), μειώνοντας το υπολογιστικό κόστος σημαντικά [4]. Ο επιλύτης PUMA επιλύει αριθμητικά τις τριδιάστατες μέσες χρονικά κατά Reynolds εξισώσεις Navier-Stokes χρησιμοποιώντας τη μέθοδο των κεντροκομβικών πεπερασμένων όγκων, παρέχοντας δεύτερης τάξης ακρίβεια, για ροές ασυμπίεστου ρευστού. Επιπρόσθετα, ο επιλύτης PUMA χρησιμοποιεί το μοντέλο τύρβης των Spalart-Allmaras (S-A) [8] και εφαρμόζεται η μέθοδος της τεχνητής συμπίεστότητας για την επιτάχυνση του ρυθμού της συγκλίσης [6], [7].

## 7.3 Εργαλεία Υποβοήθησης της Ένταξης του Επιλύτη PUMA

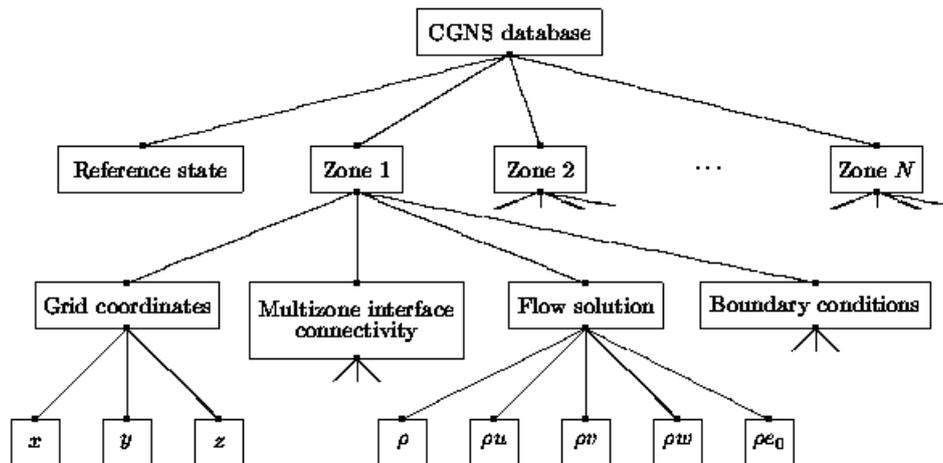
Τα εργαλεία που χρειάζεται να προγραμματιστούν για την ομαλή ένταξη του PUMA στο βιομηχανικό εργαλείο ανάλυσης και βελτιστοποίησης στροβιλομηχανών της Andritz Hydro περιλαμβάνουν τους δυο μετατροπείς του τύπου των πλεγμάτων, έναν προεπεξεργαστή, κάποιες τροποποιήσεις στον μετεπεξεργαστή και το συντονιστικό εργαλείο που εναρμονίζει την εκτέλεση όλων των υπολοίπων εργαλείων.

### 7.3.1 Μετατροπείς Πλεγμάτων

Αρχικά, τα εργαλεία που χρειάστηκε να προγραμματιστούν είναι οι δυο μετατροπείς πλεγμάτων που μετατρέπουν τα τριδιάστατα πλέγματα, που δημιουργούνται από τον πλεγματοποιητή, που αναπτύχθηκε και χρησιμοποιείται από την Andritz Hydro, από το ένα τύπο (CGNS) στον άλλο (LTT), και αντίστροφα. Αφενός, το πρώτο εργαλείο μετατρέπει το αρχείο .cgns, που περιέχει την επιθυμητή γεωμετρία, στο πρότυπο LTT το οποίο απαιτείται από τον επιλύτη PUMA. Αφετέρου, ο δεύτερος μετατροπέας λαμβάνει το αρχείο εξόδου από τον επιλύτη PUMA, το οποίο είναι ένα αρχείο τύπου Tecplot (.plt), το οποίο περιέχει την προκύπτουσα ροή αποθηκευμένη στους κόμβους του πλέγματος, και το μετατρέπει σε ένα αρχείο .cgns το οποίο μπορεί να χρησιμοποιηθεί για μετεπεξεργασία. Αυτό το καινούριο αρχείο .cgns, περιέχει πλέον τόσο το πλέγμα όσο και την προκύπτουσα ροή μέσα στον υδροστρόβιλο.

#### Ο Τύπος CGNS

Σύμφωνα με τον τύπο αρχείου CGNS, οι πληροφορίες μέσα στο αρχείο οργανώνονται σε ένα σύνολο "κόμβων", οι οποίοι ιεραρχούνται σχηματίζοντας μια δενδροειδή δομή. Συγκεκριμένα, κάθε "κόμβος" αφορά σε μια ζώνη του πλέγματος (π.χ. την είσοδο ή την έξοδο του στροβίλου κλπ.) και περιέχει όλες τις πληροφορίες (π.χ. μεταβλητές ροής, στοιχεία του πλέγματος κλπ.) που αφορούν τη ζώνη αυτή [14], [15]. Σχηματικά αυτή η δομή παρουσιάζεται στο σχήμα 7.1.



Σχήμα 7.1: Γραφική αναπαράσταση της δενδροειδούς δομής του αρχείου *.cgns*.

## Ο Τύπος LTT

Ο Τύπος LTT περιλαμβάνει ένα σύνολο από τρία αρχεία. Αυτά τα αρχεία είναι:

- Το αρχείο *.nod*, το οποίο περιέχει το συνολικό αριθμό κόμβων και τις  $x$ ,  $y$ ,  $z$  συντεταγμένες όλων των κόμβων. Παράδειγμα ενός αρχείου *.nod* παρουσιάζεται στο σχήμα 7.2.
- Το αρχείο *.hyb*, το οποίο περιέχει το συνολικό αριθμό πλεγματικών στοιχείων κάθε είδους (τετράεδρα, πυραμίδες, πρίσματα και εξαέδρα) και τον αύξοντα αριθμό των κόμβων που απαρτίζουν κάθε στοιχείο. Παράδειγμα ενός αρχείου *.hyb* παρουσιάζεται στο σχήμα 7.3.
- Το αρχείο *.patch*, το οποίο περιέχει πληροφορίες για τα όρια του πλέγματος, όπως είναι το όνομα και ο τύπος του ορίου και το σύνολο των κόμβων που ανήκουν σε αυτό το όριο. Παράδειγμα ενός αρχείου *.patch* παρουσιάζεται στο σχήμα 7.4.

```

nbNodes
nodeX1 nodeX2 nodeX3 ...
nodeY1 nodeY2 nodeY3 ...
nodeZ1 nodeZ2 nodeZ3 ...
  
```

Σχήμα 7.2: Παράδειγμα της δομής του αρχείου *.nod* όπου  $nodeX^i$ ,  $nodeY^i$ ,  $nodeZ^i$  είναι οι  $x$ ,  $y$  και  $z$  συντεταγμένες, αντίστοιχα, του  $i$ -στού κόμβου και  $nbNodes$  είναι ο συνολικός αριθμός κόμβων.

```

nbTet  nbPyr  nbPri  nbHex
iTet11 iTet12 iTet13 iTet14 ...
iPyr11 iPyr12 iPyr13 iPyr14 iPyr15 ...
iPri11 iPri12 iPri13 iPri14 iPri15 iPri16 ...
iHex11 iHex12 iHex13 iHex14 iHex15 iHex16 iHex17 iHex18 ...

```

**Σχήμα 7.3:** Παράδειγμα της δομής του αρχείου *.hyb* όπου *nbTet*, *nbPyr*, *nbPri*, *nbHex* είναι ο αριθμός τετραέδρων, πυραμίδων, πρισμάτων και εξαέδρων αντίστοιχα και *iTet<sub>i</sub><sup>j</sup>*, *iPyr<sub>i</sub><sup>j</sup>*, *iPri<sub>i</sub><sup>j</sup>*, *iHex<sub>i</sub><sup>j</sup>* είναι ο αύξων αριθμός του *j*-οστού κόμβου του *i*-οστού στοιχείου κάθε είδους (τετράεδρο, πυραμίδα, πρίσμα και εξαέδρο αντίστοιχα).

```

nbPatches
patchName_1
{
    BCType boundaryConditionType | geometricType
    ConnectionId 1
    ⋮
    Nodes nbNodes
    list_of_patch_node_ids
}
patchName_2
{
    BCType boundaryConditionType | geometricType
    ConnectionId 1
    ⋮
    Nodes nbNodes
    list_of_patch_node_ids
}
...

```

**Σχήμα 7.4:** Παράδειγμα της δομής του αρχείου *.patch* όπου *nbPatches* είναι ο συνολικός αριθμός ορίων που υπάρχουν και *patchName<sub>i</sub>* είναι το όνομα του *i*-οστού ορίου.

Οι λέξεις-κλειδιά *boundaryConditionType* και *geometricType* μπορούν να λάβουν διάφορες τιμές, οι οποίες παρέχουν πληροφορία για το είδος του ορίου και της οριακής συνθήκης που επιβάλλεται. Οι διάφορες επιτρεπόμενες τιμές παρουσιάζονται στον πίνακα 7.1:

**Πίνακας 7.1:** Επιτρεπόμενες τιμές για τις λέξεις-κλειδιά *boundaryConditionType* και *geometricType*.

<i>boundaryConditionType</i>	<i>geometricType</i>		
<i>InviscidStationaryWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>ViscousStationaryWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>InviscidRotatingWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>ViscousRotatingWall</i>	<i>Blade</i>	<i>Hub</i>	<i>Shroud</i>
<i>MatchingConnection</i>	<i>PeriodicSideX</i>	<i>PeriodicSideY</i>	
<i>Inlet</i>	—		
<i>Outlet</i>	—		
<i>InletRSI</i>	—		
<i>OutletRSI</i>	—		

Τέλος, σε περίπτωση που περιλαμβάνει την αλληλεπίδραση ρότορα-στάτορα (RSI), η λέξη-κλειδί *ConnectionId* προσδιορίζει την αντίστοιχη "έξοδο" του upstream στοιχείου και "είσοδο" του downstream στοιχείου της RSI διεπιφάνειας.

## Ο Τύπος Tecplot

Κάθε αρχείο *.plt* ξεκινάει με μια "επικεφαλίδα" η οποία περιέχει γενικές πληροφορίες για το πλέγμα και τις μεταβλητές ροής, όπως ο συνολικός αριθμός κόμβων και στοιχείων, το όνομα και ο τύπος των μεταβλητών που εμπεριέχονται στο αρχείο. Στο σχήμα 7.5, παρουσιάζεται ένα παράδειγμα μιας τέτοιας "επικεφαλίδας".

```

title = filename
variables = var_1, var_2, var_3 ...
zone t = zone_name
Nodes = nbNodes
Elements = nbElements
ZoneType = {FeBrick, FePoint}
DataPacking = {Block, Brick}
DT = (var_1_DataType, var_2_DataType, var_3_DataType ...)
Results

```

**Σχήμα 7.5:** Παράδειγμα "επικεφαλίδας" ενός αρχείου *.plt*.

Μετά την "επικεφαλίδα" ακολουθούν οι τιμές των ροϊκών μεγεθών γραμμένα σε μια από τις δυο μεθόδους γραφής, την *point – wise* και την *block – wise*. Στην *point – wise* μέθοδο, σε κάθε γραμμή του αρχείου γράφονται οι τιμές όλων των μεταβλητών που αφορούν ένα κόμβο, ενώ στην *block – wise* μέθοδο γράφονται οι τιμές της κάθε

μεταβλητής για όλους τους κόμβους σε ξεχωριστά τμήματα (blocks). Παραδείγματα των δυο μεθόδων φαίνονται στα σχήματα 7.6 και 7.7 αντίστοιχα.

$$\begin{aligned} var\_1\_Node^1, var\_2\_Node^1, var\_3\_Node^1 & \dots \\ var\_1\_Node^2, var\_2\_Node^2, var\_3\_Node^2 & \dots \\ var\_1\_Node^3, var\_2\_Node^3, var\_3\_Node^3 & \dots \end{aligned}$$

**Σχήμα 7.6:** Παράδειγμα της μεθόδου γραφής *point – wise*, όπου  $var\_i\_Node^j$  είναι η τιμή της  $i$ -στής μεταβλητής του  $j$ -οστού κόμβου.

$$\begin{aligned} var\_1\_Node^1, var\_1\_Node^2, var\_1\_Node^3 & \dots \\ var\_2\_Node^1, var\_2\_Node^2, var\_2\_Node^3 & \dots \\ var\_3\_Node^1, var\_3\_Node^2, var\_3\_Node^3 & \dots \end{aligned}$$

**Σχήμα 7.7:** Παράδειγμα της μεθόδου γραφής *block – wise*, όπου  $var\_i\_Node^j$  είναι η τιμή της  $i$ -στής μεταβλητής του  $j$ -οστού κόμβου.

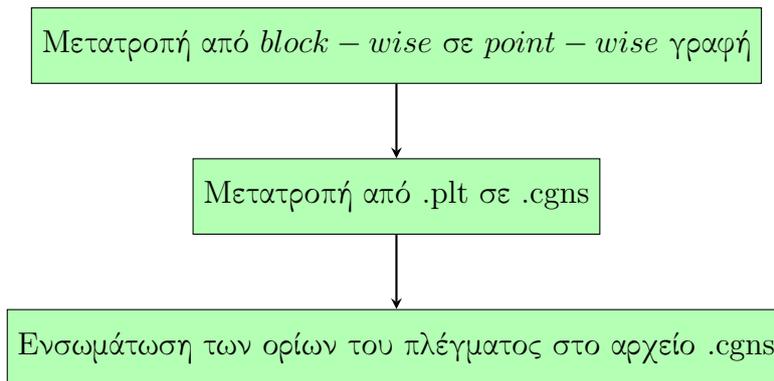
## Ο Μετατροπέας `cgnsToLtt`

Το πρώτο εργαλείο μετατροπής πλεγμάτων που προγραμματίστηκε στα πλαίσια της διπλωματικής εργασίας, είναι το `cgnsToLtt` που μετατρέπει το αρχείο εισόδου `.cgns` στην δομή των τριών αρχείων που συντιστούν τον τύπο LTT. Η μετατροπή επιτυγχάνεται μέσω της χρήσης της βιβλιοθήκης `Cgns Mid-Level Library`, η οποία παρέχει πρόσβαση στις πληροφορίες που υπάρχουν στο αρχείο `.cgns`.

Πιο συγκεκριμένα, η σειρά των εργασιών που εκτελούνται είναι η εξής: (α) διαβάζεται η λίστα των κόμβων και γράφεται το αρχείο `.nod`, (β) διαβάζεται η λίστα των στοιχείων του πλέγματος και δημιουργείται το αρχείο `.hyb` και (γ) διαβάζονται οι απαραίτητες πληροφορίες για τα όρια του πλέγματος και γράφεται το αρχείο `.patch`

## Ο Μετατροπέας `pltToCgns`

Ο δεύτερος μετατροπέας που προγραμματίστηκε είναι ο `pltToCgns`, ο οποίος χρησιμοποιείται για την συμβατότητα ανάμεσα στον επιλύτη PUMA και τον μετεπεξεργαστή, διότι ο PUMA παράγει ένα αρχείο `.plt` και ο μετεπεξεργαστής χρειάζεται ένα αρχείο `.cgns`. Η μετατροπή αυτή γίνεται σε δυο βήματα: (α) μετατρέπεται η *block – wise* γραφή του αρχείου `.plt` σε *point – wise* και (β) γίνεται η μετατροπή του αρχείου `.plt` σε `.cgns`. Η συνολική διαδικασία παρουσιάζεται γραφικά στο σχήμα 7.8



**Σχήμα 7.8:** Γραφική αναπαράσταση της ροής εργασιών που εκτελούνται στο μετατροπέα *pltToCgns*.

### 7.3.2 Ο Προεπεξεργαστής για τον Επιλύτη PUMA

Ο προεπεξεργαστής που προγραμματίστηκε, έχει σκοπό προετοιμασία του τρεξίματος του επιλύτη PUMA. Εκτελεί μια σειρά από υπολογισμούς που μετατρέπουν τις μεταβλητές του υδροστροβίλου, όπως είναι το υδραυλικό ύψος  $H$  και η παροχή  $Q$  σε μεταβλητές που χρειάζονται για τον επιλύτη PUMA, όπως είναι η ολική πίεση  $p_{tot}$  και το μέτρο της ταχύτητας  $V$ . Συγκεκριμένα οι υπολογισμοί που εκτελούνται είναι:

- Η μετατροπή της γωνιακής ταχύτητας από  $rpm$  σε  $\frac{rad}{s}$ :

$$\omega = \frac{2\pi n}{60} \quad (7.1)$$

όπου  $\omega$  είναι η γωνιακή ταχύτητα σε  $\frac{rad}{s}$  και  $n$  είναι η γωνιακή ταχύτητα σε  $rpm$ .

- Ο υπολογισμός του μέτρου της ταχύτητας στην είσοδο  $V_{in}$  του στροβίλου με βάση την παροχή  $Q$  που περνάει από ένα κανάλι του δρομέα:

$$V_{in} = \frac{Q}{A_{in} \sin(a)} \quad (7.2)$$

όπου  $A_{in}$  είναι η επιφάνεια της εισόδου ενός καναλιού (σε  $m^2$ ) και  $a$  είναι η γωνία εισόδου της ροής σε μοίρες ( $^\circ$ ).

- Ο υπολογισμός της στατικής πίεσης στην έξοδο του στροβίλου  $p_{s,out}$ :

$$p_{s,out} = p_{atm} + \rho g(z_{ref} - z_{out} - Z_s) - \Delta p_{s,out} \quad (7.3)$$

όπου  $p_{atm}$  είναι η ατμοσφαιρική πίεση (σε  $Pa$ ),  $z_{ref}$  είναι το επίπεδο αναφοράς (σε  $m$ ),  $z_{out}$  είναι το επίπεδο της εξόδου του στροβίλου (σε  $m$ ),  $Z_s$  είναι το ύψος αναρρόφησης (σε  $m$ ) και  $\Delta p_{s,out}$  είναι η εκτιμώμενη πτώση πίεσης στον αγωγό εξόδου.

Επίσης, ανάλογα τον τύπο της οριακής συνθήκης που επιβάλλεται στην είσοδο υπολογίζονται:

- Η γωνία εισόδου της ροής (αν επιβάλλεται η ταχύτητα στην είσοδο):

$$\alpha = \arctan\left(\frac{\omega R Q}{A_{in} g H_{net}}\right) \quad (7.4)$$

όπου  $H_{net}$  είναι το υδραυλικό ύψος (σε  $m$ ) που καλείται να διαχειριστεί ο στρόβιλος.

- Ο υπολογισμός της ολικής πίεσης στην είσοδο  $p_{tot,in}$  (αν επιβάλλεται η ολική πίεση στην είσοδο):

$$p_{tot,in} = \rho g H_{net} + p_{s,out} + p_{dyn,out} \quad (7.5)$$

όπου  $p_{dyn,out}$  είναι η δυναμική πίεση στην έξοδο του στροβίλου και υπολογίζεται ως εξής:  $p_{dyn,out} = \frac{1}{2} \rho \overline{V_{out}}^2$ , με  $\overline{V_{out}}$  να είναι το μέτρο της μέσης ταχύτητας στην έξοδο του στροβίλου.

### 7.3.3 Το εργαλείο set-up-run-gpu

Το εργαλείο set-up-run-gpu προγραμματίστηκε για να έχει συντονιστικό ρόλο. Οργανώνει και εναρμονίζει την εκτέλεση όλων των υπολοίπων εργαλείων, ώστε να επιτυγχάνεται η εύρυθμη και ομαλή λειτουργία της συνολικής αλυσίδας εργαλείων.

### 7.3.4 Μετεπεξεργαστικές υπορουτίνες

Επιθυμώντας την χρήση του προϋπάρχοντος μετεπεξεργαστή, ορισμένες από τις χρησιμοποιούμενες υπορουτίνες πρέπει να τροποποιηθούν για να μπορούν να υποστηρίξουν το προκύπτον αρχείο από τον επιλύτη PUMA.

Οι υπορουτίνες που τροποποιήθηκαν περιλαμβάνουν:

- Τον υπολογισμό της παροχής μάζας:

$$\dot{m} = \iint_A \rho V_n dA \quad (7.6)$$

όπου  $V_n$  είναι το διάνυσμα της ταχύτητας που είναι κάθετο στην επιφάνεια όπου υπολογίζεται η παροχή μάζας.

- Υπολογισμός του μέσου όρου κατά μάζα της μεταβλητής  $\Phi$ :

$$\bar{\Phi}_{mass} = \frac{\iint_A \rho V_n \Phi dA}{\iint_A \rho V_n dA} \quad (7.7)$$

- Ολοκλήρωση κατά μάζα της μεταβλητής  $\Phi$ :

$$\iint_A \rho V_n \Phi dA \quad (7.8)$$

- Υπολογισμός της ροπής

$$M_{tot} = M_{p,z} + M_{v,z} \quad (7.9)$$

όπου  $M_{p,z} = \iint_{A_{bl,x}} xp_{abs} dA - \iint_{A_{bl,y}} yp_{abs} dA$  είναι ο όρος της ροπής που προκύπτει από την πίεση που ασκείται στην επιφάνεια και  $M_{v,z} = \iint_{A_{bl}} x\tau_{w,y} dA - \iint_{A_{bl}} y\tau_{w,x} dA$  είναι ο όρος της ροπής που προκύπτει από τις διατμητικές τάσεις που αναπτύσσονται στην επιφάνεια.

Συγκεκριμένα οι διατμητικές τάσεις υπολογίζονται ως εξής:

$$\begin{aligned} - \tau_{w,x} &= (\mu + \mu_t) \left( 2n_x \frac{\partial u}{\partial x} + n_y \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + n_z \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right). \\ - \tau_{w,y} &= (\mu + \mu_t) \left( n_x \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + 2n_y \frac{\partial v}{\partial y} + n_z \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right). \end{aligned}$$

## 7.4 Βιομηχανικές Εφαρμογές και Σύγκριση

Για την πιστοποίηση του επιλύτη PUMA μελετώνται τρεις περιπτώσεις υδροστροβίλων, κλιμακούμενης πολυπλοκότητας ως προς την διάταξη, και γίνεται η σύγκριση των αποτελεσμάτων του PUMA με τα αποτελέσματα που προκύπτουν από έναν εμπορικό επιλύτη ΥΡΔ, που παρέχεται από την Andritz Hydro.

Οι διατάξεις των τριών υδροστροβίλων που παρουσιάζονται είναι οι εξής: (α) ο δρομέας ενός υδροστροβίλου Francis, (β) τα οδηγία πτερύγια και ο δρομέας ενός υδροστροβίλου τύπου προπέλα και (γ) τα οδηγία πτερύγια, ο δρομέας και ο αγωγός εξόδου ενός υδροστροβίλου Kaplan. Οι τρεις στροβίλοι παρουσιάζονται στο σχήμα 7.9.

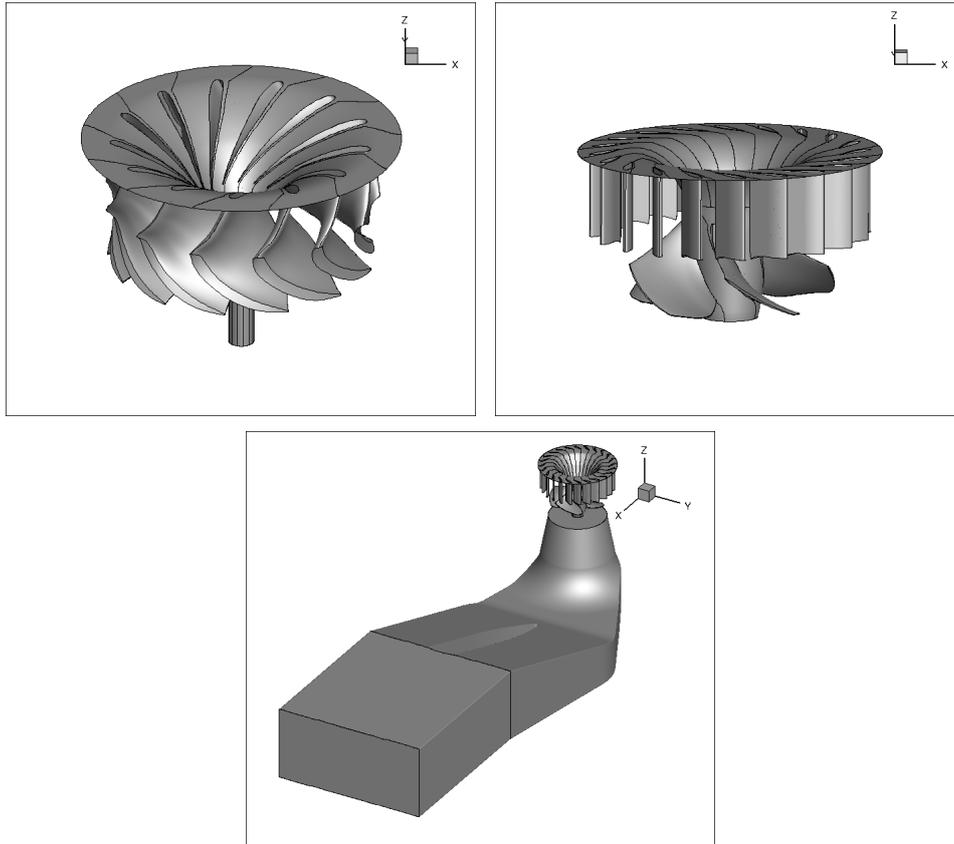
Για την αλληλεπίδραση σταθερού και περιστρεφόμενου τμήματος του υδροστροβίλου χρησιμοποιείται η μέθοδος *MixingPlane*. Τέτοιες περιπτώσεις συναντώνται στην διεπιφάνεια των οδηγίων πτερυγίων-δρομέα (σχήμα 7.10) και δρομέα-αγωγού εξόδου (σχήμα 7.11).

Στην σύγκριση των αποτελεσμάτων, κυρίως, παρουσιάζονται:

- Οι κατανομές του συντελεστή πίεσης  $C_p$  σε τρεις ακτινικές θέσεις του πτερυγίου, τη ρίζα (hub) το μέσο (mid-span) και το άκρο (shroud). Ο συντελεστής πίεσης  $C_p$  υπολογίζεται από τον τύπο:

$$C_p = \frac{p - p_{vap}}{\rho g H_{net}} \quad (7.10)$$

όπου  $p_{vap}$  είναι η πίεση στην οποία το νερό μετατρέπεται σε ατμό.



**Σχήμα 7.9:** Οι τρεις υδροστροβίλοι που μελετήθηκαν. Ο δρομέας Francis (πάνω-αριστερά), τα οδηγιά πτερύγια και ο δρομέας του υδροστροβίλου τύπου προπέλα (πάνω-δεξιά) και τα οδηγιά πτερύγια, ο δρομέας και ο αγωγός εξόδου του Kaplan (κάτω).

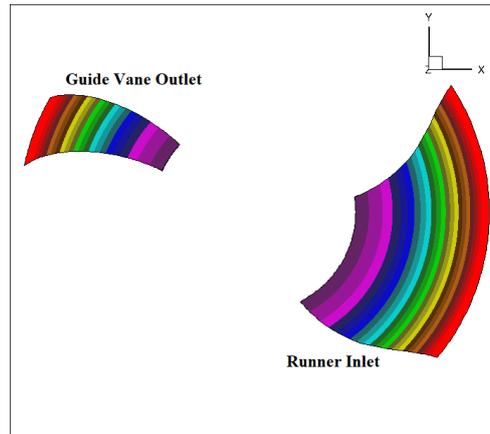
- Οι κατανομές των αδιάστατων κυλινδρικών απόλυτων ταχυτήτων (περιφερειακή  $c_u^*$ , ακτινική  $c_r^*$  και μεσημβρινή  $c_m^*$ ), που υπολογίζονται με βάση τους τύπους:

$$\begin{aligned}
 c_u^* &= \frac{c_u}{\sqrt{2gH_{net}}} \\
 c_r^* &= \frac{c_r}{\sqrt{2gH_{net}}} \\
 c_m^* &= \frac{c_m}{\sqrt{2gH_{net}}}
 \end{aligned}
 \tag{7.11}$$

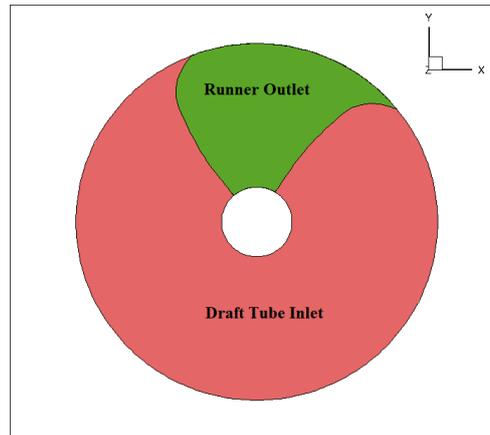
- Τα υπολογιζόμενα μεγέθη που περιγράφουν την λειτουργία του υδροστροβίλου (α) υδραυλικό ύψος  $H_{calc} = \frac{P_{t,in} - P_{t,out}}{\rho g}$ , (β) παροχή  $Q$ , (γ) την ροπή  $T$  και τον βαθμό απόδοσης του στροβίλου  $\eta = \frac{T\omega}{\rho g H_{calc} Q}$ .

Στους πίνακες 7.2, 7.3, 7.4 παρουσιάζονται τα σημεία λειτουργίας για τους τρεις υδροστροβίλους που μελετώνται.

Στα σχήματα 7.12, 7.13, 7.14, παρουσιάζεται ο ρυθμός σύγκλισης των πέντε ΜΔΕ που



**Σχήμα 7.10:** ΥΡΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Οι ακτινικές ζώνες που δημιουργούνται στις δύο πλευρές της διεπιφάνειας μεταξύ οδηγών πτερυγίων και δρομέα για τον υπολογισμό των μέσων τιμών κατά μάζα των ροϊκών μεγεθών.



**Σχήμα 7.11:** ΥΡΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Karlan χρησιμοποιώντας τον PUMA. Οι δυο πλευρές της διεπιφάνειας δρομέας (πράσινο)-αγωγός εξόδου (κόκκινο). Για τον δρομέα χρησιμοποιείται ένα κανάλι για την ανάλυση, ενώ ο αγωγός εξόδου χρησιμοποιείται ολόκληρος.

**Πίνακας 7.2:** ΥΡΔ ανάλυση του δρομέα Francis χρησιμοποιώντας τον PUMA. Σημείο λειτουργίας του στροβίλου.

Μέγεθος	Τιμή
H (m)	34.59
X ( $\frac{m^3}{s}$ )	0.57
N (rpm)	1021.04

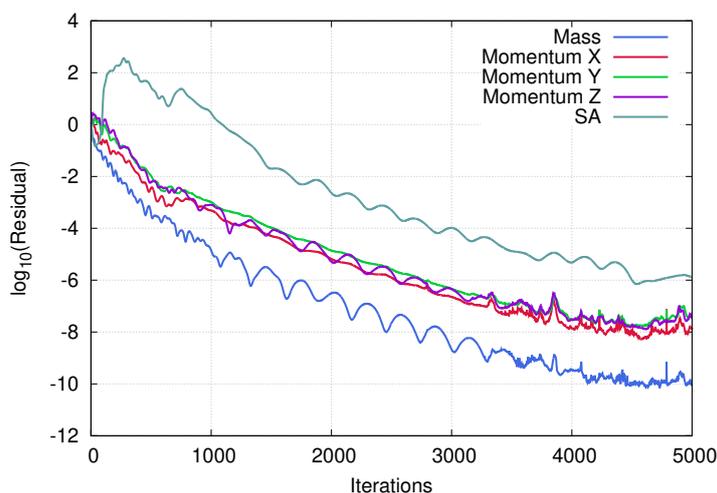
επιλύονται (εξ. συνέχειας, τρεις εξ. ορμής και η εξ. S-A) για τις τρεις διατάξεις και στα σχήματα 7.15, 7.16, 7.17 η εξέλιξη των ολοκληρωματικών μεγεθών της παροχής, της ολικής και της στατικής πίεσης στην είσοδο και την έξοδο του εκάστοτε υδροστροβίλου.

**Πίνακας 7.3:** ΥΡΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Σημείο λειτουργίας του στροβίλου.

Μέγεθος	Τιμή
H (m)	13.22
X ( $\frac{m^3}{s}$ )	0.51
N (ρπμ)	1651.87

**Πίνακας 7.4:** ΥΡΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Karlan χρησιμοποιώντας τον PUMA. Σημείο λειτουργίας του στροβίλου.

Μέγεθος	Τιμή
H (m)	6.43
X ( $\frac{m^3}{s}$ )	0.42
N (ρπμ)	1160.67

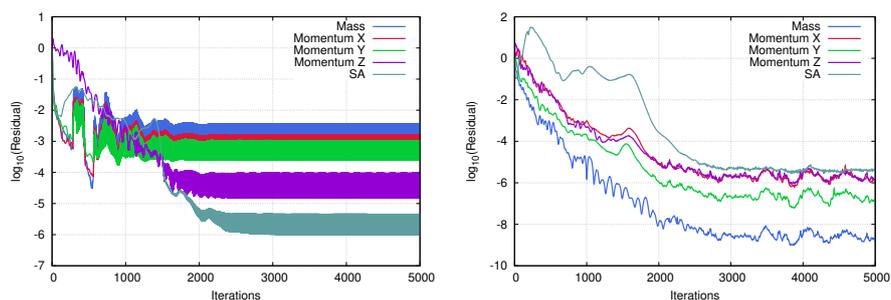


**Σχήμα 7.12:** ΥΡΔ ανάλυση του δρομέα Francis χρησιμοποιώντας τον PUMA. Σύγκλιση των πέντε εξισώσεων (συνέχειας, ορμής κατά x, ορμής κατά y, ορμής κατά z και μοντέλο τύρβης S-A).

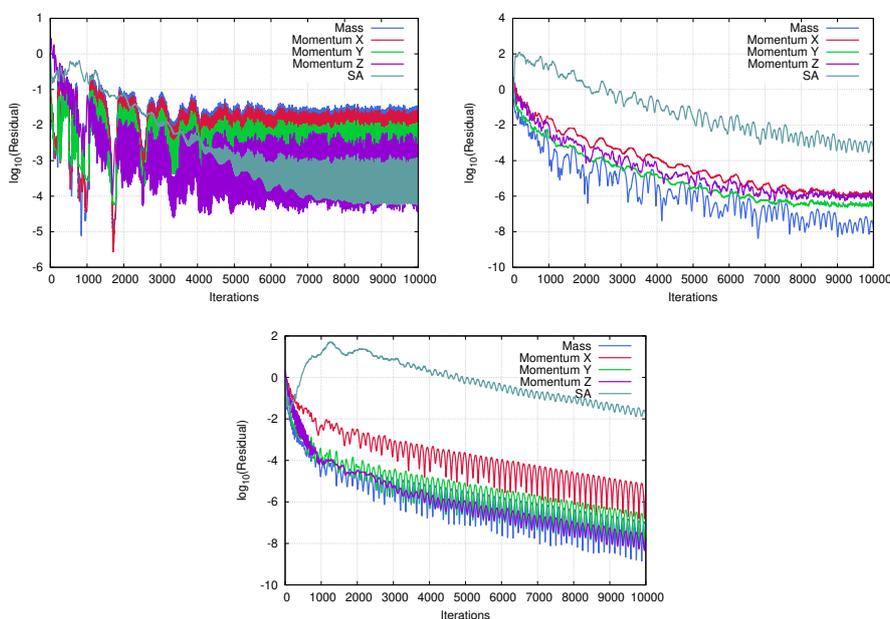
Στα σχήματα 7.18, 7.19, 7.20, παρουσιάζονται οι κατανομές του  $C_p$  στις τρεις ακτινικές θέσεις του πτερυγίου, ενώ στο σχήμα 7.21, 7.22, 7.23, παρουσιάζονται οι κατανομές των τριών απόλυτων κυλινδρικών ταχυτήτων στην έξοδο του δρομέα του κάθε υδροστροβίλου.

Τέλος, στους πίνακες 7.5, 7.6, 7.7, παρουσιάζονται οι τιμές των υπολογιζόμενων μεγεθών που περιγράφουν τη λειτουργία του εκάστοτε στροβίλου.

Είναι χρήσιμο να αναφερθεί ότι, εν γένει, η πρόβλεψη της ροής μέσα στον αγωγό εξόδου είναι ένα εγχείρημα το οποίο είναι απαιτητικό για κάθε επιλύτη ροής. Ακόμα και τα αποτελέσματα του εμπορικού επιλύτη δεν είναι ικανοποιητικά συγκρινόμενα με



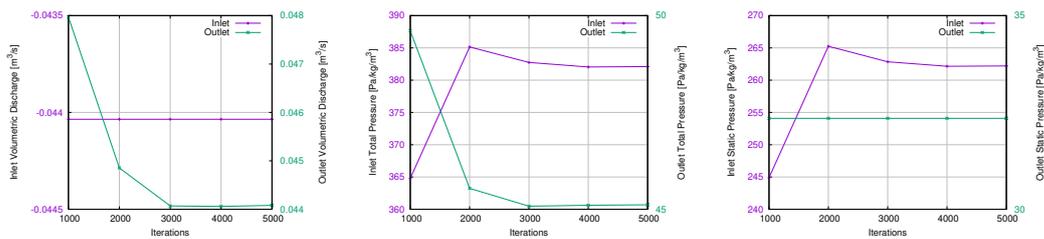
**Σχήμα 7.13:** ΥΡΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Σύγκλιση των πέντε εξισώσεων (συνέχειας, ορμής κατά  $x$ , ορμής κατά  $y$ , ορμής κατά  $z$  και μοντέλο τύρβης  $S-A$ ) για τα οδηγά πτερύγια (αριστερά) και το δρομέα (δεξιά).



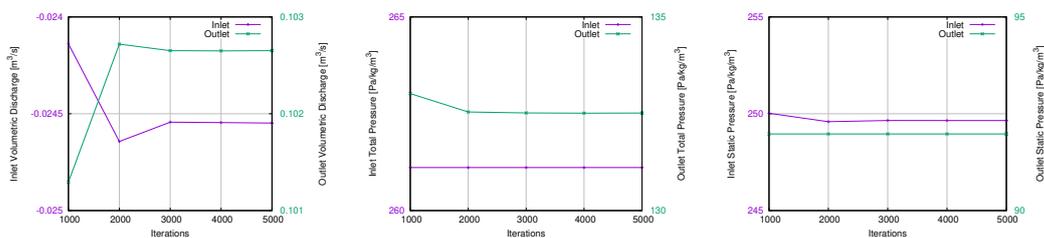
**Σχήμα 7.14:** ΥΡΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Karlan χρησιμοποιώντας τον PUMA. Σύγκλιση των πέντε εξισώσεων (συνέχειας, ορμής κατά  $x$ , ορμής κατά  $y$ , ορμής κατά  $z$  και μοντέλο τύρβης  $S-A$ ) για τα οδηγά πτερύγια (πάνω-αριστερά) το δρομέα (πάνω-δεξιά) και τον αγωγό εξόδου (κάτω).

τις αντίστοιχες μετρήσεις που πραγματοποιούνται στο εργαστήριο. Για να μελετηθεί, περαιτέρω, η επίδραση της παρουσίας του αγωγού εξόδου δημιουργούνται οι ισοϋψείς καμπύλες των τριών αδιάστατων απόλυτων κυλινδρικών ταχυτήτων πάνω σε τρεις επιφάνειες σταθερού  $z$ , μια κοντά στην ακμή εκφυγής του πτερυγίου του δρομέα, μια στη διεπιφάνεια του δρομέα και του αγωγού εξόδου και μια σε μια ενδιάμεση θέση. Οι τρεις επιφάνειες παρουσιάζονται στο σχήμα 7.24, ενώ οι ισοϋψείς καμπύλες των ταχυτήτων παρουσιάζονται στο σχήμα 7.25.

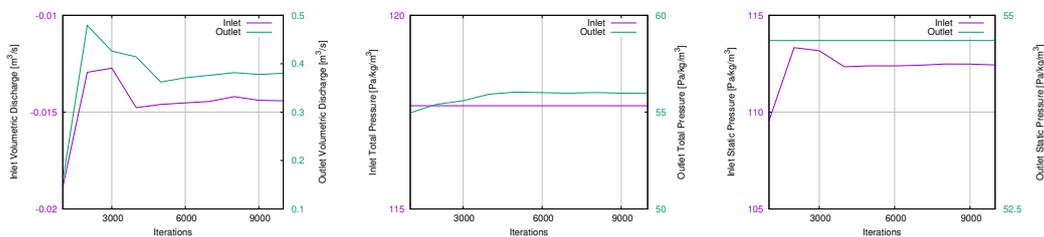
Όπως φαίνεται από το σχήμα 7.25 οι αποκλίσεις στα αποτελέσματα των δυο επιλυτών



**Σχήμα 7.15:** ΥΡΔ ανάλυση του δρομέα Francis χρησιμοποιώντας τον PUMA. Εξέλιξη των ολοκληρωματικών μεγεθών της παροχής (αριστερά), της ολικής (μέση) και της στατικής (δεξιά) πίεσης στην είσοδο και την έξοδο του δρομέα.



**Σχήμα 7.16:** ΥΡΔ ανάλυση των οδηγών περυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Εξέλιξη των ολοκληρωματικών μεγεθών της παροχής (αριστερά), της ολικής (μέση) και της στατικής (δεξιά) πίεσης στην είσοδο των οδηγών περυγίων και την έξοδο του δρομέα.

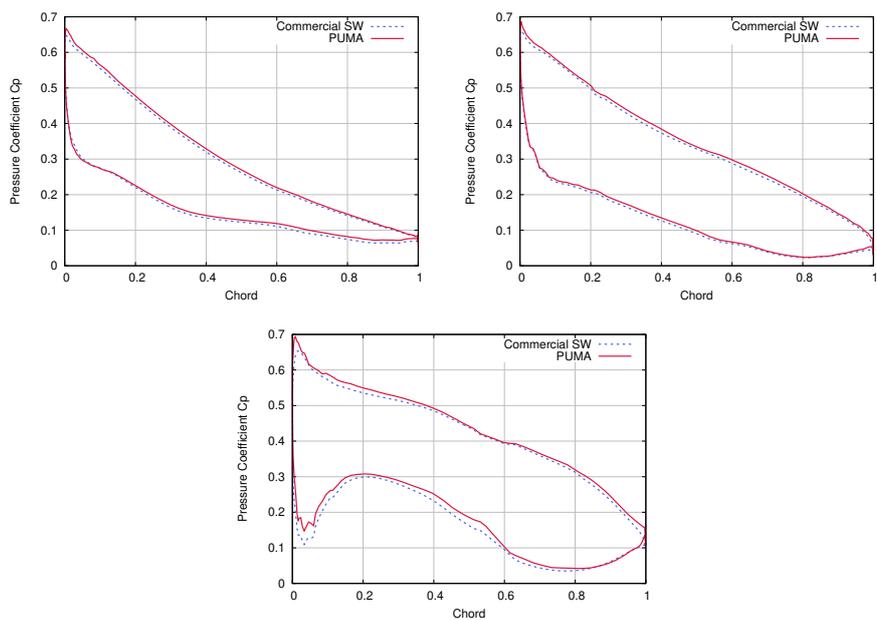


**Σχήμα 7.17:** ΥΡΔ ανάλυση των οδηγών περυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Karlan χρησιμοποιώντας τον PUMA. Εξέλιξη των ολοκληρωματικών μεγεθών της παροχής (αριστερά), της ολικής (μέση) και της στατικής (δεξιά) πίεσης στην είσοδο και την έξοδο του δρομέα.

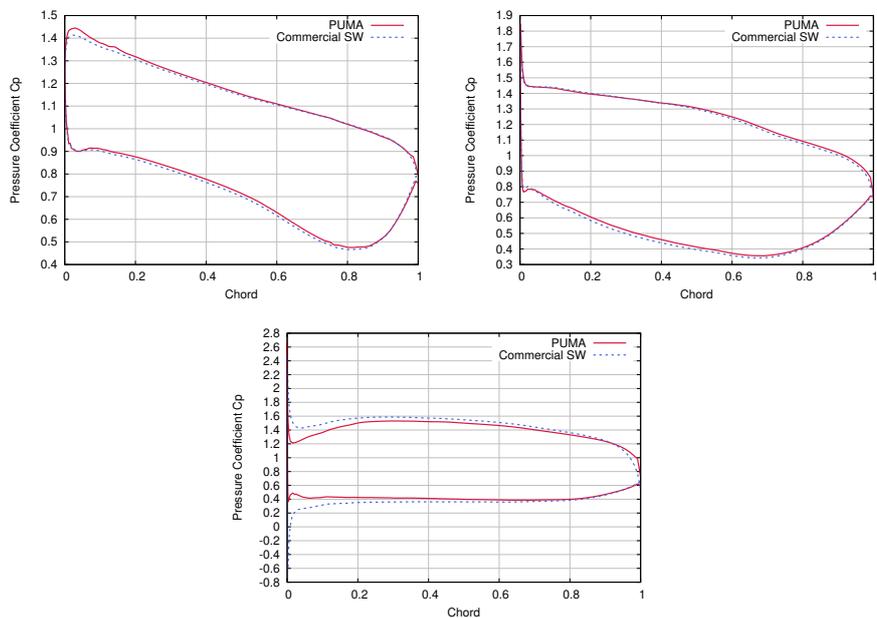
μεγαθύνονται όσο πλησιάζουμε στην διεπιφάνεια δρομέα-αγωγού εξόδου, υποδηλώνοντας την επίδραση που έχει ο αγωγός εξόδου.

#### 7.4.1 Συμπεράσματα από τη Σύγκριση

Ο ρυθμός σύγκλισης των ΜΔΕ που επιλύονται και η εξέλιξη των ολοκληρωματικών μεγεθών υποδηλώνουν ότι η σύγκλιση έχει επιτευχθεί αρκετά πριν από τις 5000 επαναλήψεις. Έτσι, το υπολογιστικό κόστος μπορεί να μειωθεί ακόμη περισσότερο.

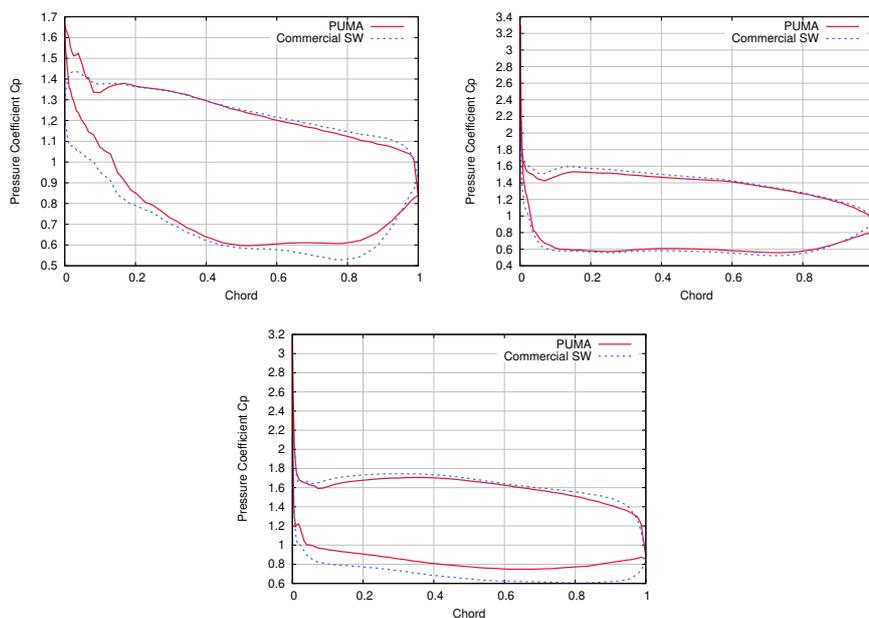


Σχήμα 7.18: ΥΡΔ ανάλυση του δρομέα Francis χρησιμοποιώντας τον PUMA. Κατανομές του συντελεστή πίεσης  $C_p$  στη ρίζα (πάνω-αριστερά), το μέσο (πάνω-δεξιά) και το άκρο (κάτω) του πτερυγίου.

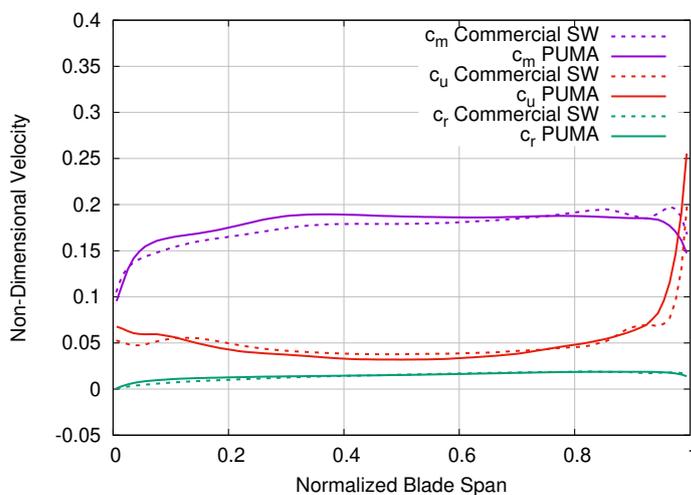


Σχήμα 7.19: ΥΡΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Κατανομές του συντελεστή πίεσης  $C_p$  στη ρίζα (πάνω-αριστερά), το μέσο (πάνω-δεξιά) και το άκρο (κάτω) του πτερυγίου.

Εξαιρέση αποτελεί η περίπτωση του υδροστροβίλου Kaplan με τον αγωγό εξόδου, όπου χρειάζονται περίπου 8000 επαναλήψεις.

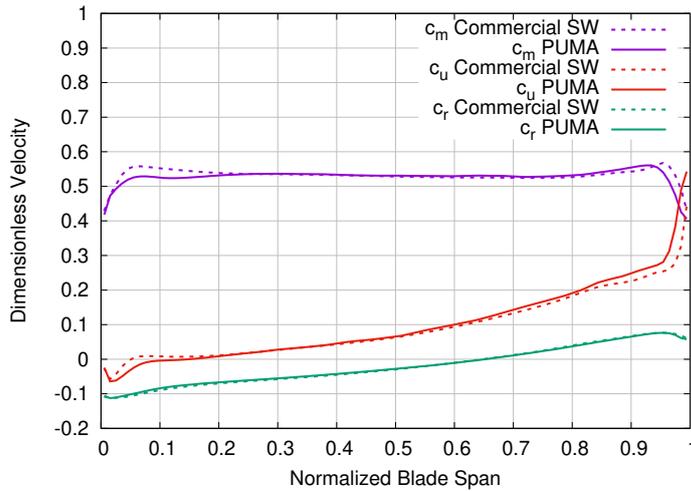


**Σχήμα 7.20:** ΥΠΔ ανάλυση των οδηγών περυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου *Karlan* χρησιμοποιώντας τον PUMA. Κατανομές του συντελεστή πίεσης  $C_p$  στη ρίζα (πάνω-αριστερά), το μέσο (πάνω-δεξιά) και το άκρο (κάτω) του περυγίου.

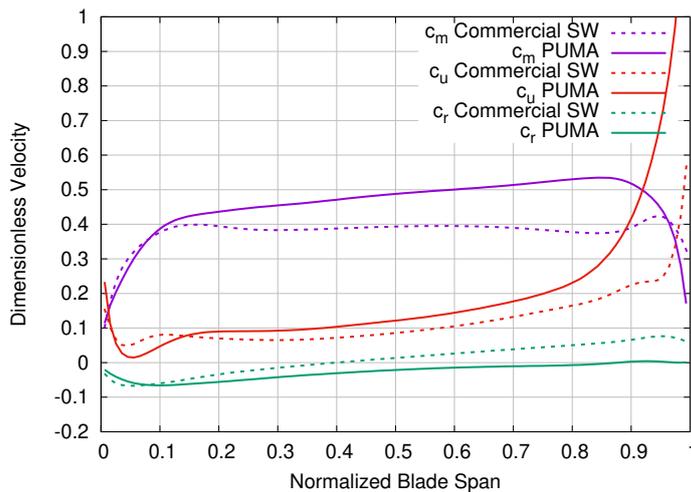


**Σχήμα 7.21:** ΥΠΔ ανάλυση του δρομέα *Francis* χρησιμοποιώντας τον PUMA. Κατανομές των αδιάστατων απόλυτων κυλινδρικών ταχυτήτων στην έξοδο του δρομέα.

Με βάση τα σχήματα 7.18, 7.21, 7.19, 7.22, 7.20, 7.23 οι αποκλίσεις που παρατηρούνται μεταξύ των κατανομών που προκύπτουν από τους δυο επιλύτες είναι αμελητέες. Το ίδιο ισχύει και για τη σύγκριση των συνολικών μεγεθών που παρατίθενται στους πίνακες 7.5, 7.6 και 7.7. Εκεί όπου παρατηρείται η μεγαλύτερη απόκλιση είναι στις τιμές της ροπής και κατ' επέκταση και στις τιμές του βαθμού απόδοσης. Όσον αφορά τις τιμές της υπολογιζόμενης ροπής, οι αποκλίσεις μπορούν να αποδοθούν στις διαφορές των



**Σχήμα 7.22:** TPΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Κατανομές των αδιάστατων απόλυτων κυλινδρικών ταχυτήτων στην έξοδο του δρομέα.



**Σχήμα 7.23:** TPΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Kaplan χρησιμοποιώντας τον PUMA. Κατανομές των αδιάστατων απόλυτων κυλινδρικών ταχυτήτων στην έξοδο του δρομέα.

δυσ επιλυτών (διαφορετικό μοντέλο τύρβης και διαφορετικές συναρτήσεις τοίχου).

### 7.4.2 Παραμετρική Μελέτη της Ψευδοσυμπίεστικότητας $\beta$

Στην ενότητα αυτή, πραγματοποιείται μια μελέτη της επίδρασης της ψευδοσυμπίεστικότητας στο ρυθμό σύγκλισης της επίλυσης και στα αποτελέσματα της ανάλυσης της ροής. Η ψευδοσυμπίεστικότητα εκφράζεται μέσω της παραμέτρου  $\beta$ , η οποία αναπαριστά μια τεχνητή ταχύτητα του ήχου. Ο τρόπος ελέγχου της παραμέτρου  $\beta$ , που παρέχεται

**Πίνακας 7.5:** ΥΡΔ ανάλυση του δρομέα Francis χρησιμοποιώντας τον PUMA. Συνολικά μεγέθη που περιγράφουν τη λειτουργία του στροβίλου. Για τον βαθμό απόδοσης  $\eta$ , παρουσιάζεται η εκατοστιαία απόκλιση από το αντίστοιχο αποτέλεσμα του εμπορικού επιλύτη.

Μεταβλητή	ΠΥΜΑ	δμμερσιαλ ΣΩ
H ( $\mu$ )	34.37	33.92
X ( $\frac{m^3}{s}$ )	0.573	0.571
$\delta\eta(\%)$	+2.4	-
T (N $\mu$ )	1792.21	1720.75

**Πίνακας 7.6:** ΥΡΔ ανάλυση των οδηγών πτερυγίων και του δρομέα υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Συνολικά μεγέθη που περιγράφουν τη λειτουργία του στροβίλου. Για τον βαθμό απόδοσης  $\eta$ , παρουσιάζεται η εκατοστιαία απόκλιση από το αντίστοιχο αποτέλεσμα του εμπορικού επιλύτη.

Μεταβλητή	ΠΥΜΑ	δμμερσιαλ ΣΩ
H ( $\mu$ )	13.20	13.18
X ( $\frac{m^3}{s}$ )	0.513	0.517
$\delta\eta(\%)$	+1.4	-
T (N $\mu$ )	375.86	372.76

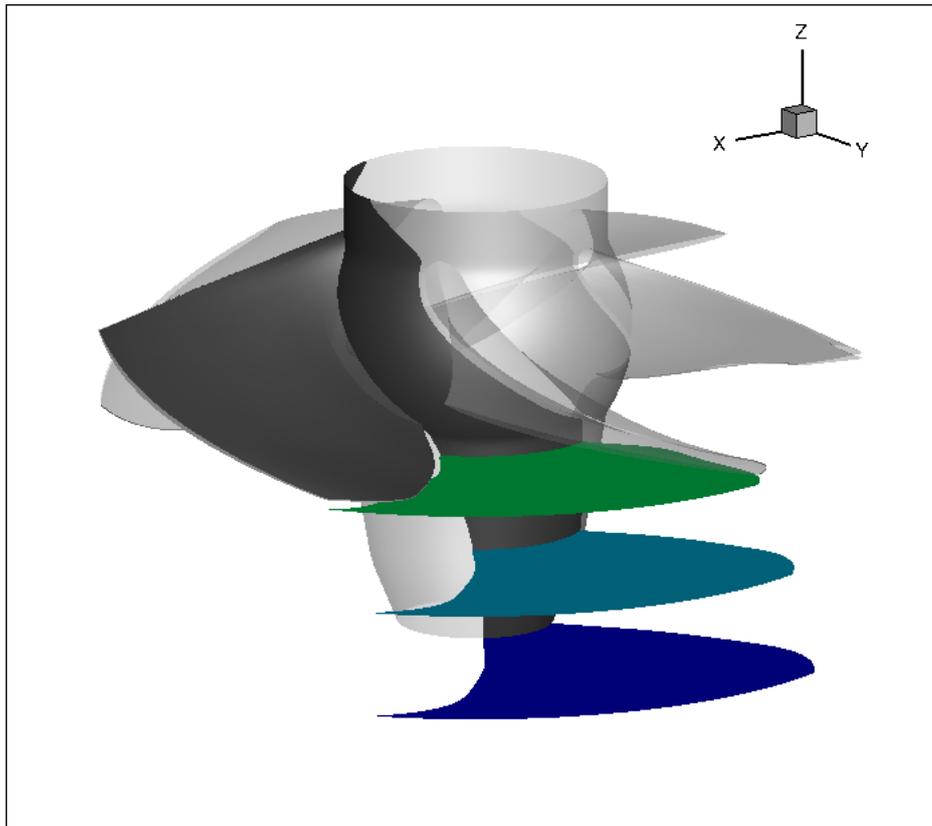
**Πίνακας 7.7:** ΥΡΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Kaplan χρησιμοποιώντας τον PUMA. Συνολικά μεγέθη που περιγράφουν τη λειτουργία του στροβίλου. Για τον βαθμό απόδοσης  $\eta$ , παρουσιάζεται η εκατοστιαία απόκλιση από το αντίστοιχο αποτέλεσμα του εμπορικού επιλύτη.

Μεταβλητή	ΠΥΜΑ	δμμερσιαλ ΣΩ
H ( $\mu$ )	6.11	6.41
X ( $\frac{m^3}{s}$ )	0.379	0.398
$\delta\eta(\%)$	-0.9	-
T (N $\mu$ )	173.67	193.41

από τον επιλύτη PUMA, είναι μέσω της τιμής μιας ταχύτητας αναφοράς  $U_{ref}$  η οποία πολλαπλασιάζεται με την αρχική τιμή του  $\beta$  (3.33).

Η παραμετρική μελέτη πραγματοποιείται για τον υδροστροβίλο τύπου προπέλα που συνδυάζει τη χρήση οδηγών πτερυγίων και δρομέα. Οι τέσσερις τιμές που επιλέγονται για την ταχύτητα αναφοράς ( $U_{ref}$ ) είναι 10, 20, 30 και 40  $\frac{m}{s}$ . Η πρώτη τιμή επιλέγεται έτσι ώστε να είναι κοντά στην τιμή του μέτρου της ταχύτητας εισόδου ( $V_{in} = 4.89\frac{m}{s}$ ) και οι υπόλοιπες είναι πολλαπλάσια αυτής.

Στο σχήμα 7.26 παρουσιάζεται ο ρυθμός σύγκλισης των πέντε εξισώσεων για τα οδηγία πτερύγια και το δρομέα για τις τέσσερις διαφορετικές τιμές της  $U_{ref}$ . Είναι φανερό ότι η σύγκλιση επιταχύνεται σημαντικά στην περίπτωση όπου η ταχύτητα αναφοράς είναι όσο το δυνατόν κοντύτερα στην τιμή του μέτρου της ταχύτητας εισόδου ( $V_{in} = 4.89\frac{m}{s}$ ).



**Σχήμα 7.24:** ΥΠΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Kaplan χρησιμοποιώντας τον PUMA. Θέσεις των τριών επιφανειών σταθερού  $z$ .

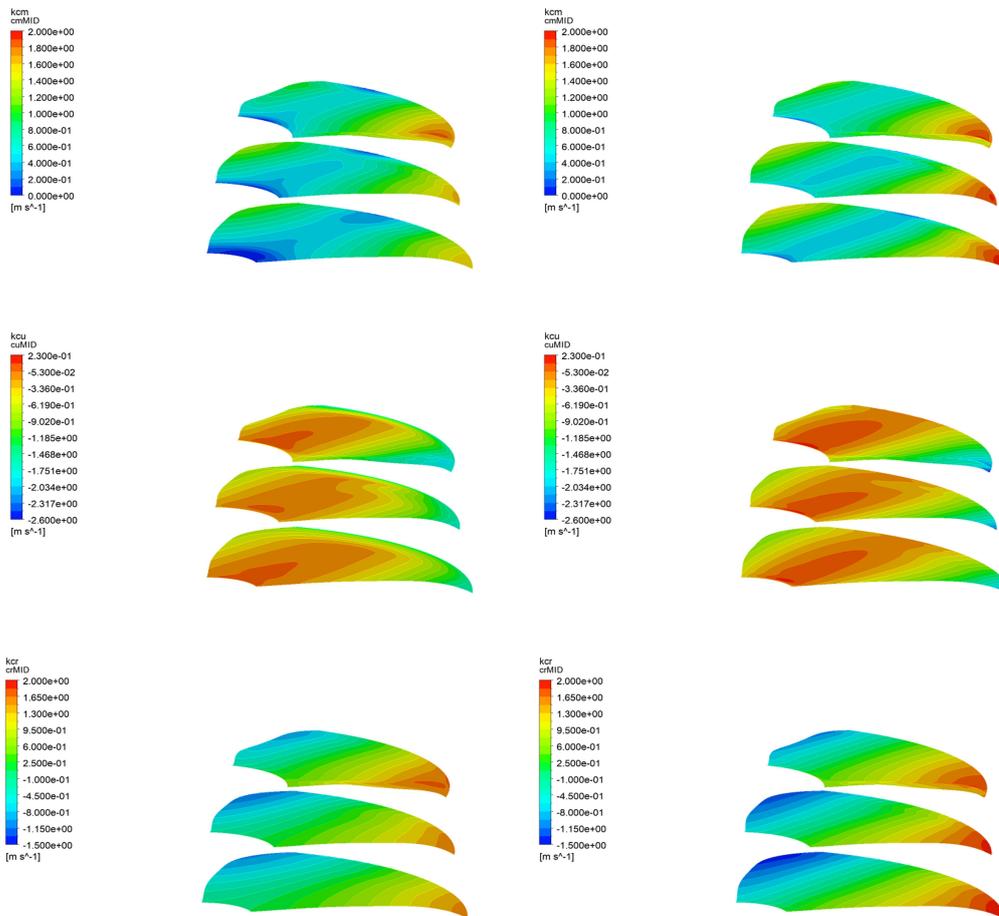
Σημαντικό είναι να αναφερθεί πως για ταχύτητα αναφοράς  $U_{ref} = 5 \frac{m}{s}$  ο επιλύτης απέκλινε.

Επίσης, στο σχήμα 7.27, παρουσιάζονται τα πεδία της πίεσης πάνω στην επιφάνεια του πτερυγίου για τις τέσσερις διαφορετικές τιμές της  $U_{ref}$ .

Από το σχήμα 7.27, μπορεί να εξαχθεί το συμπέρασμα ότι παρουσιάζονται μικρές αλλαγές στα προκύπτοντα πεδία πίεσης, κάτι που σημαίνει ότι τα αποτελέσματα δεν είναι παντελώς ανεξάρτητα από την τιμή της παραμέτρου  $\beta$ . Συνεπώς, η τιμή της πρέπει να επιλέγεται προσεκτικά.

## 7.5 Βελτιστοποίηση Πτερυγίου Δρομέα Υδροστροβίλου Kaplan

Το ενιαίο εργαλείο ανάλυσης που έχει δημιουργηθεί χρησιμοποιείται στη συνέχεια για τη βελτιστοποίηση ενός πτερυγίου δρομέα υδροστροβίλου Kaplan. Η μέθοδος βελ-



**Σχήμα 7.25:** ΥΡΔ ανάλυση των οδηγών πτερυγίων, του δρομέα και του αγωγού εξόδου του υδροστροβίλου Karlan χρησιμοποιώντας τον PUMA. Ισοϋψείς καμπύλες των τριών αδιάστατων απόλυτων κυλινδρικών ταχυτήτων:  $c_u^*$  (πάνω),  $c_r^*$  (μέση) ανδ  $c_m^*$  (κάτω) υπολογιζόμενες από τον επιλύτη PUMA (αριστερά) και τον εμπορικό επιλύτη (δεξιά).

τιστοποίησης που χρησιμοποιείται είναι αυτή των Εξελικτικών Αλγορίθμων (ΕΑ), μια στοχαστική μέθοδος, όπως εφαρμόζεται από το εργαλείο βελτιστοποίησης Evolutionary Algorithm SYstem (EASY), το οποίο αναπτύχθηκε από τη ΜΠΥΡ&Β/ΕΜΠ.

### 7.5.1 Μεταβλητές Σχεδιασμού

Σημαντικός για τη διαδικασία της βελτιστοποίησης με ΕΑ είναι ο καθορισμός των μεταβλητών σχεδιασμού και του εύρους τιμών τους. Αυτές είναι οι μεταβλητές που μεταβάλλονται σε κάθε υποψήφια λύση κάθε γενιάς του ΕΑ και ελέγχουν το σχήμα του πτερυγίου του δρομέα. Στην προκειμένη περίπτωση, οι μεταβλητές σχεδιασμού επιλέγονται να είναι οι 30 τιμές των σημείων ελέγχου των καμπυλών *Bézier* που καθορίζουν τις κατανομές της γωνίας μετάλλου  $\beta$  του πτερυγίου (15 σημεία ελέγχου

κατά μήκος της ακμής πρόσπτωσης και 15 σημεία ελέγχου κατά μήκος της ακμής εκφυγής). Στον πίνακα 7.8 παρουσιάζονται το εύρος τιμών για την κάθε μεταβλητή σχεδιασμού [1], [16].

**Πίνακας 7.8:** Βελτιστοποίηση ενός δρομέα Karlan χρησιμοποιώντας τον EASY και τον PUMA. Ελάχιστες και μέγιστες τιμές για κάθε μεταβλητή σχεδιασμού όπου  $\beta_i$  είναι το  $i$ -οστό σημείο ελέγχου της κατανομής της γωνίας μετάλλου  $\beta$ . Οι ελάχιστες και μέγιστες τιμές κάθε μεταβλητής προκύπτουν από μια εκατοστιαία μεταβολή της τάξης του  $\pm 5\%$  στις αντίστοιχες τιμές της αρχικής γεωμετρίας.

Μεταβλητή	Ελ. Τιμή	Μεγ. Τιμή	Μεταβλητή	Ελ. Τιμή	Μεγ. Τιμή
$\beta_1$	65.6812	72.595	$\beta_{16}$	45.2388	50.0008
$\beta_2$	51.8671	57.3267	$\beta_{17}$	40.5096	44.7738
$\beta_3$	41.1341	45.4641	$\beta_{18}$	30.7535	33.9907
$\beta_4$	28.6676	31.6852	$\beta_{19}$	22.0966	24.4226
$\beta_5$	23.5894	26.0726	$\beta_{20}$	22.5649	24.9401
$\beta_6$	55.3698	61.1982	$\beta_{21}$	42.6302	47.1176
$\beta_7$	46.7945	51.7203	$\beta_{22}$	35.7783	39.5445
$\beta_8$	35.1295	38.8273	$\beta_{23}$	28.7235	31.7471
$\beta_9$	27.0282	29.8732	$\beta_{24}$	21.8111	24.1071
$\beta_{10}$	22.9273	25.3407	$\beta_{25}$	22.335	24.686
$\beta_{11}$	45.8818	50.7114	$\beta_{26}$	35.0848	38.778
$\beta_{12}$	40.3341	44.5797	$\beta_{27}$	30.366	33.5624
$\beta_{13}$	33.17	36.6616	$\beta_{28}$	24.4629	27.0379
$\beta_{14}$	23.9992	26.5254	$\beta_{29}$	19.1359	21.1503
$\beta_{15}$	22.3561	24.7093	$\beta_{30}$	21.5761	23.8473

Επίσης, κάποιες βασικές παράμετροι του EASY παρουσιάζονται στον πίνακα 7.9.

**Πίνακας 7.9:** Βελτιστοποίηση ενός δρομέα Karlan χρησιμοποιώντας τον EASY και τον PUMA. Τιμές για τις βασικές παραμέτρους του EASY.

Παράμετρος	Σύμβολο	Τιμή
Μέγεθος πληθυσμού γονέων	$\mu$	15
Μέγεθος πληθυσμού απογόνων	$\lambda$	30
Μέγεθος πληθυσμού elite	$e$	20
Πιθανότητα διασταλурωσης	—	0.9
Μέγιστος αριθμός αξιολογήσεων	-	12000
Κωδικοποίηση μεταβλητών	-	Πραγματική

## 7.5.2 Αντικειμενικές Συναρτήσεις

Για τη βελτιστοποίηση θέτονται δυο αντικειμενικές συναρτήσεις:

- Η μεγιστοποίηση του βαθμού απόδοσης που εκφράζεται ως

$$F_1 = \min(-\eta) \quad (7.12)$$

- Η μεγιστοποίηση της ελάχιστης εμφανιζόμενης στατικής πίεσης, με σκοπό την αποφυγή εμφάνισης του φαινομένου της σπηλαίωσης. Αυτή η αντικειμενική συνάρτηση εκφράζεται ως

$$F_2 = \min(-p_{s,min}) \quad (7.13)$$

### 7.5.3 Περιορισμοί

Ο περιορισμός που τίθενται αφορά στις αποκλίσεις των κατανομών των αδιάστατων απόλυτων ταχυτήτων (περιφερειακή  $c_u^*$  και μεσημβρινή  $c_m^*$ ) στην έξοδο του δρομέα από κάποιες αντίστοιχες επιθυμητές κατανομές ταχυτήτων. Αυτός ο περιορισμός εξασφαλίζει τις ελάχιστες δυνατές απώλειες στον αγωγό εξόδου και εκφράζεται ως

$$C_m \leq 0.04$$

$$C_u \leq 0.04 \quad (7.14)$$

$$(7.15)$$

όπου:

$$C_m = \int_{A_{out}} (c_m^* - c_{m,tar}^*)^2 dr$$

$$C_u = \int_{A_{out}} (c_u^* - c_{u,tar}^*)^2 dr \quad (7.16)$$

$$(7.17)$$

### 7.5.4 Αποτελέσματα της Βελτιστοποίησης

Μετά από 21 γενιές του EA, το προκύπτον μέτωπο Pareto παρουσιάζεται στο σχήμα 7.28. Στο σχήμα σημειώνονται οι δυο ακραίες βέλτιστες λύσεις ( $A$  και  $C$ ), καθώς επίσης και μια ενδιάμεση ( $B$ ).

Στο σχήμα 7.29, παρουσιάζονται οι κατανομές  $C_p$  σε τρεις ακτινικές θέσεις για τις τρεις βέλτιστες λύσεις ( $A$ ,  $B$ ,  $C$ ) σε σχέση με την αντίστοιχη κατανομή της αρχικής γεωμετρίας. Παράλληλα, στον πίνακα 7.10, παρουσιάζεται η εκατοστιαία αύξηση του βαθμού απόδοσης σε σχέση με την αρχική γεωμετρία.

Τέλος, στο σχήμα 7.30, παρουσιάζονται οι προκύπτουσες κατανομές των αδιάστατων απόλυτων ταχυτήτων σε σχέση με τις αντίστοιχες επιθυμητές κατανομές που επιβάλλο-

**Πίνακας 7.10:** Βελτιστοποίηση ενός δρομέα Karlan χρησιμοποιώντας τον EASY και τον PUMA. Εκατοστιαία αύξηση στο βαθμό απόδοσης για κάθε επιλεχθείσα βέλτιστη λύση σε σύγκριση με την αρχική γεωμετρία.

Βέλτιστη λύση	Εκατοστιαία μεταβολή στο β.α. (%)
Βέλτιστη λύση A	+1.58
Βέλτιστη λύση B	+0.34
Βέλτιστη λύση C	+0.01

νται από τον περιορισμό, ενώ στον πίνακα 7.11 φαίνονται και οι τιμές που υπολογίζονται από τις εξ. 7.16.

**Πίνακας 7.11:** Βελτιστοποίηση ενός δρομέα Karlan χρησιμοποιώντας τον EASY και τον PUMA. Οι προκύπτουσες τιμές για τους περιορισμούς, όπως υπολογίζονται από τις εξ.7.14 για κάθε επιλεχθείσα βέλτιστη λύση.

Βέλτιστη λύση	Τιμή του περιορισμού για το $c_m^*$	Τιμή του περιορισμού για το $c_u^*$
Βέλτιστη λύση A	0.01134	0.01688
Βέλτιστη λύση B	0.00881	0.00287
Βέλτιστη λύση C	0.00852	0.00365

## 7.6 Σύνοψη και Συμπεράσματα - Ιδέες για Μελλοντική Δουλειά

### 7.6.1 Σύνοψη και Συμπεράσματα

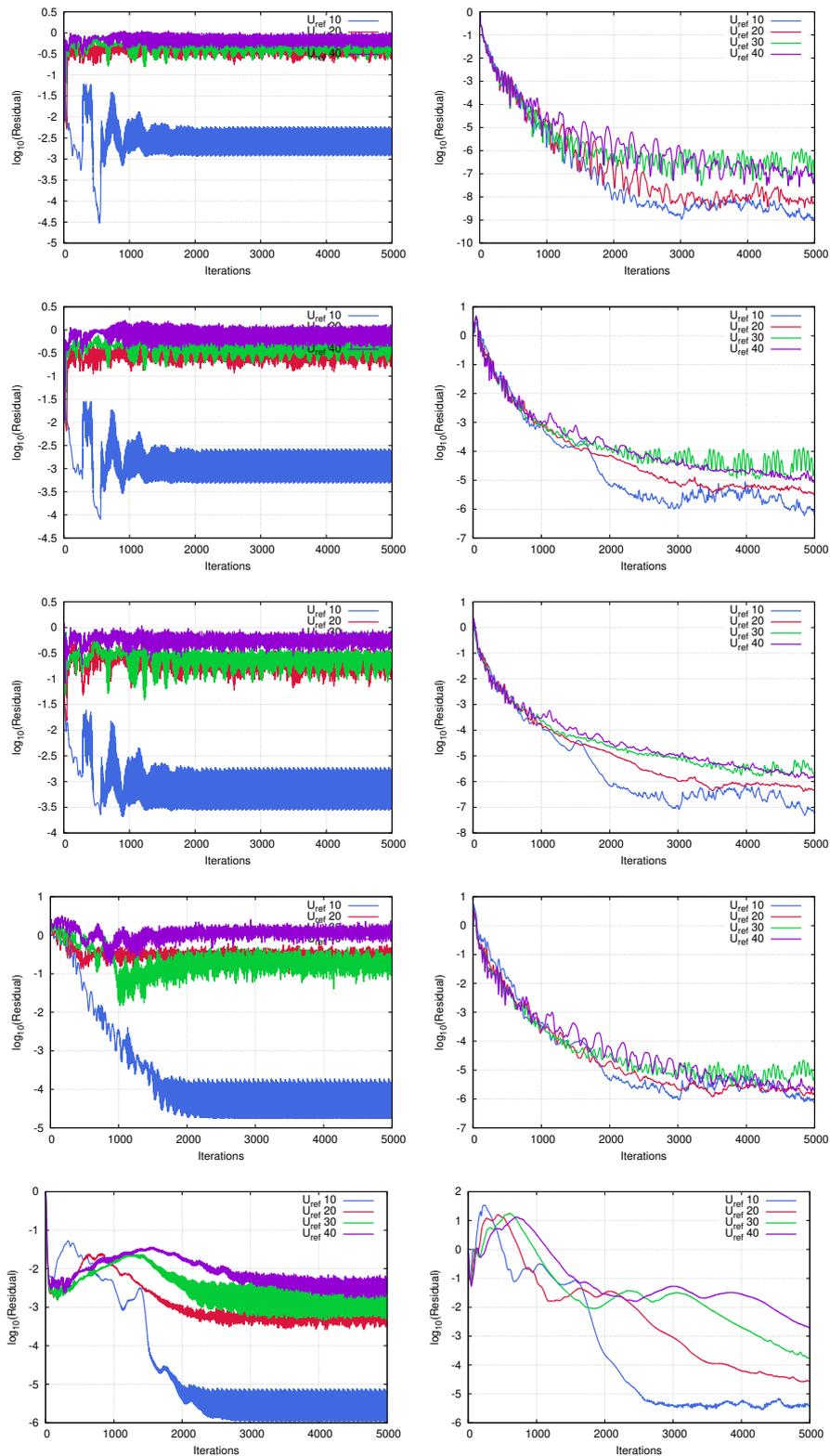
Η ένταξη και η πιστοποίηση του επιλύτη ροής PUMA στο βιομηχανικό εργαλείο ανάλυσης και βελτιστοποίησης της AndritzHydro θεωρείται επιτυχής. Συγκεκριμένα, τα ακόλουθα συμπεράσματα προκύπτουν:

- Η ομαλή ένταξη του PUMA στο βιομηχανικό εργαλείο ήταν επιτυχής. Το αποτέλεσμα είναι μια ενιαία αλυσίδα εργαλείων η οποία είναι φιλική προς το χρήστη και χρησιμοποιείται από τους σξεδιαστές της AndritzHydro.
- Η σύγκριση του επιλύτη PUMA με τον εμπορικό επιλύτη είναι, επίσης, επιτυχής, αφού οι αποκλίσεις μεταξύ των δυο επιλυτών είναι αμελητέες. Έτσι, περισσότεροι σχεδιαστές ενθαρρύνονται να χρησιμοποιήσουν τον επιλύτη PUMA για να τρέξουν τις προσομοιώσεις τους.
- Η χρήση του PUMA ως εργαλείο αξιολόγησης για τον EASY φαίνεται να είναι μια βιώσιμη επιλογή, αφού τα αποτελέσματα της βελτιστοποίησης είναι ικανοποιητικά και το υπολογιστικό κόστος είναι σχετικά μικρό.

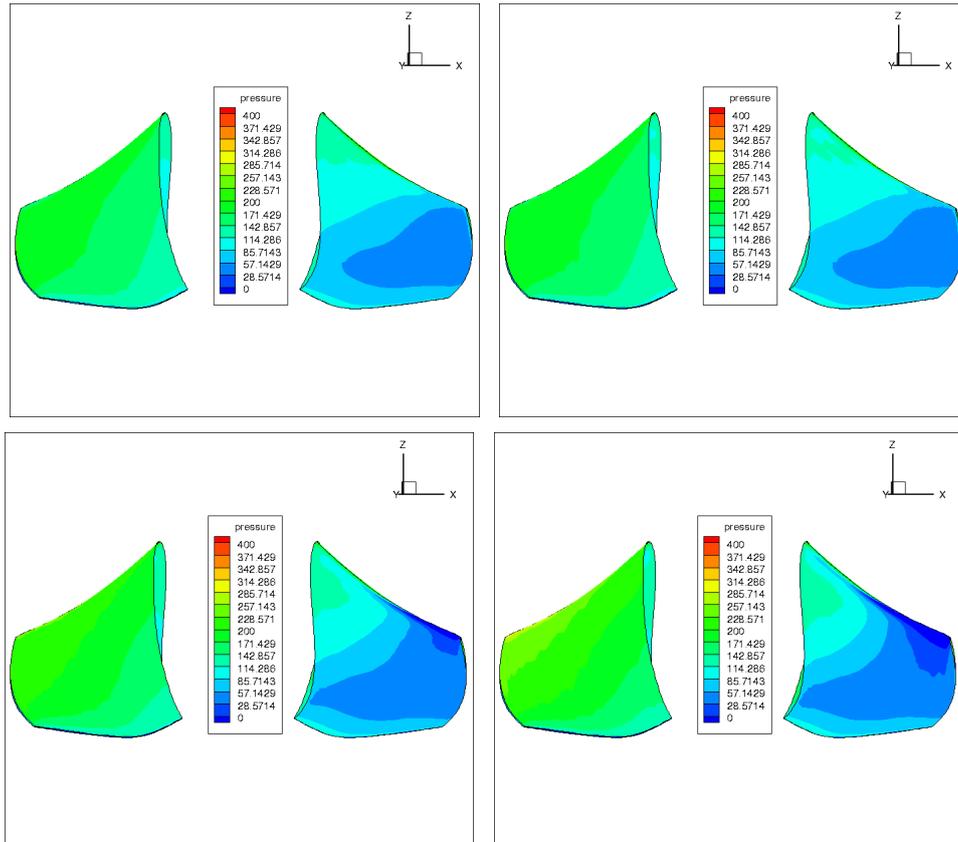
## 7.6.2 Προτάσεις για Μελλοντική Δουλειά

Όσον αφορά ιδέες και προτάσεις για μελλοντική δουλειά, τα ακόλουθα παρατίθενται:

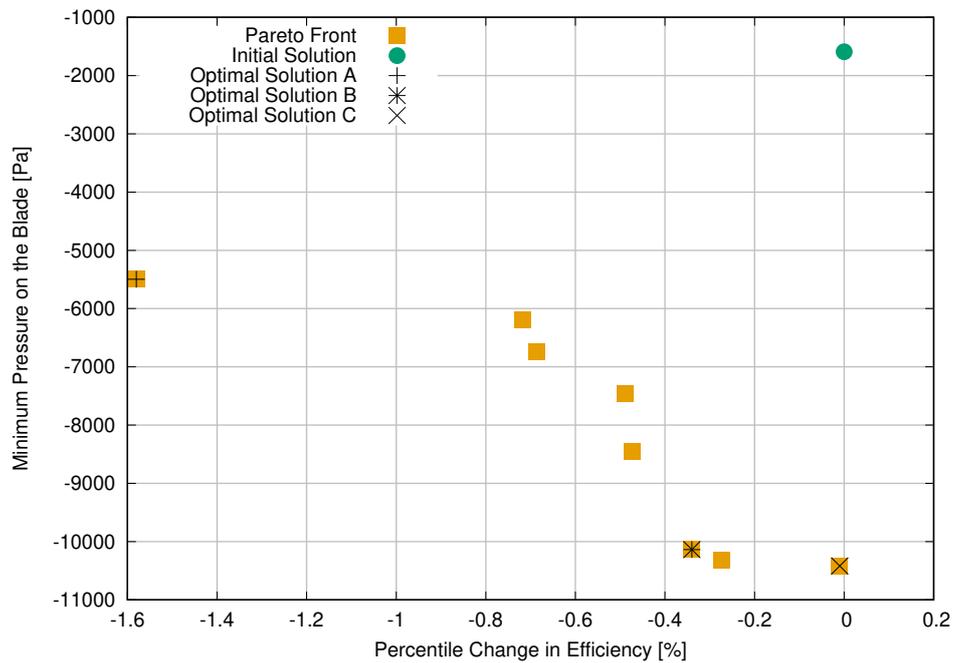
- Περαιτέρω ανάπτυξη των μετατροπών του τύπου του πλέγματος.
- Η ένταξη των μετατροπών μέσα στον επιλύτη PUMA.
- Περαιτέρω δοκιμή του επιλύτη PUMA, μελετώντας ακόμη πιο περίπλοκες διάταξεις υδροστροβίλων, με σκοπό την διερεύνηση των δυνατοτήτων του PUMA.
- Προσομοίωση και δοκιμή πραγματικά μη-μόνιμων ροών ασυμπίεστου ρευστού μέσα στους υδροστροβίλους.



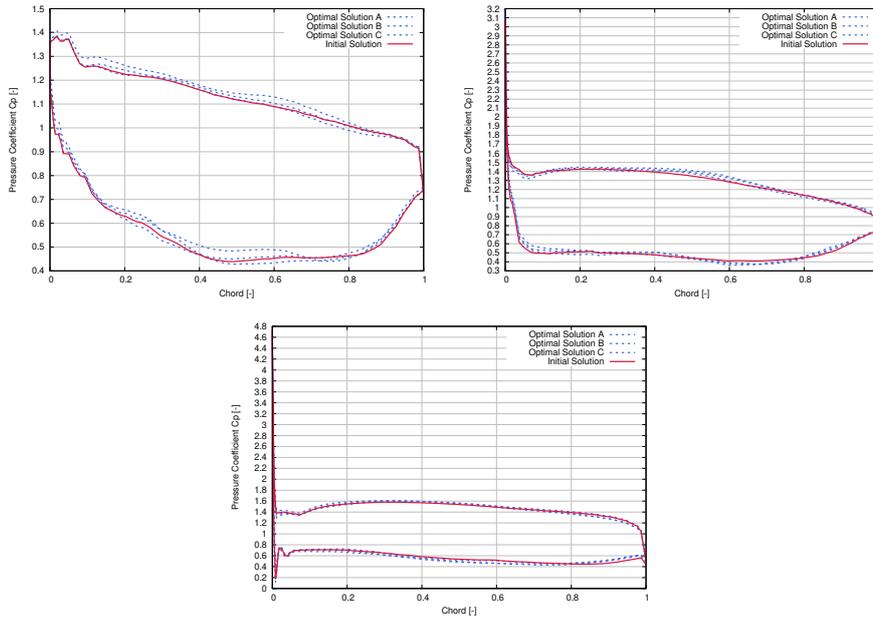
**Σχήμα 7.26:** Παραμετρική μελέτη των οδηγιών περυγίων και του δρομέα του υδροστροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Ρυθμός σύγκλισης των πέντε εξισώσεων για τα οδηγία πτερύγια (αριστερά) και το δρομέα (δεξιά) για τις τέσσερις τιμές της  $U_{ref}$  (10, 20, 30, 40  $\frac{m}{s}$ ).



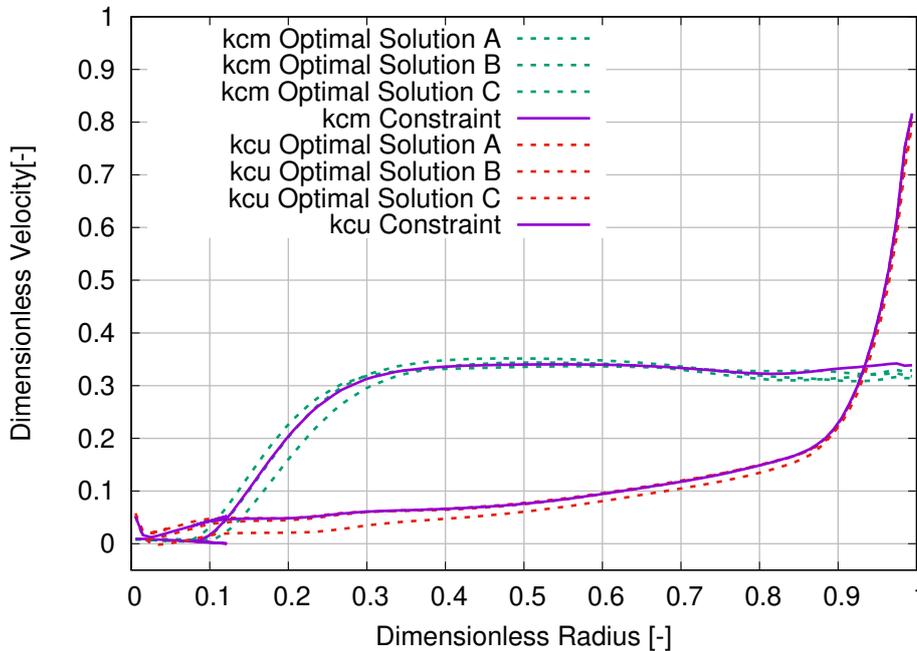
**Σχήμα 7.27:** Παραμετρική μελέτη των οδηγών πτερυγίων και του δρομέα του υδρο-στροβίλου τύπου προπέλα χρησιμοποιώντας τον PUMA. Τα προκύπτοντα πεδία πίεσης του πτερυγίου του δρομέα για τις τέσσερις τιμές της  $U_{ref}$  (πάνω-αριστερά προς κάτω-δεξιά, 10, 20, 30 ανδ 40  $\frac{m}{s}$ ).



**Σχήμα 7.28:** Βελτιστοποίηση ενός δρομέα Karlan χρησιμοποιώντας τον EASY και τον PUMA. Το μέτωπο Pareto της δικριτριακής βελτιστοποίησης. Κάθε τετραγωνικό σημείο αναπαριστά μια βέλτιστη λύση με τις αντίστοιχες τιμές των δυο αντικειμενικών συναρτήσεων ( $F_1$  και  $F_2$ ). Επίσης, στην πάνω-δεξιά γωνία βρίσκεται η αρχική γεωμετρία (πράσινος κύκλος).



**Σχήμα 7.29:** Βελτιστοποίηση ενός δρομέα Karman χρησιμοποιώντας τον EASY και τον PUMA. Η κατανομή του  $C_p$  για τις τρεις επιλεγθείσες βέλτιστες λύσεις και την αρχική γεωμετρία σε τρεις ακτινικές θέσεις, τη ρίζα (πάνω-αριστερά), το μέσο (πάνω-δεξιά) και το άκρο (κάτω) του περυγίου.



**Σχήμα 7.30:** Βελτιστοποίηση ενός δρομέα Karman χρησιμοποιώντας τον EASY και τον PUMA. Οι κατανομές των αδιάστατων απόλυτων ταχυτήτων ( $c_u^*$  και  $c_m^*$ ) στην έξοδο του δρομέα σε σύγκριση με τις αντίστοιχες επιθυμητές κατανομές.

## Bibliography

- [1] Kyriacou, S.: *Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery*. PhD thesis, Lab of Thermal Turbomachines, N.T.U.A., Athens, 2013.
- [2] *Turbine Selection, webpage*. <http://rivers.bee.oregonstate.edu/book/export/html/35>.
- [3] Καμπόλης, Ι.: *Πολυεπίπεδοι, πολυεπεξεργαστικοί αλγόριθμοι αεροδυναμικής βελτιστοποίησης στις στροβιλομηχανές*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2009.
- [4] Τρομπούκης, Ξ.: *Αριθμητική επίλυση προβλημάτων αεροδυναμικής-αεροελαστικότητας σε επεξεργαστές καρτών γραφικών*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2012.
- [5] Τσιάκας, Κ.: *Ανάπτυξη μεθόδων βελτιστοποίησης με χρήση επεξεργαστών καρτών γραφικών και εφαρμογή στις στροβιλομηχανές*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, σε εξέλιξη.
- [6] Chorin, A.: *A numerical method for solving incompressible viscous flow problems*. Journal of Computational Physics, 2(1):12–26, 1967.
- [7] Turkel, E.: *Preconditioned methods for solving the incompressible and low speed compressible equations*. Journal of Computational Physics, 72:277–298, 1987.
- [8] Spalart, P. and Allmaras, S.: *A one-equation turbulence model for aerodynamic flows*. AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA, January 6–9 1992.
- [9] Spalding, D. B.: *A single formula for the law of the wall*. Journal of Applied Mechanics, 28:455–457, 1961.
- [10] Κουμπογιάννης, Δ.: *Αριθμητική επίλυση των εξισώσεων Navier–Stokes με χρήση μη-δομημένων πλεγμάτων σε περιβάλλον παράλληλης επεξεργασίας*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 1998.

- [11] Ασούτη, Β.: *Μέθοδοι αεροδυναμικής ανάλυσης και σχεδιασμού για ροές υψηλών και χαμηλών ταχυτήτων, σε πολυεπεξεργαστικό περιβάλλον*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2009.
- [12] Ζερβογιάννης, Θ.: *Μέθοδοι βελτιστοποίησης στην αεροδυναμική και τις στροβιλομηχανές με χρήση συζυγών τεχνικών, υβριδικών πλεγμάτων και του ακριβούς εσσιανού μητρώου*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2011.
- [13] Kampolis, I., Trompoukis, X., Asouti, V., and Giannakoglou, K.: *CFD-based analysis and two-level aerodynamic optimization on Graphics Processing Units*. *Computer Methods in Applied Mechanics and Engineering*, 199(9–12):712–722, 2010.
- [14] Rumsey, C. L., Poirier, D. M. A., Bush, R. H., and Towne, C. E.: *CFD General Notation System A User's Guide To CGNS*. 2011.
- [15] CGNS Project Group: *CFD General Notation System Standard Interface Data Structures*. 2011.
- [16] Zormpa, M.: *Programming of a "Back-to-CAD" Tool for Use in the Analysis and Optimization of Turbomachinery Rows. Industrial Applications*. Diploma Thesis. Lab of Thermal Turbomachines, NTUA, 2018.
- [17] *Nanobubbles*, webpage. <http://www.nanobubbles.com/nanobubbles-2/how-are-nanobubbles-generated/#.W6klnuV1PIV>.
- [18] *Hydrodynamics of Pumps*, webpage. <https://authors.library.caltech.edu/25019/1/chap6.htm>.