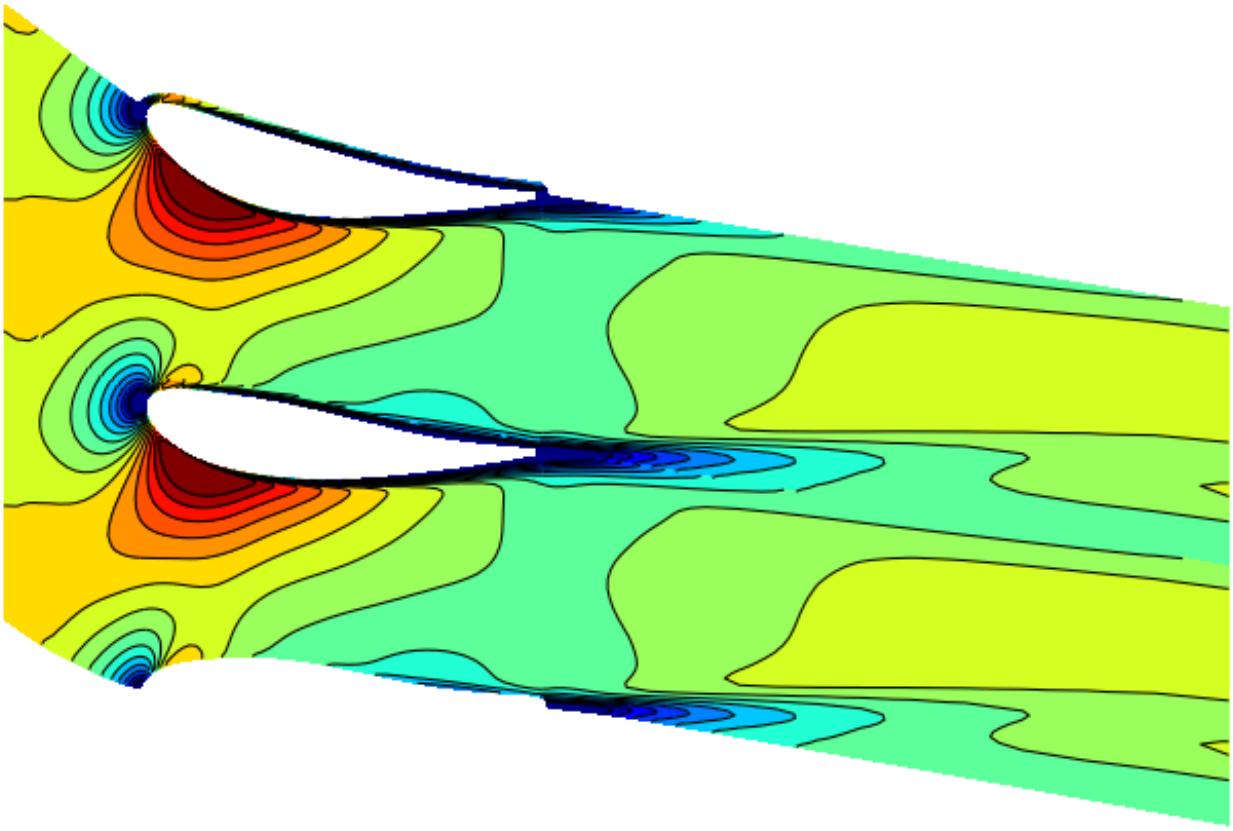
CHALMERS
UNIVERSITY OF TECHNOLOGY



Evaluation of an in-house GPU based CFD solver

Master's thesis in Applied mechanics

Christoffer Johansson

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

MASTER'S THESIS IN APPLIED MECHANICS

Evaluation of an in-house GPU based CFD solver

Christoffer Johansson

Department of Mechanics and Maritime Sciences
Division of Fluid mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Evaluation of an in-house GPU based CFD solver

Christoffer Johansson

Master's Thesis 2018:12
Department of Mechanics and Maritime Sciences
Division of Fluid mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Mach number at span wise coordinate 0.25 for TRS4 at ADP. Simulation from the CUDA solver.

Department of Mechanics and Maritime Sciences
Gothenburg, Sweden 2018-08-02

Evaluation of an in-house GPU based CFD solver

Master's thesis in Master's Applied mechanics
Christoffer Johansson
Department of Mechanics and Maritime Sciences
Division of Fluid mechanics
Chalmers University of Technology

Abstract

This thesis describes the evaluation of an in-house CFD solver at GKN aerospace. The solver is called CUDA since it is GPU based, and CUDA cores are thereby used. Legacy turbine rear structures, designs from previous projects, are used when evaluating the solver.

The first part of this thesis compares two different versions of CUDA. A correction in the definition of axial wall shear stress makes the new version more correct. By post-processing flow parameters directly from the solution the new version is 25% faster than the previous version. The new version was officially released as a result of the validations performed in this thesis.

The second part of this thesis evaluates the CUDA solver by comparing CFD results with the commercial CFD solver Fluent. CUDA shows similar trends as fully resolved k-ε realizable in Fluent. CUDA uses k-ε with realizability limiters as well, but for the wall treatments wall-functions are used. One evaluated flow parameter is total pressure loss. The pressure loss is presented in three different ways with respect to the inlet swirl angle, using a so called loss bucket and also two factors with normalized result. The two normalized factors are called off-design factor and loss difference. For all these three ways of presenting the pressure loss CUDA and Fluent predicts similar trends for all turbine rear structures. This correlation in pressure loss is predicted until separation occurs. Separation occurs later for CUDA than Fluent. For CUDA the point of separation is predicted to occur in average at 4 degrees more swirl than fully resolved k-ε realizable and 11.5 degrees later than k-ω SST for the turbine rear structures. The pressure loss is predicted to be lower in CUDA. CUDA predicts between 1% and 13% lower pressure loss than fully resolved k-ε realizable in Fluent, in average 6.75%.

Key words: Aerodynamics, CFD solver, CUDA, GKN aerospace, GPU, in-house code, jet engine, turbine rear structure, pressure loss.

Contents

Preface

This thesis has been carried out from January to July 2018. The work was performed at GKN aerospace in Trollhättan. Together with the two supervisors at GKN, Pär Nylander and Jonas Larsson, the evaluation of the in-house CFD solver has been possible. I would like to thank both of them for the help I got during this period of work. With help from some more colleagues, especially Lars Ellbrant and Jonas Karlsson, description of the infrastructure for the CUDA solver could be included in the thesis.

I would also like to thank Niklas Andersson, who is the examiner of this thesis. Niklas works at the department of Mechanics and Maritime Sciences, at Chalmers University of Technology. Thereby, this thesis belongs to that department.

Gothenburg 2018-08-02

Christoffer Johansson

Nomenclature

Roman letters

C_p	heat capacity at constant pressure
C_v	heat capacity at constant volume
$c_{\varepsilon 1}$	constant
$c_{\varepsilon 2}$	constant
c_μ	constant
e	energy
e_0	total enegry
g	gravitational constant
k	turbulent kinetic energy
M	Mach number
P	static pressure
P0	total pressure
Pr	laminar Prandtl number
q_j	heat flux (given by Fourier's law)
S_{ij}^*	trace-less viscous strain rate tensor
T	temperature
t	time
u	velocity
u_τ	wall friction velocity
x	locational coordinate axial
y	locational coordinate sideways direction (wall distance)
y^+	normalized wall distance

Greek letters

β	volumetric constant
γ	heat capacity ratio
δ_{ij}	Kronecker delta
ε	turbulent dissipation
θ	temperature
μ	dynamic viscosity
ν	kinematic viscosity
ν_t	turbulent kinematic viscosity
π	pi
ρ	density
σ_k	constant
σ_ε	constant
τ_{ij}	viscous stress tensor
τ_w	wall shear stress
ψ	generic flow parameter
ω	specific dissipation rate

Mathematical notation

$\frac{\partial}{\partial}$	partial derivative
$\overline{\psi}$	generic flow parameter mean
ψ'	generic flow parameter fluctuation
$\lvert\psi\rvert$	generic flow parameter absolute value

Abbreviations

ADP	aerodynamic design point
BC	boundary conditions
CFD	computational fluid dynamics
CUDA	compute unit device architecture
DEV	development
FVM	finite volume method
geo	geometry
GPU	graphics processing unit
hp	credits of university
kwSST	k-ω shear stress transport
LowRe	low Reynolds
LPT	low pressure turbine
Non-eq-wf	non equilibrium wall-function
OEM	original equipment manufacturer
PD	pressure distortion factor
PDE	partial differential equation
RKE	k-ε realizable
PROD	production
profout	outlet profile
RANS	Reynolds averaged Navier-Stokes
TRS	turbine rear structure

1 Introduction

This thesis is about an in-house CFD code. This CFD code can be used for simulating the flow in one of the components in a jet engine. The work presented in this thesis includes both knowledge in aerodynamics and programming. The evaluation of the code is done to ensure the accuracy and documentation of the solver.

1.1 Background

GKN aerospace engine systems Sweden AB, now owned by Melrose plc, is one of the world leading companies for design and manufacturing of aero-engine components. When flying in a commercial airplane, there is a great possibility that the engines contain parts from GKN. As about 90% of all large commercial engines contain parts from GKN. This is due to cooperation with several large original equipment manufacturers (OEM).

One of the last part of a commercial jet engine is a component called the turbine rear structure (TRS), see the marked component in Figure 1. The TRS has two major purposes:

- The first purpose is to remove the rotational angle of the exhaust jet flow. The rotational angle of the flow is called swirl. Swirl appears in the flow because of rotating components in the engine. One of the rotating components is the low pressure turbine (LPT), which is located directly in front of the TRS, creating the incoming swirl. The reason to remove the swirl of the flow is to maximized propulsion force in the direction of the aircraft. With stationary vanes in the TRS, the exhaust jet flow is guided in the direction of the aircraft.

- The second purpose is to connect the engine to the aircraft. This is done by using three different designs for the vanes in the TRS. Regular, tube and mount vanes. Tube vanes are thicker than the regular vanes due to that oil-pipes have to fit inside these vanes. Mount vanes transfers the structural loads from the low pressure turbine axis to engine mounts, attached to the wing of the aircraft. The mount vanes have a bump at the shroud (outer case of the TRS) to reduce the bending moment due to the engine mounts. In this thesis only regular and tube vanes are used for simulations.

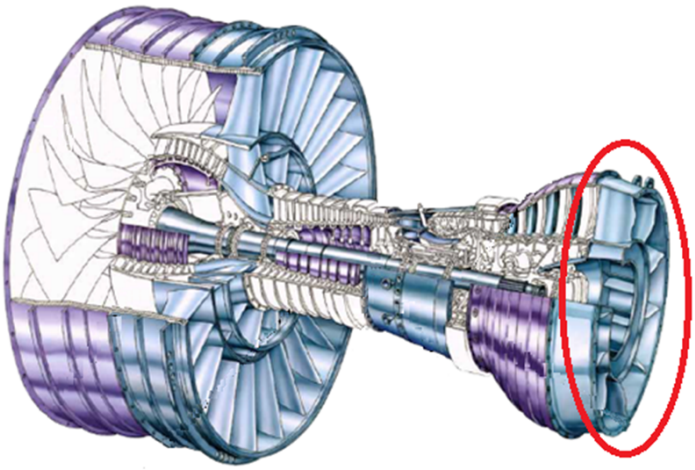


Figure 1 A commercial jet engine, with the turbine rear structure noted. Figure from [1].

To design vanes in the TRS, GKN has developed an in-house code called VolVane. VolVane uses a camber line together with a thickness distribution to define vane profiles. The aerodynamic performance of the vanes are evaluated using commercial CFD codes, such as Fluent or CFX. CFD means computational fluid dynamics, which is an engineering method to determine the flow field by solving an approximation of the governing equations. Recently an in-house, GPU (graphic process unit) based, CFD solver called CUDA has been developed and integrated into VolVane. GPU-based solvers have many slow cores instead of a few fast ones. The calculations for CFD simulations are divided into many easier calculations. This makes GPU more profitable in terms of number of calculations needed [2].

1.2 Purpose

The main purpose of this thesis is to evaluate the CUDA solver, using legacy components, compared to a commercial CFD solver, which in this thesis is Fluent. Since there is a new version of the CUDA solver this version has to be verified first. The differences between the versions of CUDA are in the post-processing. Hence, it is this part of the process that needs to be compared between the versions. The evaluation and comparison is done by analyzing flow parameters.

1.3 Problem description

The problem description of this thesis has focused on three different topics: how the solver works, verification of the new CUDA version and evaluation of CUDA compared to the commercial software Fluent.

Simulation process:

- How does the infrastructure of the simulation process work?
- How does the code for the post-processing work?

CUDA versions:

- Is the new version of the solver ready to be implemented?

Evaluation of the CUDA solver:

- Are the results comparable between CUDA and Fluent?
- Can CUDA be used in the design process of TRS?
- Are there some limitations of CUDA that needs to take into account when designing a TRS?

1.4 Limitations

In 20 weeks, corresponding to 30 hp (credits of the university), the work will be both carried out and documented. Since the time is limited, some limitations of the project are needed. The infrastructure and post-processing will be documented, but not the source code. The source code is written in C++ and will not be investigated at all. For the components analyzed there will only be simulations of TRS. There will not be any simulations from software besides Fluent and CUDA.

2 Theory

This Chapter has two major topics. The first topic describes the fluid mechanics needed to understand the results and the analyses. The majority of the theory is retrieved from Davidson L (2018) [3]. The second topic describes the infrastructure of the in-house solver. The infrastructure includes every process from setting boundary conditions to getting the post-processed results.

2.1 Fluid mechanics

In aerodynamic simulations the engineering strategy CFD is used. CFD predicts an approximation of the governing equations of the flow. The governing equations for the fluids that are used in, for example, a jet engine are called Navier-Stokes, see equations (1-4). These equations are partial differential equations (PDE) that demands a lot of computational power to be solved. In PDE there are derivatives of more than one variable included in the equations. The PDE are coupled as well, which means that all variables are dependent on each other.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j\right) = 0 \tag{1}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_i u_j + P\delta_{ij} - \tau_{ji}\right] = 0 \; for \; i = 1,2,3 \tag{2a}$$

$$\tau_{ij} = 2\mu S_{ij}^* \tag{2b}$$

$$S_{ij}^* \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{2c}$$

$$\frac{\partial(\rho e_0)}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_j e_0 + u_j P + q_j - u_i \tau_{ij}\right] = 0 \tag{3a}$$

$$q_j \equiv -C_p \frac{\mu}{Pr}\frac{\partial T}{\partial x_j} \tag{3b}$$

$$e_0 \equiv e + \frac{x_k x_k}{2} \tag{3c}$$

$$P = \rho R T\,, \qquad \gamma \equiv \frac{C_p}{C_v}\,, \qquad e = C_v T\,, \qquad C_p - C_v = P \tag{4}$$

The equations above describes the governing equations in Navier-Stokes, in tensor format. Equation (1) is the mass conservation equation, also called the continuity equation. The set of equations (2a-c) describes the momentum equations for Navier-Stokes. Equation (2a) is for all three directions in a Cartesian coordinate system. In equation (2b) the viscous stress tensor for Newtonian fluids, assuming Stokes law for mono-atomic gases, is introduced. The definition for the trace-less viscous strain-rate tensor is defined as in equation (2c). The set of equations (3a-c) describes the energy equations, where the heat flux and the total energy is defined in (3b) and (3c). To close all these equations a thermodynamic state has to be given. The equations (4) have to be used, assuming that the gas is calorically perfect. The equations include constants for the fluid: γ the heat capacity ratio, C_p the heat capacity at constant pressure, C_v the heat capacity at constant volume, Pr the laminar Prandtl number and R the gas constant. The symbol δ_{ij} denotes Kroneckers delta, which is tensor notation for 1 if the indices are equal and 0 if $i \neq j$ [4].

2.1.1 Turbulence models

Due to that Navier-Stokes demands a lot of computational power to solve, approximations have been introduced in the equations to reduce the computations. One way of reducing the number of computations are to average the equations. The solution will then give averaged quantities, but most of the information about the flow will still be determined. The equations are often averaged in time, and therefore the information of the time dependence is lost. After averaging the governing equations, they are called Reynolds averaged Navier-Stokes (RANS). In the derivation of RANS new unknowns terms appear, due to decomposition of unsteady to mean and fluctuating quantities

$$\psi = \bar{\psi} + \psi' \tag{5}$$

where ψ is any flow parameter. These new quantities that appear are called Reynolds stresses

$$\overline{v_i' v_j'} \tag{6}$$

and they have to be modelled in the computations. Modeling terms in equations is the second way of simplifying the computations. Modeling means that you approximate terms that are costly to calculate, with relations of variables that are less costly to determine. The Reynolds stresses are modelled with Boussinesq assumption

$$\overline{v_i' v_j'} = -\nu_t \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) + \frac{2}{3} \delta_{ij} \overline{v_k' v_k'} \tag{7}$$

where ν_t is turbulent viscosity. Turbulent viscosity is a property of the flow, not the fluid, which can be estimated as

$$\nu_t = c_\mu \frac{k^2}{\varepsilon} \tag{8}$$

where c_μ is a constant (further explanation can be found in Section 2.1.1.1), k is the turbulence kinetic energy and ε is the turbulence dissipation. Dissipation is the energy that transfers into the flow as losses.

When solving the equations different models can be used to determine the parameters such as the kinetic energy and the dissipation. Two of the most accepted models are k-ε realizable [5] and k-ω SST [6] which are presented in the Sections below. Realizable k-ε (RKE) has shown good agreement with wind tunnel tests. k-ω SST is the most commonly used model for CFD simulations, which makes it a good model for comparing with results from previous studies.

2.1.1.1 k-ε realizable

The k-ε model solves the transport equations for turbulent kinetic energy and dissipation.

$$\frac{\partial(\rho k)}{\partial t} + \bar{v}_j \frac{\partial(\rho k)}{\partial x_j} = \rho \nu_t \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial \bar{v}_i}{\partial x_j} + \rho g_i \beta \frac{\nu_t}{\sigma_\theta} \frac{\partial \bar{\theta}}{\partial x_i} - \rho \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial(\rho k)}{\partial x_j} \right] \tag{9}$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \bar{v}_j \frac{\partial(\rho \varepsilon)}{\partial x_j} = \frac{\rho \varepsilon c_{\varepsilon 1} \nu_t}{k} \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial \bar{v}_i}{\partial x_j} + \frac{\rho c_{\varepsilon 1} g_i \varepsilon \nu_t}{k \sigma_\theta} \frac{\partial \bar{\theta}}{\partial x_i} - \frac{\rho c_{\varepsilon 2} \varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial(\rho \varepsilon)}{\partial x_j} \right] \tag{10}$$

The constants ($c_\mu, c_{\varepsilon 1}, c_{\varepsilon 2}, \sigma_k, \sigma_\varepsilon$) in the equations have the value of (0.09, 1.44, 1.92, 1, 1.3) and β is volumetric thermal expansion. The variables solved for in the equations are defined as

$$k = \frac{1}{2}\overline{v_i' v_i'} \tag{11}$$

$$\varepsilon = \nu \overline{\frac{\partial v_i' \partial v_i'}{\partial v_j \partial v_j}}. \tag{12}$$

The term which is equal to k appears in RANS and ε is a physical property describing a part of the governing equations. The variables appears in each other's transport equation, which make them coupled to each other.

The realizable criterion in the model is added to force the solution to make physical sense. The criterion has two parts

$$\overline{v_i'^2} \geq 0 \text{ for all } i \tag{13}$$

$$\frac{\left|\overline{v_i' v_j'}\right|}{\left(\overline{v_i'^2}\ \overline{v_j'^2}\right)^{1/2}} \leq 1 \text{ no summation over } i \text{ and } j, i \neq j \tag{14}$$

where the first criterion is mathematical and the second is physical. The first criterion is explained as no real value squared can be negative. The second criterion is explained with a derivation which gives that normal stresses stay positive.

2.1.1.2 k-ω SST

Instead of solving for the dissipation, the k-ω SST model solves for the specific dissipation rate ω. The relation

$$\omega = \frac{\varepsilon}{c_\mu k} \tag{15}$$

relates the dissipation to the specific rate. The simple relation in equation (15) tells that the transport equations looks similar as well. When solving for the specific dissipation rate the solutions agrees well with tests of the flow close to the wall. The solution is more costly than RKE though, and predicts less accurate solutions in the free flow.

SST stands for shear stress transport and by modifying the transport equations for k and ω in two ways the model becomes very cost efficient, but still accurate. The SST model uses RKE away from the wall and also a limitation of the shear stress (friction between fluid particles) in regions with adverse pressure gradients (regions where the fluid separates from the surface and changes direction). RKE generally over-predicts the shear stress in adverse pressure gradient situations, but when using k-ω in these regions with an extra limiter the flow behaves like in tests.

The equations for k-ω SST are more complicated than both standard k-ε and k-ω. It uses both damping functions for the limitation of the shear stress and conditions for when to use the different turbulence models.

2.1.1.3 Wall treatment

In CFD it is always difficult to predict the flow close to walls. The assumption of "no-slip" boundary condition implies that the velocity by the wall is zero. This assumption is argumented for in a physical way. Since the velocity a bit away from the wall is non zero, large gradients makes the flow difficult to predict in these regions. Hence, the calculations have to be more resolved in these difficult regions.

CFD is based on the finite volume method (FVM), where flow parameters are stored in small boxes (called cells). These cells together creates a mesh and in the simulations these parameters flow between the cells, iterated and thereby determined by the governing equations, until equilibrium is achieved. In the regions where the solution have large gradients (for example close to the wall), there is need for higher resolution of the mesh. More cells, in order to get higher resolution, is costly in terms of calculations and thereby time. By creating a finer mesh with smaller cells the solution may get more accurate, but the time spent on the simulation is not always worth the new information. If the mesh is fine enough to resolve the boundary layer (the flow closest to the wall) the mesh is called a Low-Reynolds mesh (LowRe). Instead of resolving the boundary layer the theory from the flow on a flat plate can be applied as approximation of the flow close to the call. This is known as using wall-functions, and in this approximation of the flow the cells can be relatively large even close to the wall and thereby save computational time. Wall-functions are used in VolVane. In Fluent both resolved meshes and wall-functions can be used. After a short wall-function study non-equilibrium wall-functions were used in Fluent, to verify the results from CUDA (see more in Appendix A). This wall-function uses a modification of the mean velocity to change more easily to pressure-gradients, and also the cells closest to the wall uses a modification for the turbulent kinetic energy.

There are several rules of thumb for how to create a good mesh. To have cubic cells minimizes the computational errors, but since the mesh have to be more resolved close to the wall and the walls can also be curved, this is difficult to follow. When deciding on cell size, the non-dimensional number y^+ is used. In fluid mechanics it is convenient to use non-dimensional values, this is because the scales of the flow is calculated afterwards for the specific case. The governing equations are the same if the scales are non-dimensionalized. y^+ is know as the normalized wall distance

$$y^+ = \frac{y u_\tau}{\nu} \tag{16}$$

where y is the distance to the wall and u_τ is the wall friction velocity

$$u_\tau = \left(\frac{\tau_w}{\rho}\right)^{1/2} \tag{17}$$

and τ_w is the wall shear stress

$$\tau_w = \mu \left.\frac{\partial \bar{v}_1}{\partial y}\right|_{wall}. \tag{18}$$

Close to the wall where y^+ is below 10 the flow velocity increases rapidly when the distance from the wall increases. When y^+ is around 30 the increment of the velocity increases logarithmically in relation to the wall distance (called the log-law region). The wall-treatment when using wall-functions uses experimentally validated increment of the velocity closest to the wall and therefore the first cell size of the mesh shall be of y^+ larger than 30. When fully resolving a boundary layer the first cell size should be y^+ smaller than 1.

2.2 Infrastructure of the design process at GKN

The CUDA solver in VolVane uses the GPU-cluster due to the profitable computational time. By using the GPU-cluster, more cores can be used simultaneously, which makes the simulations faster. This Section describes the infrastructure of how files in CUDA simulations are transported.

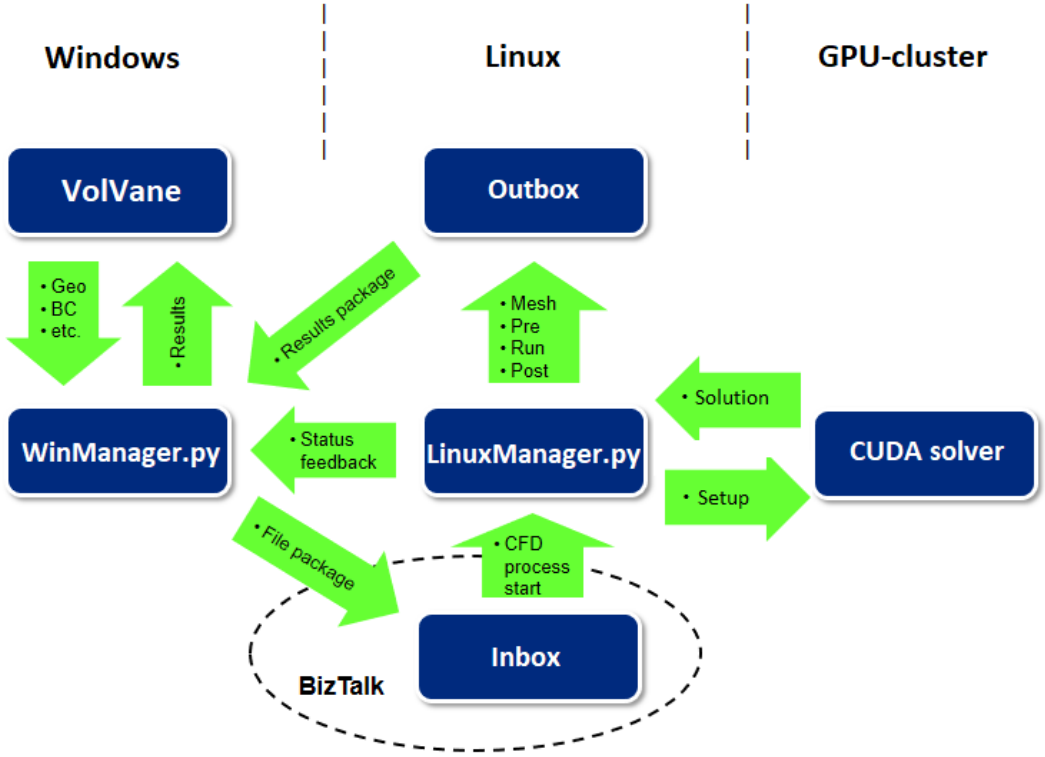


Figure 2 Overview of the CUDA solver infrastructure.

In Figure 2 the infrastructure of the design process using CUDA, between Windows, Linux and the GPU-cluster, is visualized. The communication program BizTalk is used to send files of the simulation. The file path start at the Windows side (left side of the figure) at VolVane. WinManager.py sends the files to the Inbox at the Linux side (middle of the figure). At the Linux side, LinuxManager.py sends the files to the GPU-cluster (right side of the figure) and the solver sends back the solution. LinuxManager.py handles the files and send them to the Outbox. The files get sent back to Windows and WinManager.py handles the files and send them back to VolVane. This infrastructure is needed because VolVane is an Excel-based program and thereby not available on Linux, and the cluster is connected to Linux and not to Windows. The solution for this problem has been to use the program BizTalk until a solution where the cluster can be used directly from Windows is set up.

The flow of files start with the script **WinManager.py**. WinManager.py has to be located in the folder of the VolVane-sheet. The script zips four files from VolVane with settings for the simulation: BCin_GPU.dat (inlet boundary conditions), BCout_GPU.dat (outlet boundary conditions), controlparameters.txt (several different templates for different components are available, the TRS-template is used in this thesis) and volvane_vac_export.dat (geometry file). The zipped files is sent to the Inbox in the Linux domain. In WinManager.py a line for choosing version of post-processing is included. The available versions (PROD, DEV and TEST) have different inboxes.

In the Linux-domain there is a script named **LinuxManager.py** which looks for cases (to send to the custer) in the Inbox. Information is constantly sent to the Windows domain to update a log-window on which process is currently ongoing. When all processes are complete the solution is zipped and moved to the Outbox. This folder is constantly updated and checking for a files named “simulation”.output_complete, which is created when the simulation is complete. The outbox sends the zipped files to the Windows domain, where parts of the solution can be put into VolVane. Most of the solution is presented as numbers in text files or plots.

This flow of files can be initiated in VolVane, and also, if the structure is set up locally with folder and search paths modified in the scripts pointing to the local folders. LinuxManager.py should then be executed with the argument of the zipped files in the inbox.

All the processes in the simulations are automized. The flow of files uses generic scripts for meshing, the C++ code g3dcuda as solver and generic scripts for post-processing. Post-processing is the process that has been investigated most during this thesis. The script **trollsol.py** is used to extract the flow parameters from the solution. Together with the files generated by **volumemesh.dat** (a file generated in the meshing process) the solution is posted. One of the files is **post.trollsol** which can be modified in volumemesh.dat. The file is modified by adding a name of the file, which flow parameter should be posted, name of the posted file, number of blocks and then a line of numbers representing which coordinates of the mesh the flow parameter should be extracted from. The code used is written in g3dmesh code, and the document used for understanding how to modify the post-processing, which is attached in Appendix B. In the post-processing script there is also variables of averaging. There are three types of averaging available: Area, flux and mass, denoted A, FLX and M. The averaging can be to a profile or not, denoted P if a radial profile is wanted. Averaging is denoted AVE, which together makes a line of characters. For example MPAVE gives a mass-averaged profile.

3 Methodology

This Chapter describes the methodology of the thesis. The methodology describes how the work has been carried out. Settings for the simulations, strategy for obtaining results and definitions of useful ratios for analyzing results are presented.

3.1 Simulation settings for CUDA

The settings for the CUDA simulation are set in VolVane. In this thesis, inlet boundary conditions are set as profiles for the tangential spans. For the outlet boundary condition a constant value for the static pressure is set. Meshing, solving for the solution and post-processing data is done by generic scripts, produced by pressing a button in VolVane. This is known as "One-click-CFD", and by using this generic procedure a lot of time is saved in the design process.

The computational domain is visualized in Figure 3. The domain consists of one sector of the TRS with periodic boundary conditions. This implies an assumption of rotational symmetry. This type of domain is used for all TRS designs in this thesis, both for regular and tube vanes.

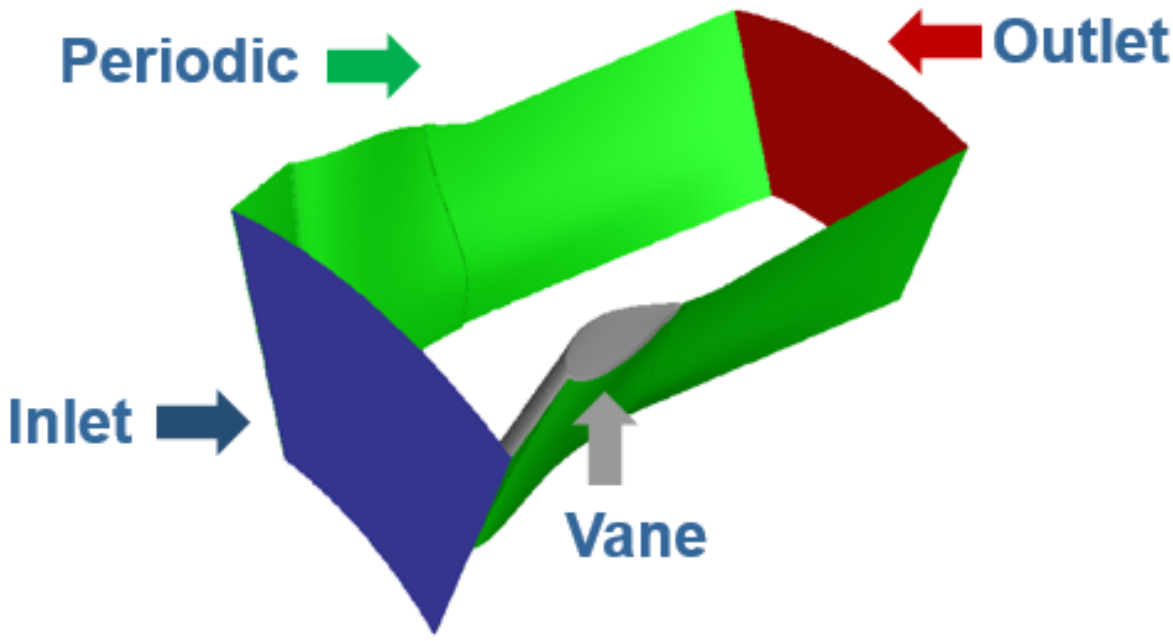


Figure 3 The computational domain of all simulations.

The domain is divided into cells, creating the mesh, to be able to predict the flow in the simulations. The mesh for the CUDA simulations is visualized in Figure 4. In the figure, there is a zoom in on the trailing edge showing that the vane is cut off. The cut off is there to prevent vortex shedding and oscillations in the flow and solution. Otherwise, the mesh is coarse and uses wall-functions as wall treatment. The values for y^+ are about 50 for the wall-function mesh.

For the numeric, the solver uses the k-ε turbulence model with a realizability limiter. For the schemes, 3rd order is used for convective fluxes and 2nd order is used for diffusive fluxes.

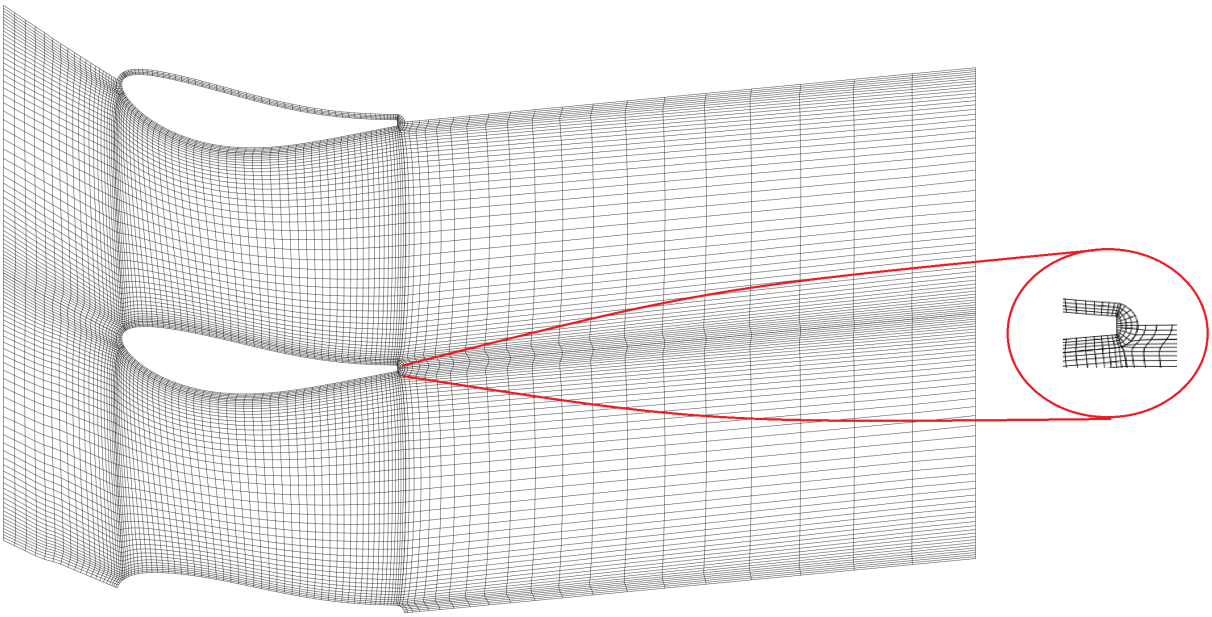


Figure 4 Mesh for CUDA, with a zoom in on the trailing edge.

3.2 Comparison between CUDA versions

One of the purposes of this thesis is to verify the new version of the CUDA solver. The current version is called PROD, which stands for production. The new version which is under development, is called DEV. These two versions are compared to verify if DEV should be implemented as the new PROD. Meaning that the new version should be used when designing TRS at GKN.

3.3 Evaluated flow parameters

From the simulations a lot of flow parameters is extracted to be able to analyze the results. The DEV-version post-process more data than PROD. All data that is posted in PROD is also posted in DEV. The post-processed data which is compared between the versions are shown in the bullet list below.

- Swirl angle (inlet and outlet)
- Mach number (span wise planes, inlet and outlet)
- Total pressure (inlet and outlet)
- Static pressure (span wise planes, vane, inlet and outlet)
- Axial wall shear stress (vane)

Some parameters are presented as profiles and/or averaged values as well. For some of the posted parameters, a non-dimensional representation is to prefer. One of these flow parameters is the total pressure loss, referred to as “pressure loss” in this thesis.

$$Pressure\ loss = \left(\frac{Inlet\ total\ pressure - Outlet\ total\ pressure}{Inlet\ total\ pressure}\right) \qquad (19)$$

The pressure loss is one mass averaged value for the domain.

3.3.1 Separation

Separation is a flow phenomenon where the flow detaches from the geometry. The criteria for the point of separation, in this thesis, is defined as when the axial wall shear stress becomes negative, $\tau_w < 0$. This means that the friction due to the fluid motion on the wall is in the negative direction, which means that the flow recirculating backwards. The axial wall shear stress is evaluated at the vane, where the separation bubble appears on the suction side. There are often some negative values of the axial wall shear stress at other locations of the vane as well, but these are not counted as separation. One of these locations is the trailing edge.

3.3.2 Swirl studies

When analyzing the pressure loss, the inlet swirl angle of the flow has a great impact. The swirl angle is defined as

$$Swirl = \left(\frac{180}{\pi}\right) * \arctan\left(\frac{v_{theta}}{v_x}\right) \quad (20)$$

where v_x and v_{theta} are the axial and tangential components of the fluid velocity. The study when investigating how the pressure loss is dependent of the swirl angle is called swirl study. In this thesis, swirl studies is carried out for four different TRS. The swirl angle which the vane is designed for is called ADP. ADP stands for aerodynamic design point. When running simulations with increased or reduced swirl than the vane is designed for, the simulations are called off-design simulations. When visualizing the pressure losses for different swirl angles, a so called loss bucket is presented. To present this even more clearly, tables which the separation points for different TRS and simulation settings are presented in the results in Chapter 4.

3.3.3 Pressure for upstream forcing

A comparison of the static pressure on the TRS inlet was carried out as well. This was because upstream forcing can be analyzed by knowing the static pressure on the inlet. Upstream forcing predicts how the presence of the vane affects the component upstream of the TRS in the engine. Since this is not included in the standard post-processing this had to be implemented in the post-processing. The tangential variation of the static pressure at the inlet were compared at different span. A coefficient called

$$Pressure\ distortion\ factor = \left(\frac{maximum\ pressure - minimum\ pressure}{average\ pressure}\right) \quad (21)$$

is calculated for each span, where the average pressure was calculated as the arithmetic mean. Abbreviation for this factor is PD-factor.

3.4 Comparison between CUDA and Fluent

For verification of the in-house CUDA solver, a comparison of the simulation results against Fluent, for all TRS, is carried out. For the CUDA solver, the DEV-version is used. This is because the results from the comparison between the versions showed that DEV is ready to be implemented. For the Fluent simulations, several different settings is used. Realizable k-ε is commonly used at GKN due to good agreement with experimental tests, which is why this model is chosen to be used in this thesis as well. For realizable k-ε, both fully resolved boundary layers and wall-functions can be used. Due to that fully resolved boundary layers probably gives more accurate predictions, but wall-function simulations can use the same mesh as the one in

CUDA, both are used. With the results from a minor wall-function study, the non-equilibrium wall-functions is chosen. The k-ω SST model is also chosen, but for this model only the LowRe-mesh is used due to that wall-functions are not available for k-ω SST in Fluent.

3.4.1 Simulation settings for Fluent

The version of Fluent used in this thesis is 18.1. For the meshes used in the simulations, the generic in-house tool Blademeshsetup is used. When running the CUDA solver in VolVane, a text-file called meshparameter.txt is created. This file contain all settings to be able to create the mesh with Blademeshsetup. For the wall-function mesh used in Fluent, identical settings are used as in CUDA and thereby the same meshparameter.txt-file is used. From previous projects, old meshparameter.txt-files are obtained. By modifying the files so just the wall refinement is added, the LowRe-meshes is created. The meshes are created as Patran neural files. In Figure 5 the LowRe-mesh is visualized. The mesh is more refined compared to the wall-function mesh in Figure 4, especially close to the walls to obtain the y^+-values less than one. This value is obtained for the whole domain, unless for some minor areas on the leading edge. The cut off on the trailing edge is not used in this mesh.

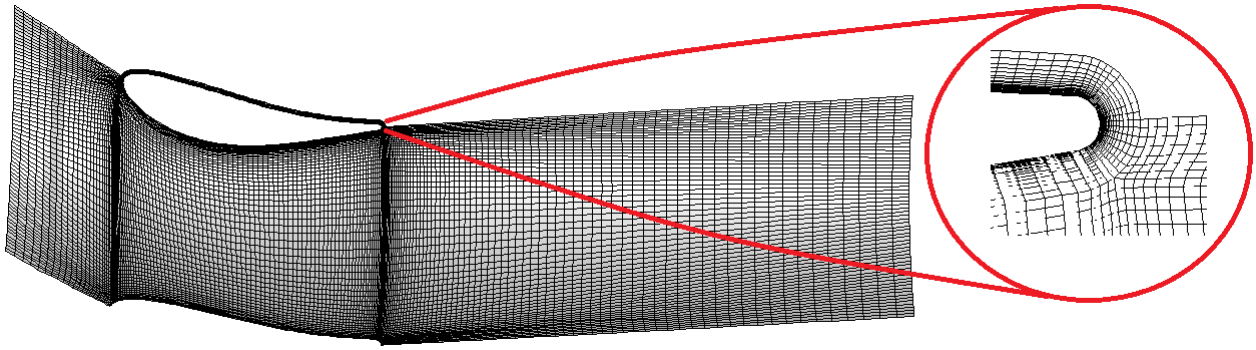


Figure 5 *Mesh for fully resolved boundary layers in Fluent, with a zoom in on the trailing edge.*

The inlet boundary conditions used in Fluent are the same as in CUDA. For the outlet boundary conditions though, is a target mass flow used. The CUDA simulations run for comparison against Fluent, is iterated to match the mass flow obtained in Fluent. Since the target mass flow can be difficult to obtain in Fluent when simulations are close to separation, the outlet static pressure corresponding to the target mass flow in ADP is used for all simulations in the swirl studies. For CUDA, the same static outlet pressure is set for all simulations in the swirl studies in a similar way.

Input scripts for Fluent from previous projects was modified to match the new boundary conditions and for the post-processing scripts, output to match the data generated by Fluent was added.

3.4.2 Pressure loss ratios

In Section 3.3 the flow parameters used for comparing simulations are described. Since the comparison between CUDA and Fluent involves, not just different solver, but also different mesh resolution and turbulence models, two more parameters are evaluated when comparing to Fluent. For comparing loss buckets, two ratios is needed. The loss bucket visualizes the absolute pressure loss between the simulations, but to visualize the shape of the loss bucket these ratios are needed.

1. The first ratio gives a value for a factor that describes the shape of the loss bucket for off-design simulations. The ratio is called "off-design factor" in this thesis.

$$Off - design\ factor = \left(\frac{Loss - ADP\ loss}{ADP\ loss}\right) \tag{22}$$

   Where the pressure loss for the evaluated swirl angle and at the ADP loss uses the same TRS and the same settings (both software and turbulence model).

2. The second ratio is a comparison between a simulation and RKE with fully resolved boundary layers in Fluent. This ratio is called "loss difference" in this thesis, since it describes a difference in loss. The loss difference is defined as

$$Loss\ difference = \left(\frac{Loss - Loss_{Fluent.RKE,LowRe}}{Loss_{Fluent,RKE,LowRe}}\right) \tag{23}$$

   where the normalization loss is the value for the LowRe-mesh simulation in Fluent at the same operating point. This means that this ratio describes a difference between losses for the same swirl angle, but different simulation settings. The software, mesh resolution or turbulence model can differ as settings.

The two ratios are added to the comparison due to that the loss buckets between different simulation settings shows trends when varying the swirl angle. This trends was wanted to describe as a factor. By multiplying a value to the pressure loss for all swirl angles in a CUDA the same value was given as the LowRe simulations with RKE in Fluent. This factor is called "multiplication factor" in this thesis. But, instead of using the pressure loss predicted from the software which is evaluated, RKE with fully resolved boundary layers in Fluent is used for normalizing. This factor is the loss difference.

Due to that CUDA and Fluent predicted different total pressure in the domain, a comparison an study for this is carried out as well. A mass averaged value for cross-sections downstream the domain, starting at the inlet, all the way to the outlet, is compared between the software.

4 Results

In this Chapter the results are presented. Firstly the results between CUDA versions are presented, where the DEV-version is verified to be the new PROD. Then, a comparison between CUDA-DEV and Fluent is presented. Both swirl studies and pressure studies are carried out when comparing CUDA to Fluent. Lastly a summary for the comparison between CUDA and Fluent is presented. TRS1-5 are used as computational domains.

4.1 Verification of the DEV-version

Figure 6 displays the total simulation time for the two CUDA versions. The new version is faster than the old version. DEV is 25% faster and it is the time for post-processing which is shortened, even though more posted data is generated.

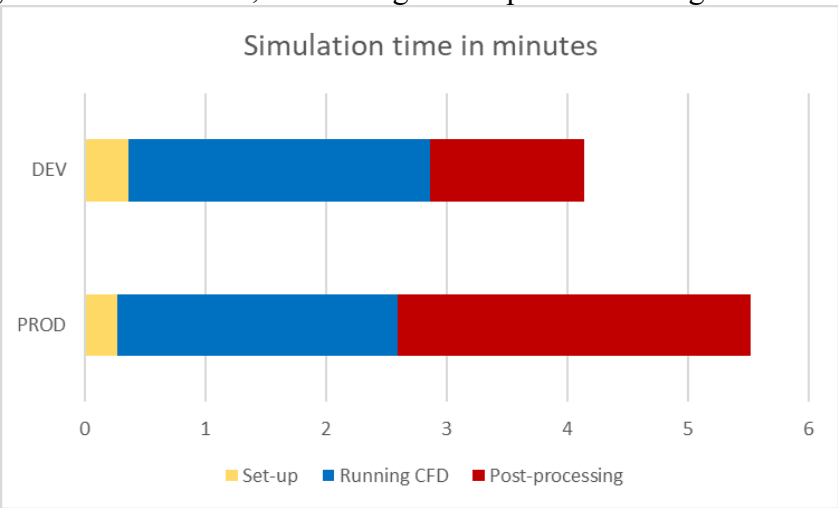


Figure 6 Simulation time for the two CUDA versions.

For the flow parameters and the pressure loss, the differences between versions are less than 0.5% between the versions. In Figure 7 the profile for outlet swirl angle and the static pressure distribution on the vane are visualized, where the results are extremely similar as well.

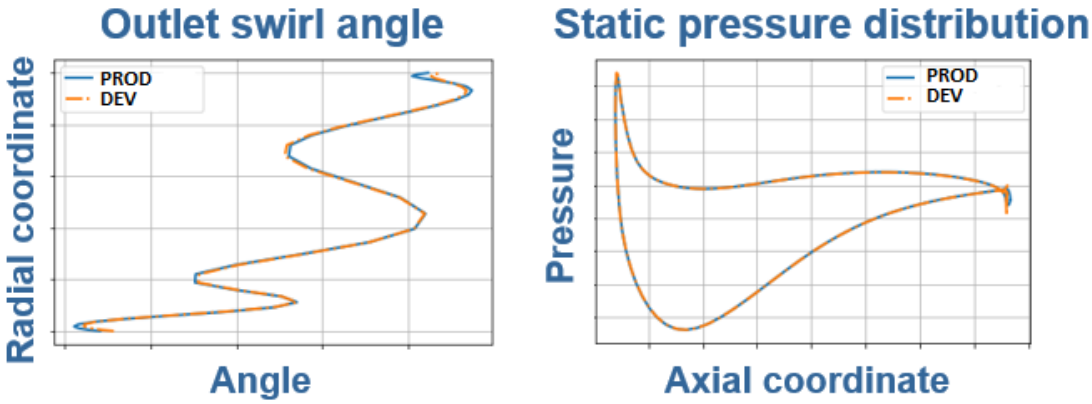


Figure 7 Two of the flow parameters evaluated.

The flow parameters are close to identical, for the posted data except for the axial wall shear stress. In Figure 8, a comparison between the CUDA versions is visualized. To verify which version predicts the correct axial wall shear stress a simulation with LowRe-RKE in Fluent is added. The DEV-version is closer to the Fluent results.

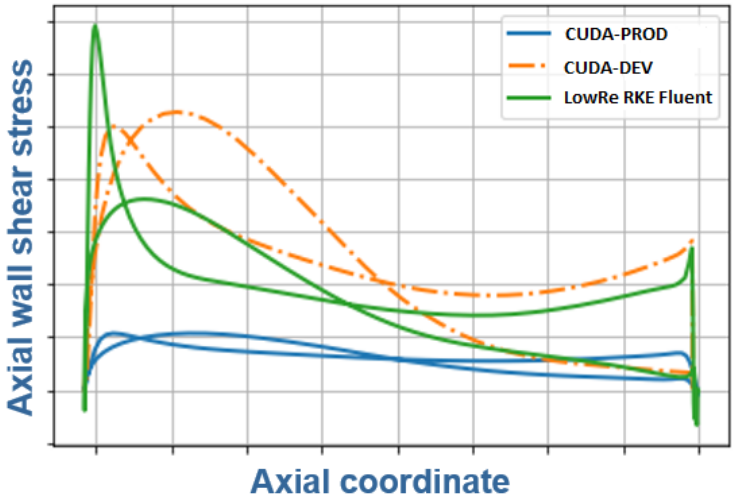


Figure 8 The axial wall shear stresses from simulations of the same TRS, predicted with the two versions of CUDA, and LowRe-RKE in Fluent for verification.

The comparison between versions is carried out with TRS1&2. To summarize the comparison between CUDA versions, the new version is faster, posts more data and is more correct. Therefore this version is now implemented as the new CUDA version for designing TRS at GKN. When comparing CUDA with Fluent in the following Sections, the DEV-version is used as CUDA-PROD. The old CUDA-PROD is renamed to PROD_old.

4.2 Comparison between CUDA and Fluent

The verification of CUDA, when comparing with Fluent, is carried out for TRS1&3-5. This is because TRS1&2 represent two different type of vanes on the same physical TRS, and the results for these vanes are almost identical. See Appendix C for a comparison of the results between TRS1&2. TRS4&5 represent two different type of vanes on the same physical TRS as well, but the results are not identical. For the CUDA-version, the version which is called DEV pervious in this thesis is used.

4.2.1 Flow parameter study

In Figure 9 the outlet swirl angle and the vane static pressure distribution are visualized. The two software predicts similar results for the flow parameters. The difference is larger than between the two CUDA versions.

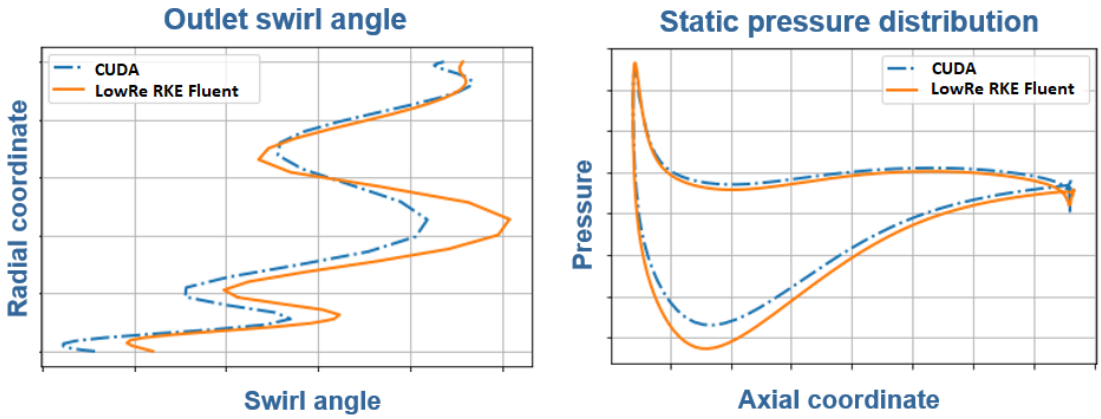


Figure 9 Two flow parameters for CUDA and Fluent.

4.2.2 Swirl studies

The swirl studies predicts how pressure loss is dependent on swirl angle. For the comparison with Fluent a short investigation of different wall-functions in Fluent is carried out, see Appendix A for more details. The non-equilibrium wall-function (non-eq-wf) is chosen in this thesis, due to that the predicted results are similar to LowRe-RKE.

The results are similar for all swirl studies, and therefore TRS1 is presented in this Section and the other swirl studies in Appendix D. In Section 0 a summary of all results are presented.

In **Error! Reference source not found.**the result of the loss bucket is presented for all simulations for TRS1. CUDA predicts lower pressure loss than Fluent, but the shape of the loss bucket for CUDA and LowRe-RKE in Fluent are similar. Therefore, the graph for CUDA multiplied with a factor of 1.15 is added to the figure, which facilitates a comparison between the software since the difference in absolute value is compensated for. This multiplication factor varies between the designs, but this trend is seen through all swirl studies. The other turbulence models in Fluent predicts less similar results in shape of the loss bucket, but the pressure losses are closer in absolute value to LowRe-RKE than what CUDA predicts. For kwSST separation occurs early, which is why only results up to 10 degrees are presented. The separation points for all simulations are presented in Table 1.

Table 1 Separation point for the different simulation settings.

CUDA	Fluent LowRe-RKE	Fluent kwSST	Fluent non-eq-wf
ADP+14	ADP+12	ADP+4	ADP+10

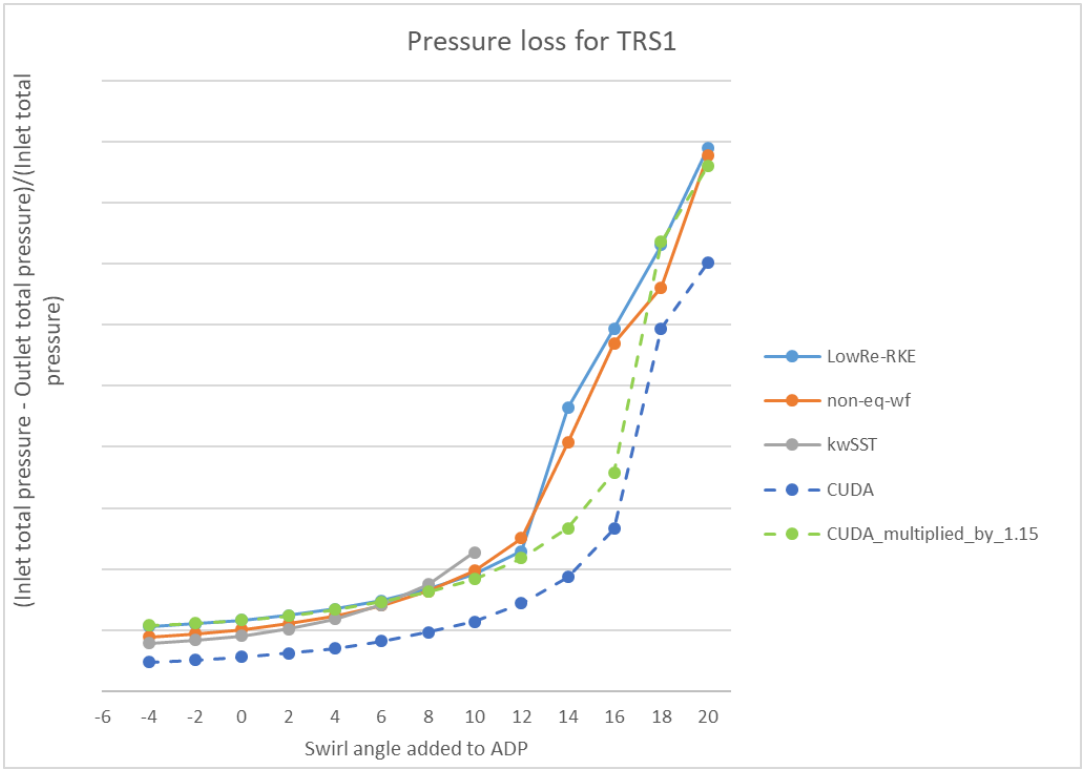


Figure 10 Loss bucket for the swirl study of TRS1.

In Figure 11 the off-design factor is visualized for TRS1. The result for each simulation is normalized (as described in Section **Error! Reference source not found.**) to show the relation between off-design pressure losses compared to the ADP loss. The pressure loss in CUDA differs from the loss in Fluent, but the off-design factor is similar, especially for lower swirl angles. kwSST predicts a much steeper shape than the RKE-simulations. For non-eq-wf, which was chosen due to similar results for LowRe-RKE, the shape is steeper as well. When separation occurs, the off-design factor increases rapidly, which is why the graphs are zoomed in on the value for non-separated simulations.

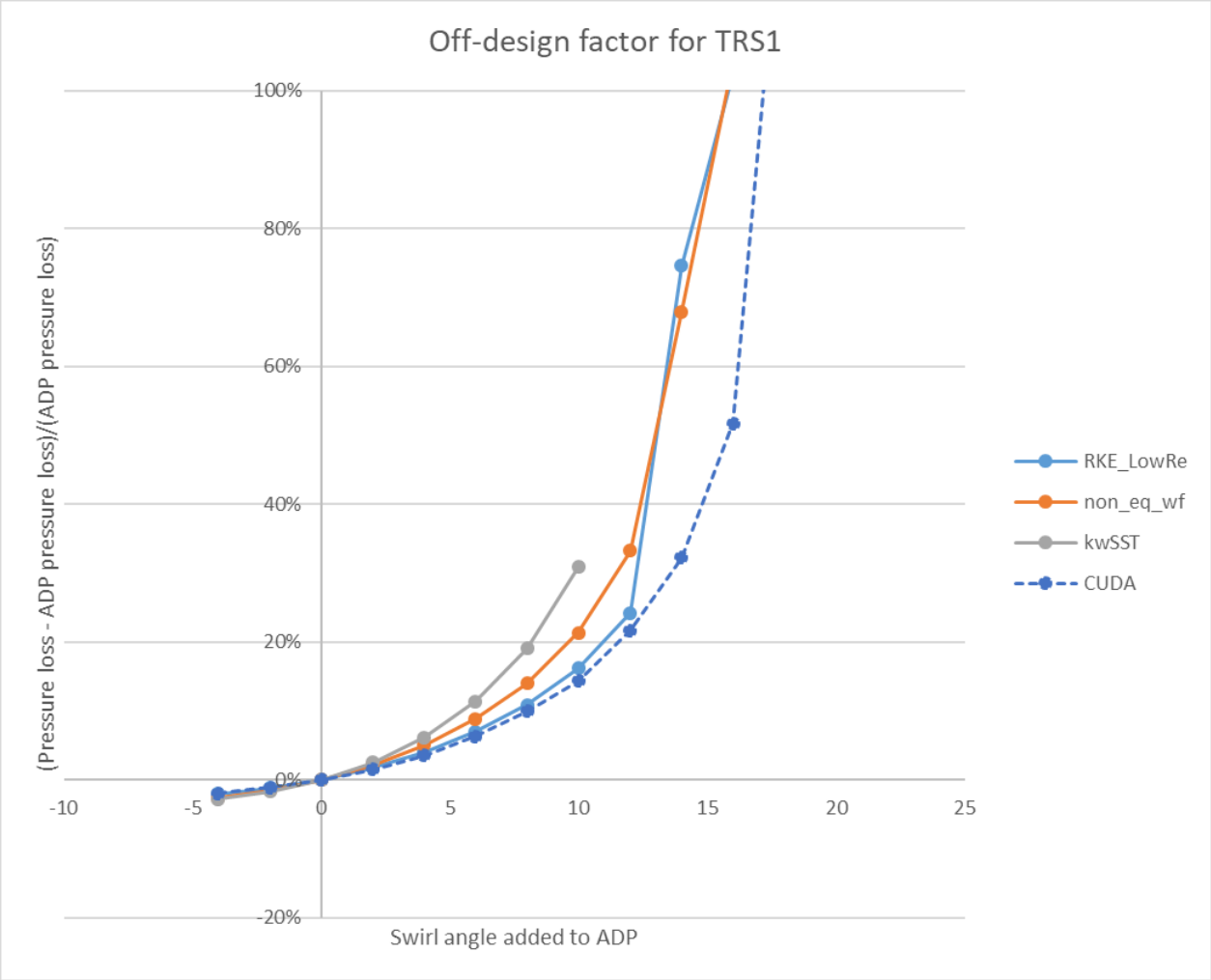


Figure 11 *Off-design factor for the swirl study of TRS1. The graphs are cut off to zoom in on the values for non-separated simulations.*

In Figure 12 the loss difference is visualized, which gives a relation between LowRe-RKE and the evaluated simulation. The Fluent simulations predict pressure losses which are close to the pressure losses in the LowRe-RKE simulations, but CUDA capture the trend more accurate. The lines for CUDA are almost constant with an offset of -13%. This loss difference corresponds to the multiplication factor, but determined from a reference level at the Fluent pressure loss instead of the CUDA results, which makes it negative and the value becomes smaller. The graphs are zoomed in on the non-separated simulations. When LowRe-RKE in Fluent separates earlier, the graph decreases rapidly and vice versa if the evaluated simulation separates earlier than LowRe in Fluent.

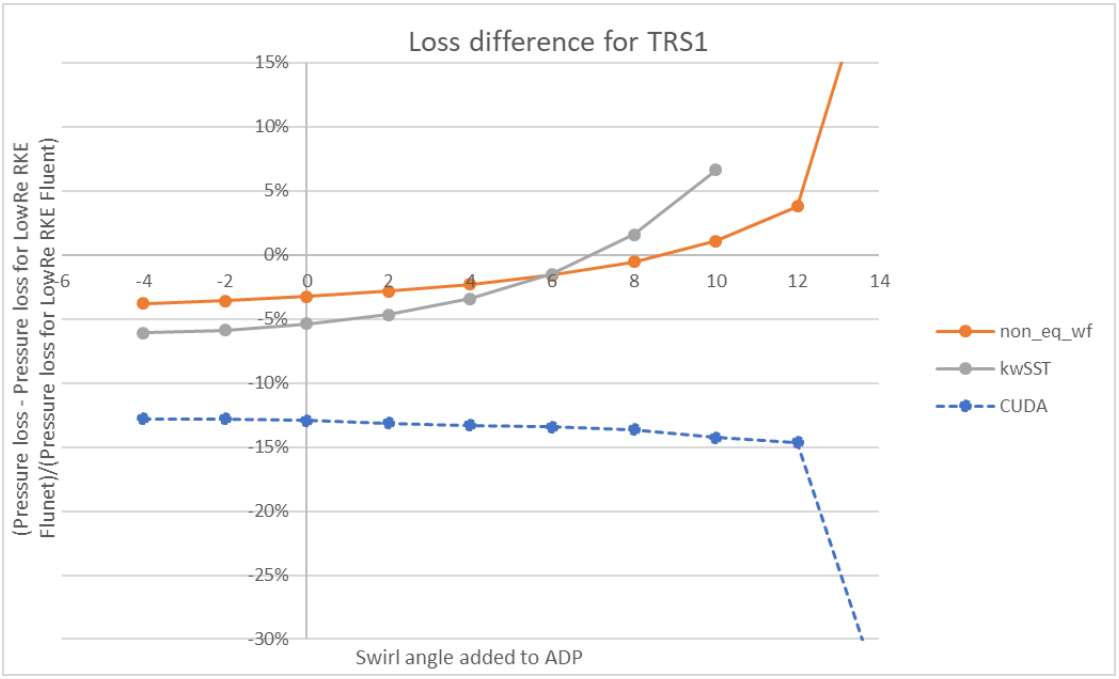


Figure 12 *Loss difference for the swirl study of TRS1. The graphs are cut off to zoom in on the values for non-separated simulations.*

4.2.3 Upstream forcing

For the pressure distortion factor (defined in Section 3.3.3) the different software give similar results, as seen in Figure 13. The figure visualizes the PD-factor for TRS4&5 using both CUDA-DEV and Fluent LowRe-RKE. Fluent generally shows less variation between spans and the predicted pressure distortion is generally higher. The largest difference between the software is for the middle span of the domain. Note that all values for the pressure distortion factor is small for both the software. In Appendix E, there is an additional investigation of static pressure at the inlet.

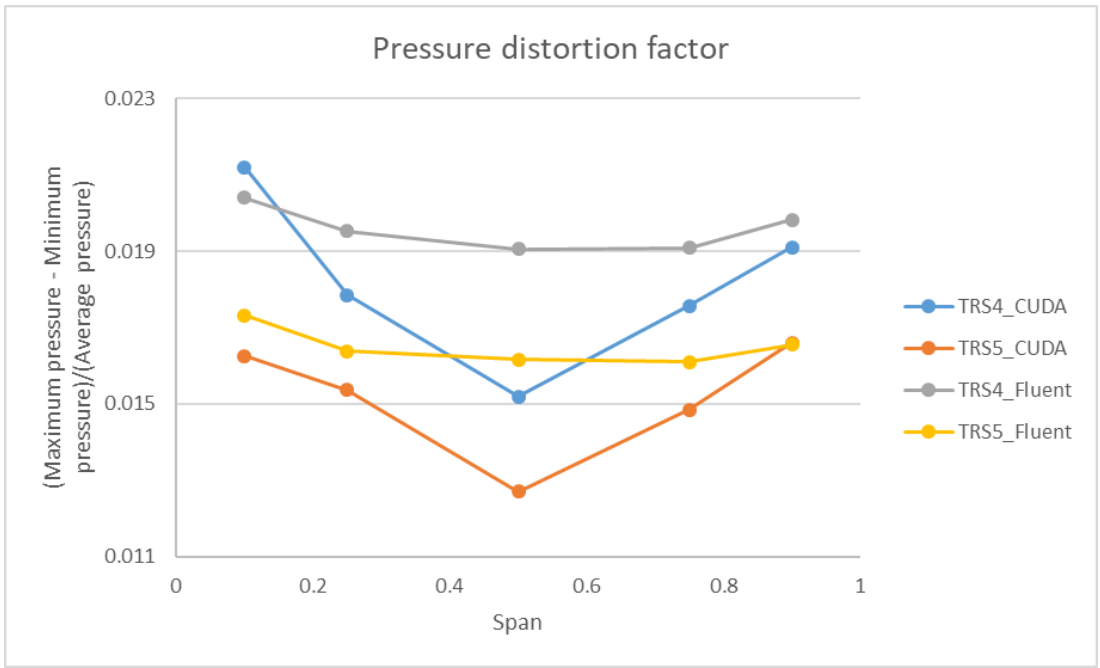


Figure 13 *Pressure distortion factor for TRS4&5, predicted by CUDA and LowRe-RKE in Fluent.*

4.2.4 Total pressure in the domain

The absolute value of the results for total pressure in the domain do not agree between CUDA and Fluent LowRe-RKE, even though most of the trends are similar. In Figure 14 the total pressure in Pascal is visualized, with values showing the absolute difference between the software. The boundary conditions are identical for both simulations, which makes this an interesting result to analyze and discuss further, which is done in Chapter 5&6. This is a known problem though, that the CUDA solver predicts incorrect absolute total pressure values in the domain. The specified total pressure at the inlet will not be set correctly as a boundary condition. For the absolute value, the CUDA simulations predicts lower total pressure than Fluent. And the difference in total pressure between the software is higher closer to the inlet.

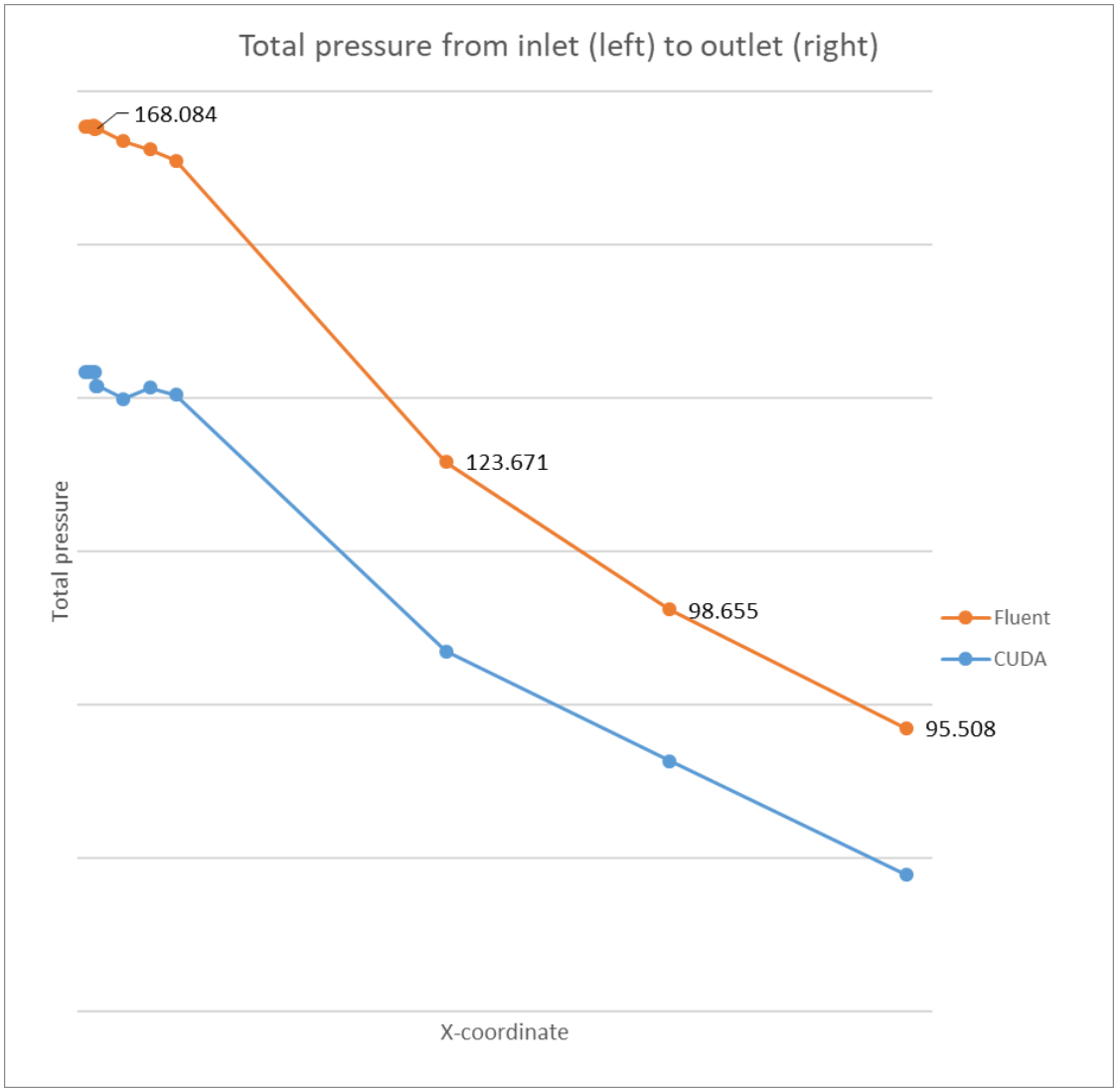


Figure 14 Total pressure through the domain for both CUDA and Fluent. The numbers represent how many Pascal higher the Fluent prediction is.

4.2.5 Summary of the comparison between CUDA and Fluent

In the comparison between CUDA and Fluent, the CUDA solver show great similarities with the LowRe-RKE simulations. The off-design factor for CUDA is even more similar to LowRe-RKE than kwSST and non-eq-wf. This is visualized in Appendix D, where comparisons with the results between CUDA and Fluent are attached. In Figure 15 the off-design factor for all TRS are visualized, where similar values are predicted for all TRS. The off-design factor for TRS4&5 are identical for lower swirl angles. The off-design factor is presented for the swirl angles before separation, for higher values the graphs increases rapidly.

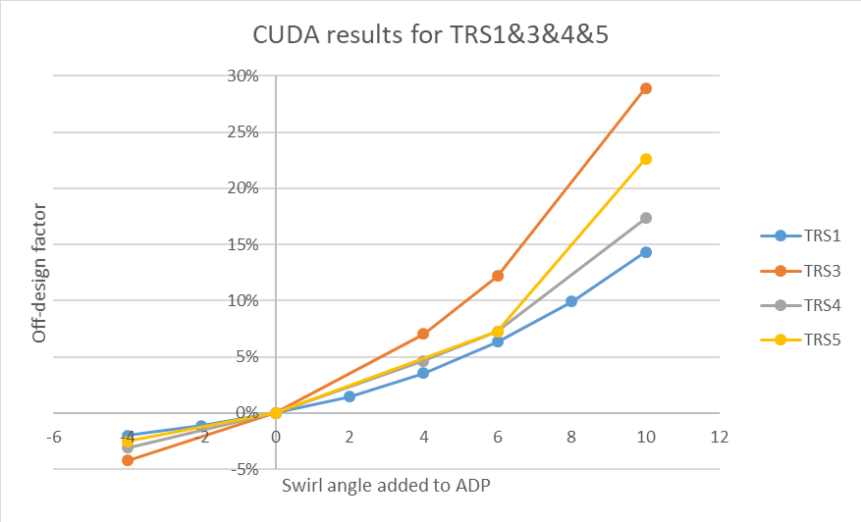


Figure 15 Off-design factor for all CUDA simulations in the TRS swirl studies. When separation occurs, at higher swirl angles, the factor increases rapidly.

In Table 2 the separation points for all simulations are presented. CUDA predicts separation later than Fluent at almost the same swirl angle for all TRS. kwSST predicts very early separation compared to the other simulations. Non-eq-wf predicts less consistent separation points than the other simulations.

Table 2 Separation point for all simulations. The values represent for which swirl angle, added to ADP, separation occurs.

	CUDA	LowRe-RKE	kwSST	Non-eq-wf
TRS1	14	12	4	10
TRS3	14	8	4	6
TRS4	14	10	0	14
TRS5	12	8	0	8
Average	13.5	9.5	2	9.5

In Figure 16 the loss differences for all simulations are visualized. For all designs, the loss difference is fairly constant until separation occurs in the swirl study. The approximation to a constant value of the loss difference, until separation occurs, is -13, -8, -5 and -1 percent for TRS1&3&4&5. Note that the loss difference is constant until separation for the Fluent simulations, then the ratio suddenly drops rapidly. The result between CUDA and Fluent LowRe-RKE shows almost identical trends for all designs, the absolute value of the pressure loss is not identical though. Detailed results from all simulations can be found in Appendix D.

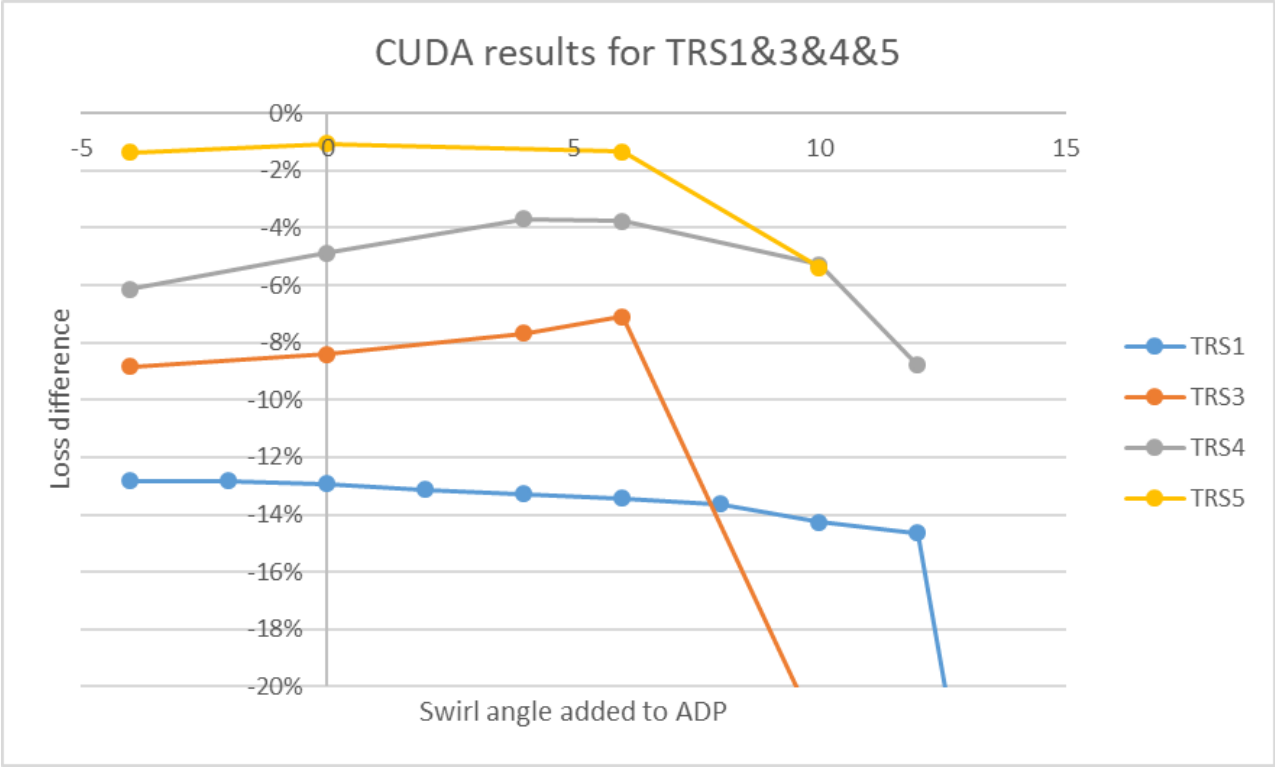


Figure 16 Loss difference for all CUDA simulations in the TRS swirl studies. The graphs are cut off to zoom in on the values for non-separated simulations.

In Table 3, the non-separated results for loss difference for all the CUDA simulations are shown. To be able to compare CUDA simulations to Fluent, a constant value of the loss difference and a consistent offset of the separation point facilitates the conversion between the software. For the loss difference the values are fairly constant until separation occurs, the values are different for all TRS though. When it comes to separation points, the offset values are constant around +4 degrees compared to LowRe-RKE and +11.5 degrees for kwSST.

Table 3 Summary of the results from the swirl studies.

	Loss difference as a approximated value for non-separated design points	Separation point for CUDA relative Fluent LowRe-RKE	Separation point for CUDA relative Fluent kwSST
TRS1	-13%	+2	+10
TRS3	-8%	+6	+10
TRS4	-5%	+4	+14
TRS5	-1%	+4	+12
Average	-6.75%	+4	+11.5

5 Analysis

This Chapter analyzes and compare all results. Discussion is left for the next Chapter.

5.1 Comparison between CUDA versions

The new version of CUDA is both faster, post-process more data and is more accurate for the axial wall shear stress. This axial wall shear stress now predicts higher values than Fluent LowRe-RKE, which is expected due to that CUDA uses wall-functions. This version is already implemented due to all advantages.

When it comes to the solver, it is the same for both versions. But, where in the solution the result is extracted from is different. Result can be extracted in node values or interpolated to the cell faces, in the new version of CUDA the extraction is the same as in Fluent, which is in the cell faces.

5.2 Analysis of the swirl studies

There are several trends that are similar between the software for all the TRS, both in terms of the loss bucket and the normalized ratios. Fluent gives a higher pressure loss than CUDA. One explanation for this can be that the total pressure differs between the solvers even though the boundary conditions are the same, see Figure 14. A small study where pressure was added to the inlet profile in CUDA was carried out to see if the results would become more similar. By adding total pressure, the inlet profiles became almost identical in the solution at the inlet, but the outlet profile differed. Hence, this is not the reason for the difference in pressure loss, this problem is more connected to the things discussed in Section 5.5. This indicates that the models and solvers are different in the two software. In the Fluent simulations, LowRe-RKE predicts the same shape of the loss bucket as CUDA does, this is good results because the in-house code should give similar results as this reliable simulation setting. For the non-eq-wf the results varies a lot compared to LowRe-RKE for different TRS, which makes it less reliable. For kwSST the predictions always show low pressure loss for lower swirl angles, but rises quickly due to early separation. When it comes to the interval of the absolute value for pressure losses between all TRS, the differences can be a reason for off-design factors and loss differences not being identical. Some of the TRS have higher pressure loss than the others, which makes a comparison between designs more difficult. The TRS that predicts less similar trends, in terms of shape and absolute value of off-design factor and loss difference, compared to the other is TRS4.

For the off-design factor kwSST rises quickly for increased swirl angles due to early separation, while the RKE-simulations are more similar. CUDA and LowRe-RKE are almost identical for most of the TRS until separation occurs. For the off-design factor, non-eq-wf predicts varying results due to the varying accuracy of the loss bucket, but this factor is still fairly similar to LowRe-RKE for all TRS. For the off-design factor of the different CUDA simulations the results are similar. TRS4&5 are almost identical for the swirl angles close to ADP and this shows robustness in the solver.

The loss difference for all the CUDA simulations are fairly constant. Since the normalization is done compared to LowRe-RKE in Fluent, this result shows that the solvers predict similar trends of the flow. The loss differences are of different values between the TRS, but the difference within each TRS is still constant. The difference is comparable to the multiplication factor in the loss bucket, the only difference is which of the simulations (CUDA or LowRe-RKE) that is used as reference level. Even for the loss difference, TRS4 shows less accuracy in the trends as seen in Appendix D. The trend is still more similar to LowRe-RKE than the other turbulence models in Fluent. Non-eq-wf varies a lot between the TRS, but the offset value is closer to zero than for CUDA. For kwSST the loss value is not comparable to LowRe-RKE, the early separation makes it difficult to find a trend.

5.3 Separation in the simulations

The pressure loss rises quickly in the loss bucket when separation occurs, and in all the simulations separation occurs at different swirl angles. The comparison between simulations is interesting for swirl angles slightly before ADP, and up to the point where separation occurs. This is both because the flight conditions are in this region and the simulations behave similarly. Trends and comparisons get more difficult to analyze close to separation, and this is why most of the trends are investigated for lower swirl angles. The definition of separation is set to when $\tau_w < 0$, which is a simple criteria. There are however, more advanced criterias for the definition of separation which were investigated a little in the project. The separation criteria was only touch upon though, due to the short period of time of the project.

By only investigating the separation point of the simulations, minor analysis can be carried out. CUDA predicts that separation occurs at ADP+14 degrees for all TRS, except for TRS5 where the prediction gives ADP+12 degrees. The interval of separation points predicted is not large for the other simulations either, but at least it varies more than for CUDA. The simulation which varies most is, as before, non-eq-wf. CUDA predicts higher values of separation point than all Fluent simulations. kwSST predicts, by far, the lowest values. The differences between CUDA and the LowRe-simulations are in average 4 degrees for RKE and 11.5 degrees for kwSST.

5.4 Upstream forcing

For the pressure distortion factor some trends are identified. CUDA predicts lower value, and Fluent predicts more constant value for this factor between different spans at the same TRS. But, when comparing losses between TRS the difference in PD-factor is almost identical. As seen in Figure 13, where two vanes for the same TRS is analyzed with both CUDA and Fluent, the difference in PD-factor is the same even though the shape and magnitude differs. One thing to add is that the magnitude of the PD-factor is very small for the TRS and the values are almost identical. Due to the small differences it is difficult to tell any difference between software in the predicted upstream forcing.

5.5 Pressure in the domain

For both CUDA and Fluent the inlet total pressure is set as a boundary condition, and the same profile is used in both software. In CUDA this profile is not determined to be the solution though, which makes a difference in the simulations. The difference probably occurs due to that the vane is located close to the inlet and a linear interpolation is used to set the value at the inlet. This causes a variation of total pressure in tangential direction. This variation is not visible in simulations from Fluent.

The static outlet pressure differs as well, since this pressure is set to match the mass flow. This parameter also differs less, between BC and solution, in Fluent than in CUDA. The pressure is solved for in the governing equations and this difference show that Fluent is more reliable than CUDA, but CUDA still have the advantage of faster simulation and built in post-processing. Looking at the trend for the total pressure in the domain in Figure 14, the trend is almost identical for the solvers. As for pressure losses, there are similar trends shown for both solvers.

6 Discussion

In this Chapter the analysis is discussed. This Chapter discusses the value of the results and how to use the results.

6.1 New versions of software

One of the outcomes of this thesis is that DEV will be the new PROD version of CUDA. Verification of this version is a work that had been wanted for a long time, since the new version is more correct, faster and post-process more data. The DEV version will save simulation time and hopefully the designs will be even better due to the corrections in the post-processing. Within this thesis there are some new ideas for posting with CUDA, but the process of implementing these changes has not been started yet. See Chapter 8 for details of suggested improvements.

6.2 How to use results of the swirl studies

As results of the swirl studies three graphs, a loss bucket and two normalized ratios, could be plotted to visualize the trends of the simulations. The trends for the different software and settings are interesting to investigate. It can be good to have the trends in mind when running simulations in the future. One clear trend is that CUDA predicts lower results for the pressure loss than Fluent, this has to be compensated for when designing new TRS in VolVane. Since LowRe-RKE in Fluent is robust, it is always good to run a simulation, at a non-separated swirl angle, with that setting and compare with the CUDA results. But, since the results of the two ratios agree well with the Fluent results, only the multiplication factor or the loss difference is needed to describe the difference between the software. If the off-design factor is identical between software, only a multiplication factor or loss difference is needed to be able to compare the software. The multiplication factor and loss difference relates CUDA to Fluent LowRe-RKE, and even though this constants are different for all TRS only one simulation is needed to get the relation. This might be used to run more simulations in CUDA and less in Fluent, thereby saving simulation time in the design process.

6.3 Separation

The separation point differs for different settings, both for different turbulence models and for the different software. kwSST predicts separation at lower swirl angles than RKE. This is an expected result due to that an approximation more prone to separate is used close to the wall. Even the results that CUDA predicts separation at higher swirl angles than Fluent is expected due to that wall-functions tend to predict higher shear stresses than resolved simulations [7]. This thesis show that there is an offset between the separation point between software and this can be good to have in mind when designing new TRS. The offset is determined with separation points based on τ_w, and for more reliable studies separation criterias can be used. A minor study for this showed that it should be possible to include a separation criteria in the post-processing. But for the Truckenbrodt criteria, which was investigated, the velocity distribution on the vane has to be extracted as well. This was not done within this thesis.

6.4 Upstream forcing

For the pressure in the domain, the static pressure shows better correlation than the total pressure, when comparing CUDA and Fluent. The pressure distortion factor shows similar trends for the software, and especially for the difference between TRS when comparing predictions of the same software. Therefore the upstream forcing can be evaluated in CUDA when comparing two designs. The difference between the designs will be similar to what Fluent would predict. The absolute value of the PD-factor is not recommended to use though, since the solver seems to be sensitive towards the blockage of the domain. This is mentioned in Section 5.5.

6.5 Pressure in the domain

When it comes to the total pressure in the domain for CUDA, the source code has to be modified. This have been investigated before, but was not prioritized to change. The investigation showed that the tangential variation on the inlet together with the linear interpolation give rise to an incorrect pressure loss. A change in the scheme between cells or an extended domain before the vane is needed. This will give a more accurate solution at the inlet. That the inlet total pressure in the solution is not the same as defined in the boundary conditions directly affects the pressure loss. This give rise to the difference compared to LowRe-RKE, since this is not shown in Fluent. With the same inlet total pressure in the solution as in the boundary condition, the solution may be very similar to the result of LowRe-RKE.

The small study carried out about adding total pressure to the inlet profile, mentioned in Section 5.2&5.5, showed that it does not solve all problems though. But running simulations with correct boundary conditions and that the solvers give similar trends, as shown with the loss bucket and ratios, will be enough to get even better results in CUDA. The results would be even closer to the results of LowRe-RKE in Fluent.

7 Conclusions

The conclusions that can be drawn from this thesis have been mentioned in previous Chapters. As a summary, this Chapter presents a bullet point list of the most important conclusions. The conclusions are related to the questions asked in the problem description, see Section 1.3.

How does the infrastructure of the simulation process work?

- BizTalk is used as a communicational program, see Section 2.2 for details.

How does the code for the post-process work?

- Trollsol includes post-processing which is described in Section 2.2 and Appendix B. The settings are edited in the file volvane_vac_export.dat.

Is the new version of the solver ready to be implemented?

- The new version (DEV) is already implemented (as PROD) due to this thesis. The solver for the two versions are the same, but the post-processing is more correct and faster in the new version. The new version also generates more post-processed data. The old version is now named PROD_old.

Are the results comparable between CUDA and Fluent?

- CUDA gives similar results as LowRe-RKE in Fluent, even more similar than other turbulence models within Fluent for the majority of the investigated parameters. This is not the case for simulations with large separation though.
- CUDA predicts lower pressure loss than Fluent. The off-design factor is almost identical for non-separated cases, which makes a constant multiplication factor / loss difference the only conversion needed between the software. In average the loss difference for CUDA is -6.75%, where the values for the different TRS analyzes varies from -13% to -1%.
- CUDA predicts separation at higher swirl angles than Fluent, in average 4 degrees later than for LowRe-RKE and 11.5 degrees later than kwSST. The range for how much later is 2 to 6 degrees for LowRe-RKE and 10 to 14 degrees for kwSST.
- For static pressure in the domain, the pressure distortion factor and thereby the upstream forcing agrees well between the software when comparing designs within the software. The absolute values are of the same order of magnitude, but the trends between different spans done not agree.

Can CUDA be used in the design process of TRS?

- Over all the results of CUDA are as reliable, in terms of stability, as the results for LowRe-RKE in Fluent. Fluent is still needed in the design process though, to predict the results with accuracy. For trends of the results CUDA can be used in the design process.

Are there some limitations of CUDA that needs to take into account when designing a TRS?

- The inlet total pressure does not agree with the boundary conditions in CUDA. For the total pressure at the inlet, the scheme or domain have to be changed in the CUDA solver. This has to be corrected in the source code or the meshing. But relations, such as the off-design factor and loss difference it is enough for now to compensate the difference in total pressure at the inlet.

8 Further work

This Chapter present the suggested areas for further work.

- **PD-factor in the post-processing:**
  Analysis of the PD-factor have been done in this thesis, but the implementation of extracting these results in the post-processing still has to be done. Both the extraction of pressure at tangential span and the determination of the PD-factor would be good to implement. This would make the post-processing more generic.

- **Separation criteria:**
  Separation criterias would be very useful both for knowing were separation occurs, but also to get a hint of how close to separation the simulation is. The work of investigating a separation criteria have been started during this thesis work, but further work is needed [8,9].

- **Improve the implementation of the CUDA solver in VolVane:**
  Development of a new version of VolVane is needed due to several problems in the Excel-file connected to CUDA. New description of cells in the Excel-file have to be done to explain the settings. The CUDA solver shares section with another solver called Euler, which solves for the potential flow in the domain. Euler and CUDA need different sections in the Excel-file, the settings needs to be distinguished. The graphical presentation in CUDA does not use the solution, instead it uses the boundary conditions, and the values are not identical as pointed out in this thesis.

- **Numerical problem with inlet total pressure:**
  Since the boundary conditions and solution is not identical, a correction of the scheme or the domain has to be done. This is not included in the scope of the thesis, but the results show that this is an investigation that is important to carry out.

- **Improve infrastructure:**
  For the infrastructure of the solver, the IT department have the ambition to make it possible to send jobs to the cluster from Windows.

9 References

[1] H. Abrahamsson, *Aerospace engine systems_16_9.pptx.* GKN aerospace, 2018.

[2] MakeTechEasier, *MTE explains: The difference between a CPU and a GPU.* [Online]. Available: https://www.maketecheasier.com/difference-between-cpu-and-gpu/ [Accessed: July 31, 2018].

[3] L. Davidson, *Fluid mechanics, turbulent flow and turbulence modeling.* Chalmers University of Technology, Gothenburg, Sweden, 2018.

[4] CFD online, *Navier-Stokes equations.* [Online]. Available: https://www.cfd-online.com/Wiki/Navier-Stokes_equations [Accessed: July 31, 2018].

[5] T.-H. Shih, W.W. Liou, A. Shabbir, Z. Yang and J.Zhu, *A new k-ε eddy viscosity model for high Reynolds number turbulent flows-model development and validation.* NASA, Lewis research center, Cleveland, 1994.

[6] F.R. Menter, *Improved two-equation k-ω turbulence models for aerodynamic flows.* NASA, Ames research center, Moffett field, 1992.

[7] V. Dippold, *Investigation of wall-function and turbulence model performance within the wind code.* NASA, Glenn research center, Cleveland, 2005.

[8] H. Schlichting, *Boundary-layer theory*. McGraw-Hill book company, 1960.

[9] F. Wallin, *Design and analysis of aggressive intermediate ducts*. Volvo aero corporation, Trollhättan, Sweden and Luleå University of Technology, Luleå, Sweden, 2003.

[10] A. Bakker. Class lecture, Topic: “Boundary layers and separation”. Fluent Inc, 2002. [Online]. Available: http://www.bakker.org/dartmouth06/engs150/11-bl.pdf [Accessed: July 31, 2018].

[11] GKN aerospace, *CFD-kurs nätgenerering*.

[12] GKN aerospace, *VOLS10181864_part000_ver01_Design_Meshing_For_TRS.*

10 Appendix A

The small wall-function study for TRS1.

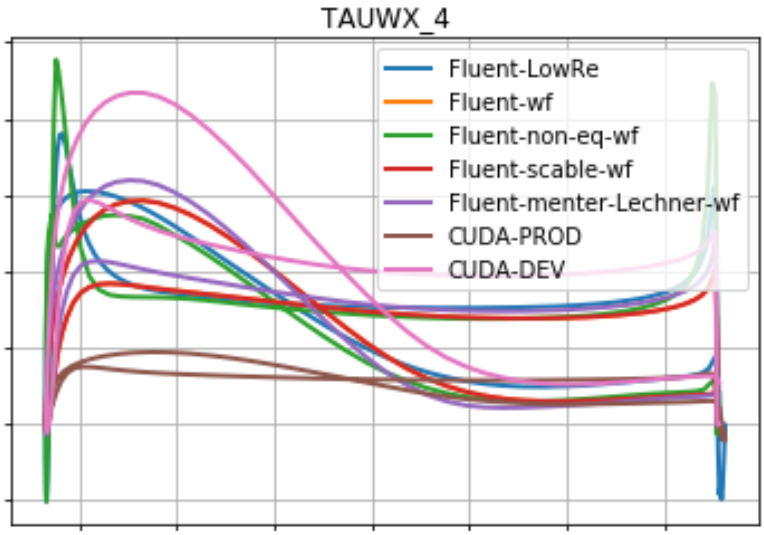


Figure 17 Wall shear stress in axial direction on the y-axis. The results are extracted from span 4, which corresponds to span 0.25. The x-axis represents axial coordinate.

In Figure 17 the results from all investigated wall-functions and also LowRe-RKE in Fluent are visualized. All available wall-functions in Fluent and also CUDA is used in this study. The study shows that the non-equilibrium wall-functions predicts result most similar to the resolved RKE simulations. Non-eq-wf are also known to account for pressure gradient effects and thereby predicting separation more accurate [10]. Because of this, the non-equilibrium wall-functions is chosen.

11 Appendix B

The previous documentation for how the g3dmesh code is written.

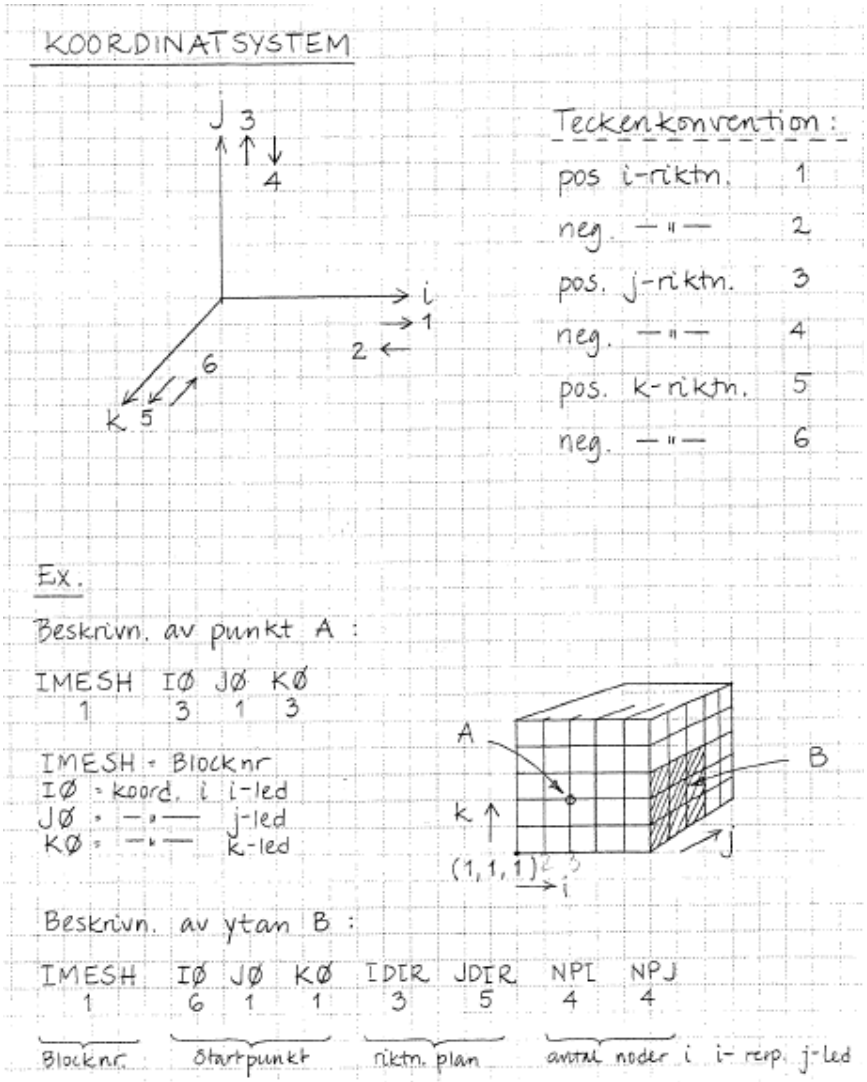


Figure 18 Description of how the g3dmesh code is written. The description is written in Swedish. Figure from [11].

As seen in Figure 18 an example of a row of g3dmesh code are a sequence of eight numbers (1 6 1 1 3 5 4 4). The convention of numbering is numbers from 1 to 6, where positive direction alternates with negative direction. Positive i-direction is denoted with the number 1 and negative i-direction is denoted 2. j- and k-direction follows the same pattern. When describing which nodes in the mesh the post-processing involves, the first number describes the block. The blocks are numbered as in Figure 19. The three following numbers in the post-processing describes the start point of the first mesh coordinate, following the description of directions described in Figure 20. The two following numbers describes the two directions of the plane, which the flow parameters should be extracted from. The last two numbers tell how many nodes should be extracted in the two directions which the plane creates. There can be one more number in the post-processing, describing a rotational coordinate. This number is used when an axial contour plot is extracted.

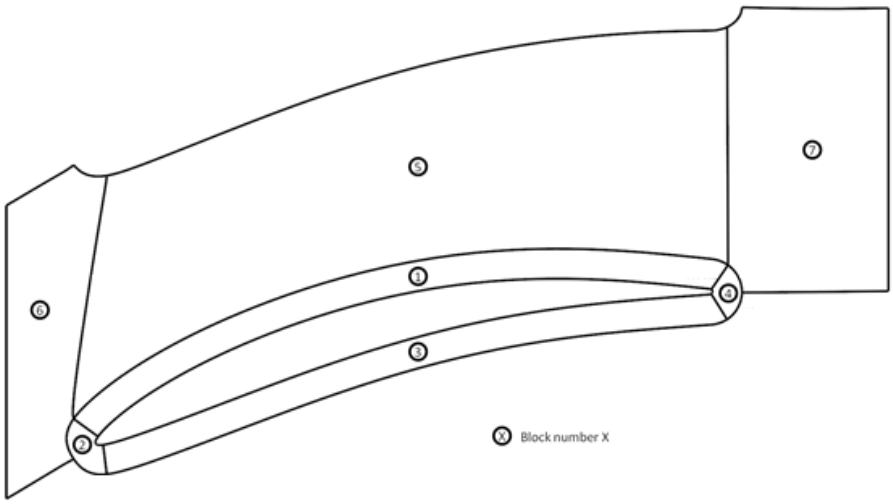


Figure 19 Numbering of blocks in a generic mesh. Figure from [12].

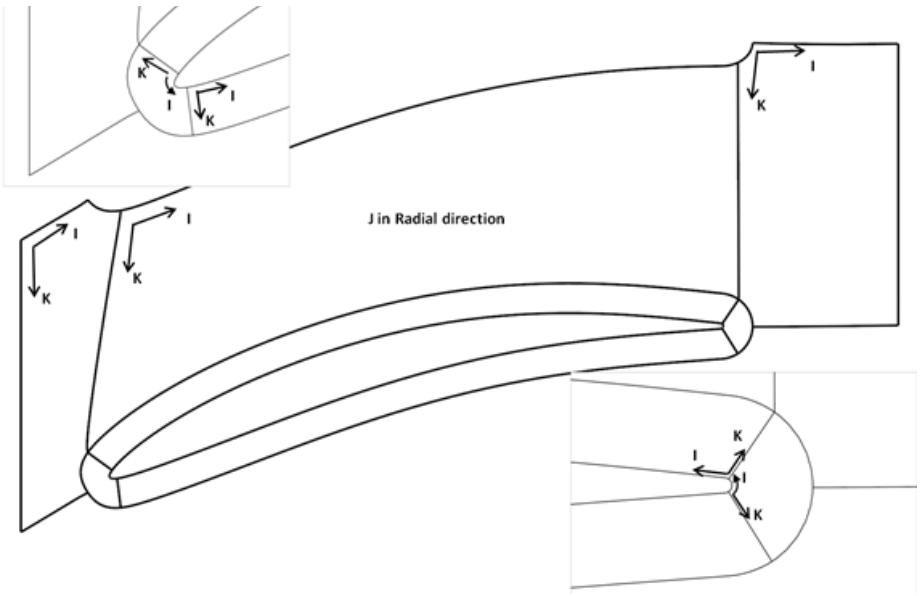


Figure 20 Directions of the mesh coordinates, in the different blocks. Figure from [12].

One thing to note is that the volumemesh.dat file is general for all simulations, in this script all parameters are expressed by generic variables. For example, the first node is denoted as minimum of the number of nodes. By keeping this notation in the general scripts, the scripts will work for the generic meshing.

12 Appendix C

Some results from TRS1&2 to show that they predict similar flow through the domain and therefore only one of the designs (TRS1) is chosen to be analyzed on the swirl studies.

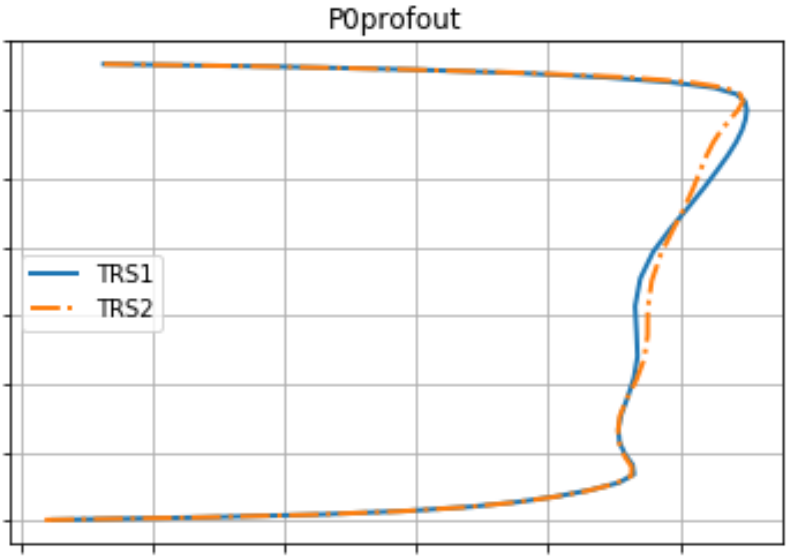


Figure 21 Total outlet pressure for TRS1&2 on the x-axis and span wise coordinate on the y-axis.

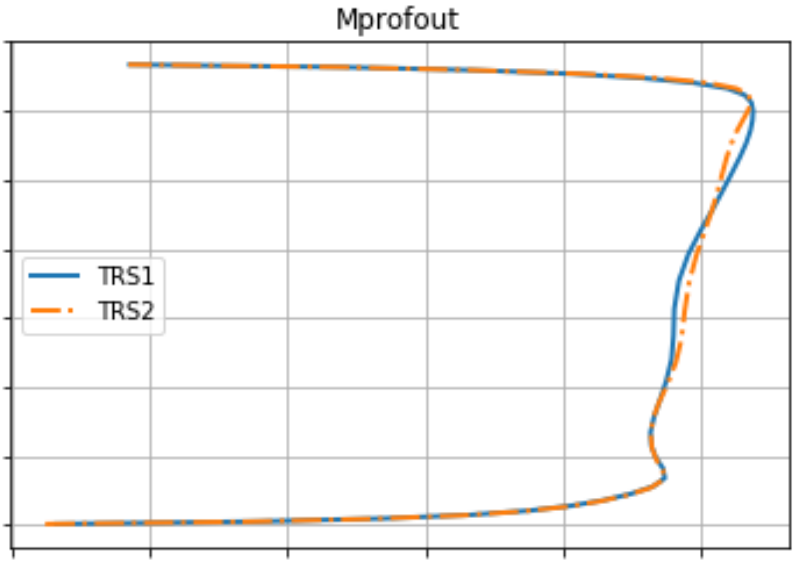


Figure 22 Outlet Mach number for TRS1&2 on the x-axis and span wise coordinate on the y-axis.

Both in Figure 21 and Figure 22 the results for TRS1&2 are identical for lower span and similar for higher span. This is because the inlet boundary conditions are identical and the vanes are identical for lower span and almost identical for higher span.

13 Appendix D

Results for TRS3-5 are presented. The results show similar trends as TRS1.

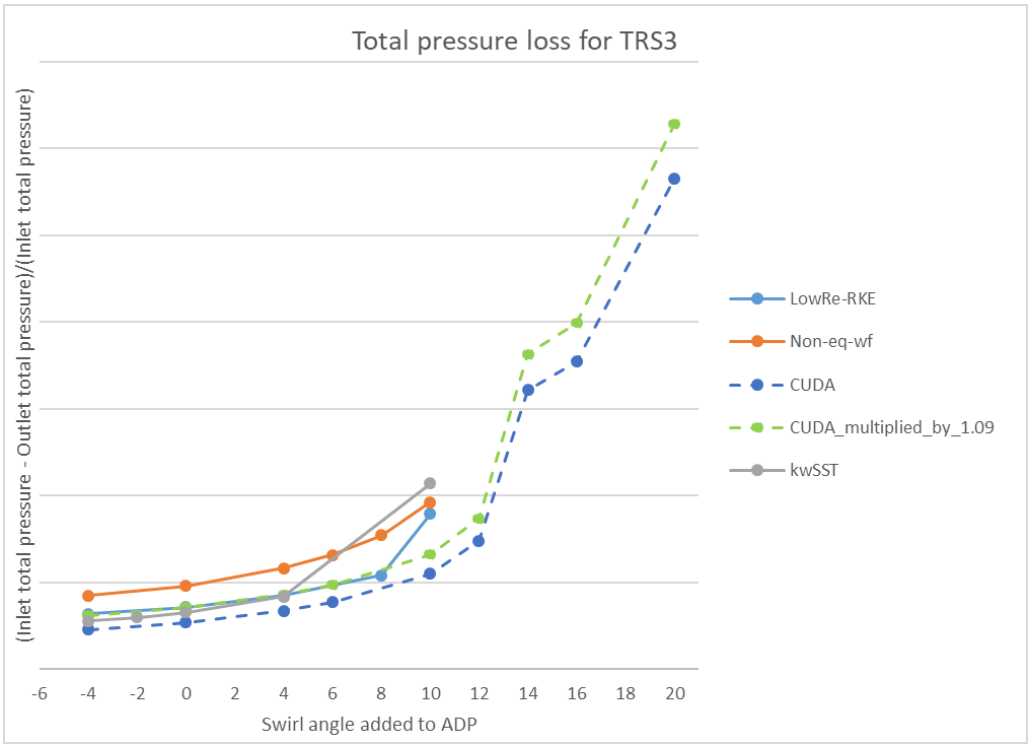


Figure 23 Loss bucket for TRS3.

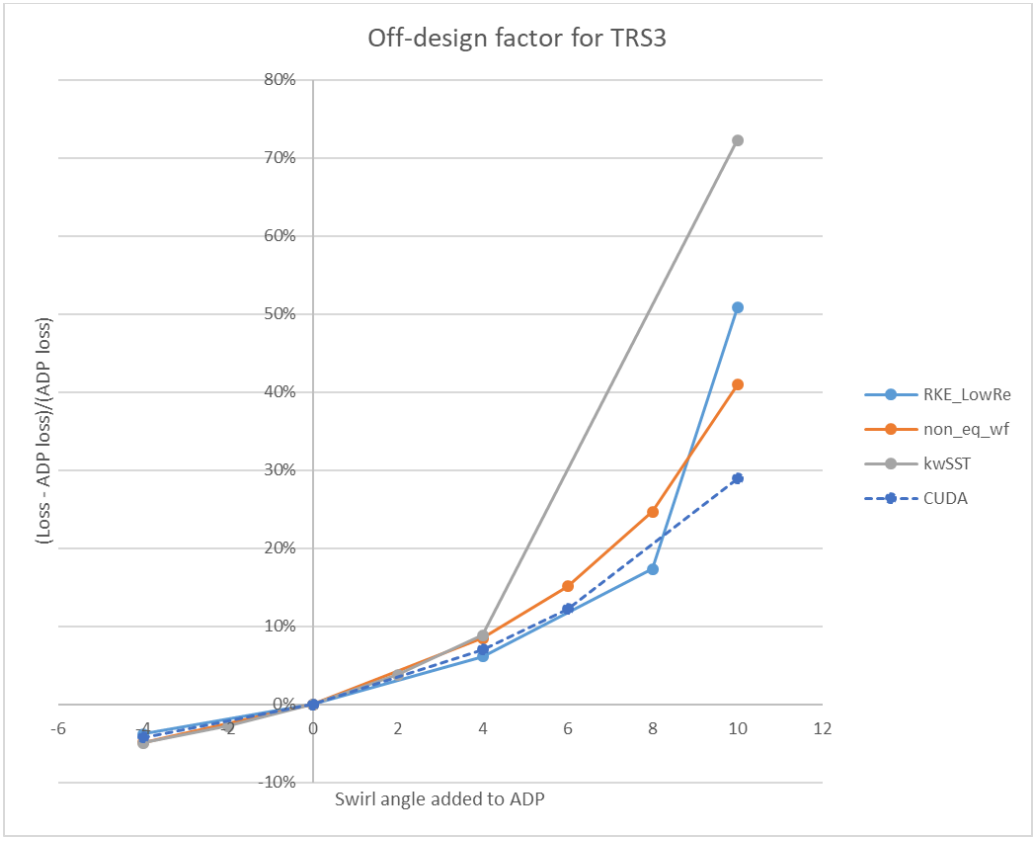


Figure 24 Off-design factor for TRS3.

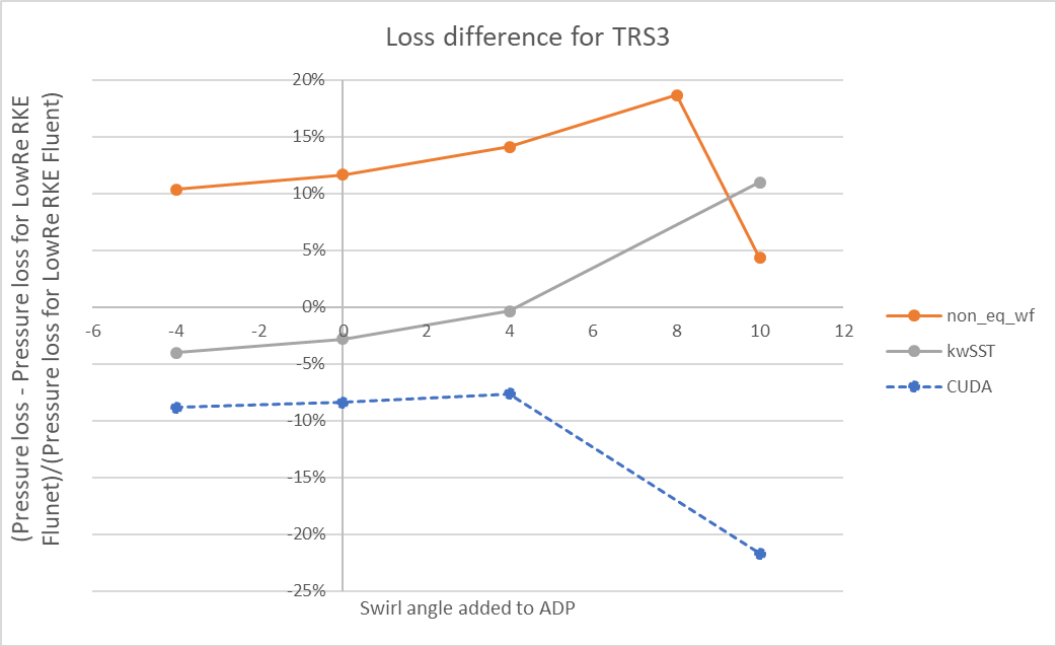


Figure 25 Loss difference for TRS3.

In Figure 23, Figure 24 and Figure 25 the results for TRS3 are presented. Most of the results are only presented up until ADP+10 degrees swirl angle. For the off-design factor CUDA predicts extremely similar values as LowRe-RKE. The loss difference for CUDA is almost constant at -8% until separation occurs for LowRe-RKE. The loss difference seems to be decreasing early due to that values for swirl angle 6 and 8 are missing. Either CUDA or LowRe-RKE are not run for these swirl angle when this happens, in this case CUDA is not run.

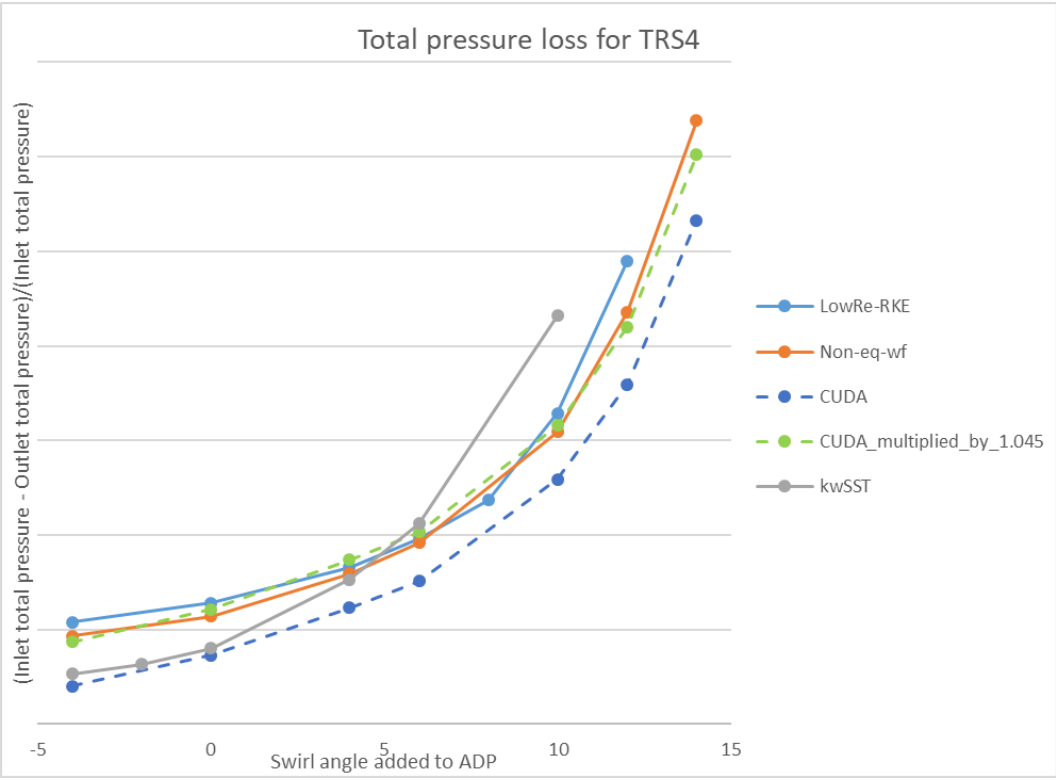


Figure 26 Loss bucket for TRS4.

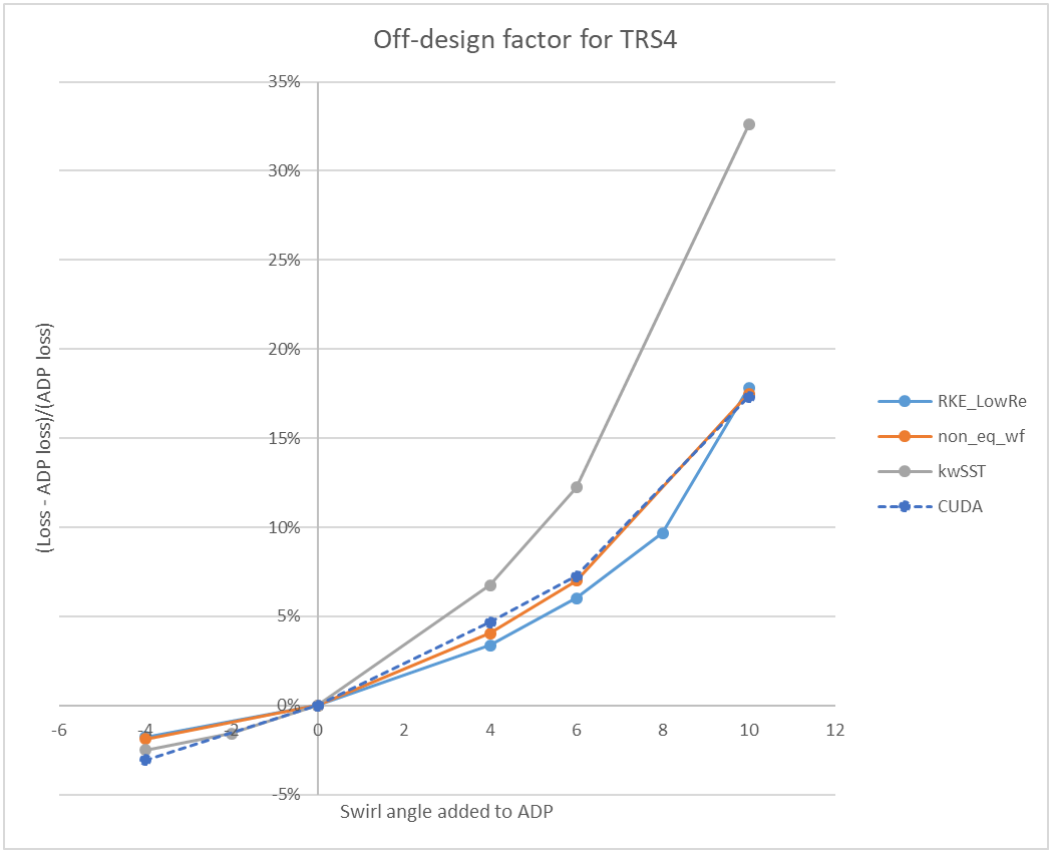


Figure 27 Off-design factor for TRS4.

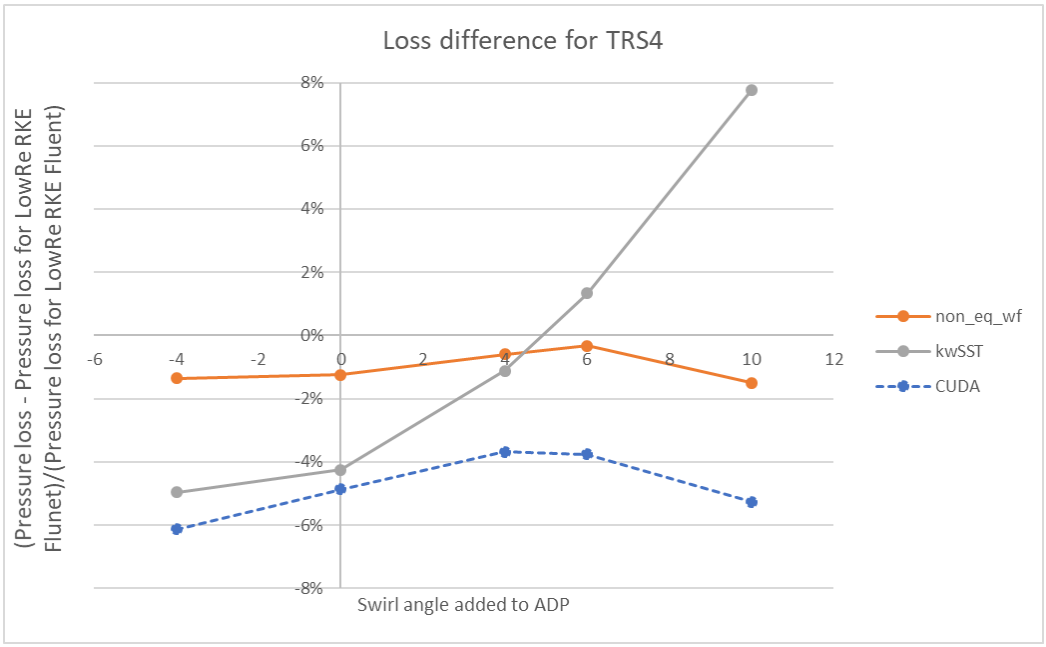


Figure 28 Loss difference for TRS4.

In Figure 26, Figure 27 and Figure 28 the results for TRS4 are presented. Some of the trends seen for the other TRS are not predicted for TRS4. For the off-design factor, non-eq-wf predicts values closer to LowRe-RKE than CUDA does. And for the loss difference, the CUDA values are non-constant even before separation occurs. The mean value of the loss difference is around the value -5%.

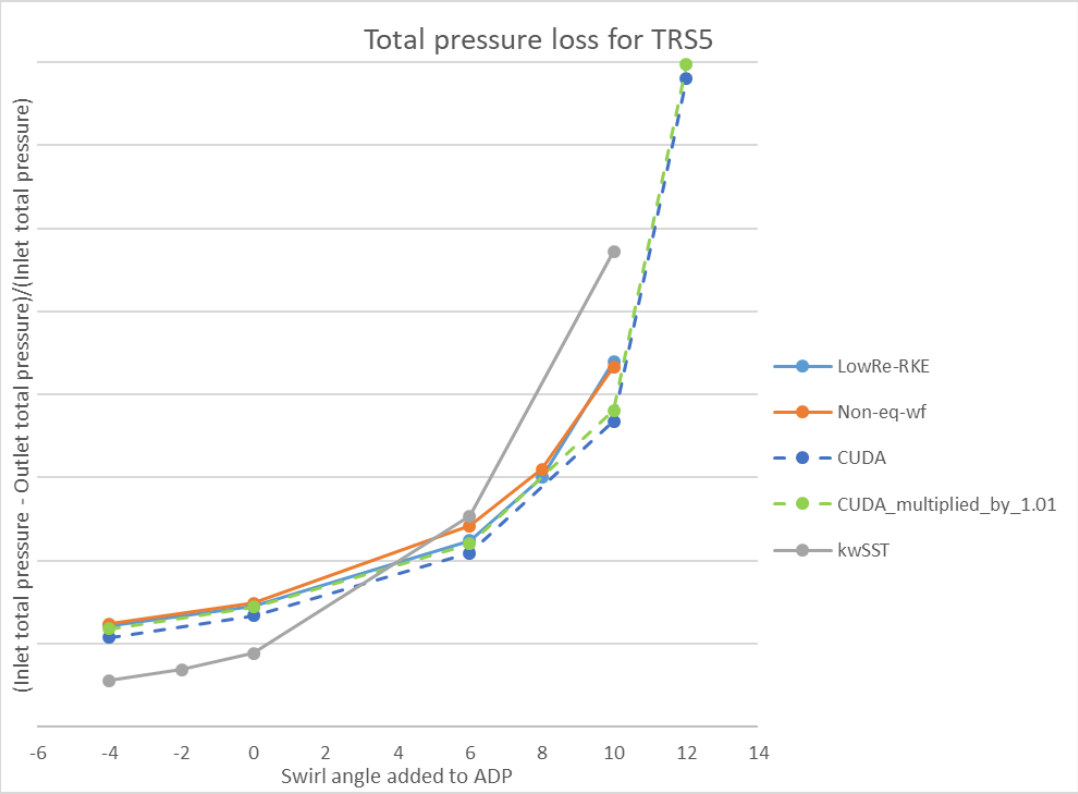


Figure 29 Loss bucket for TRS5.

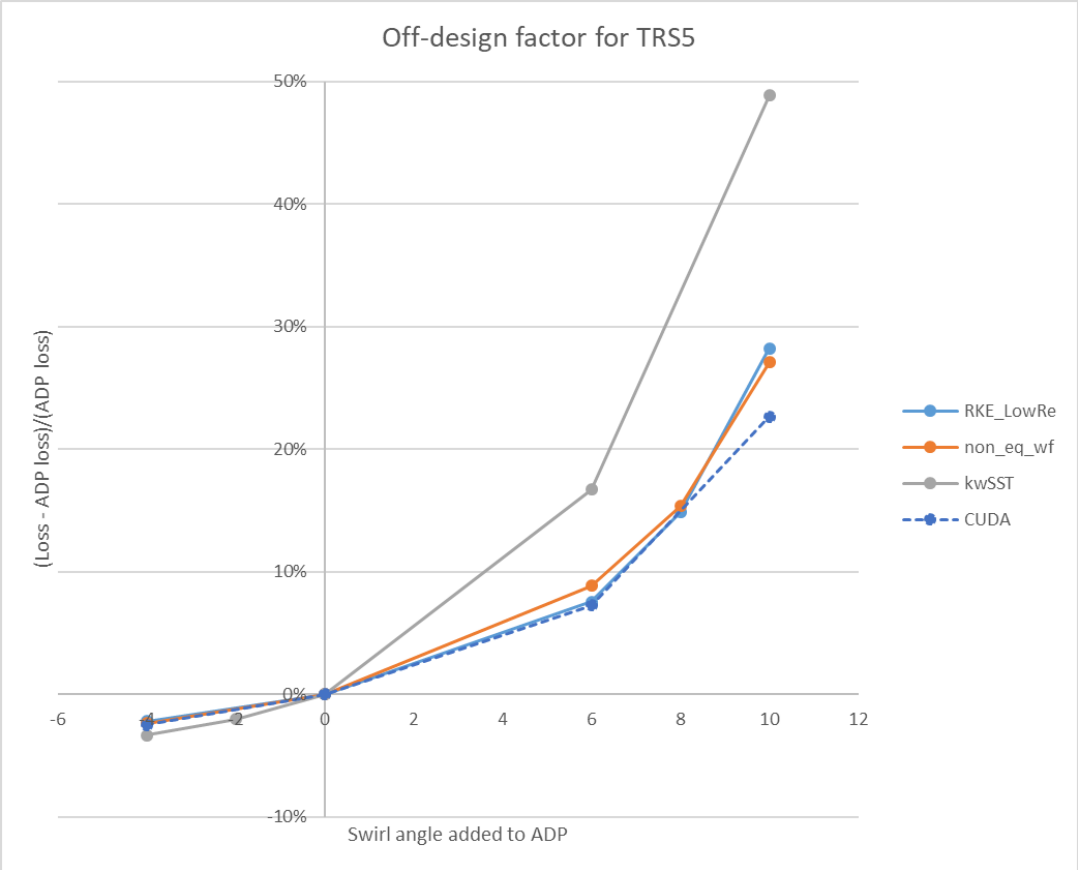


Figure 30 Off-design factor for TRS5.

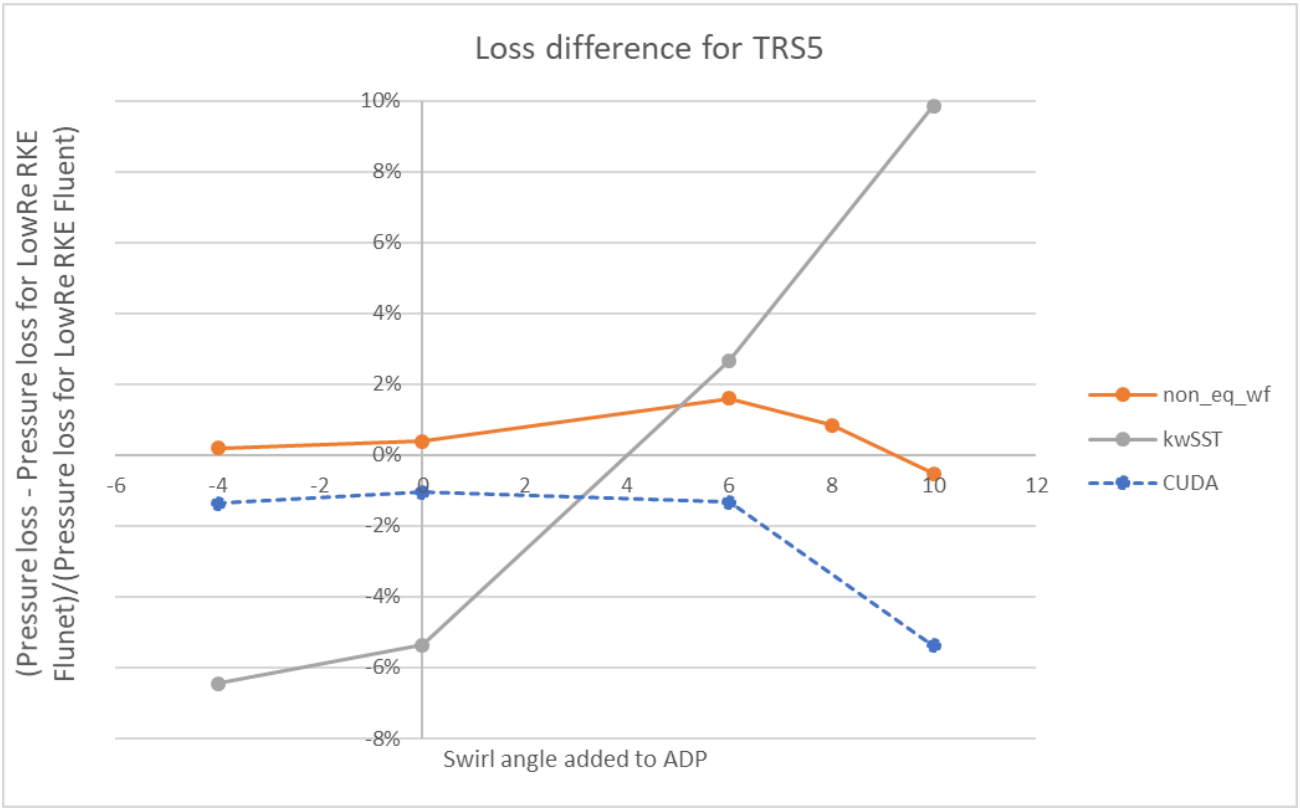


Figure 31 Loss difference for TRS5.

In Figure 29, Figure 30 and Figure 31 the results for TRS5 are presented. For TRS5 the results for all RKE simulations are similar. Even the absolute value of the pressure loss is similar for CUDA and LowRe-RKE. The off-design factor is almost identical for CUDA and LowRe-RKE and the loss difference is only -1%, until separation occurs.

14 Appendix E

For TRS4&5, a more thorough investigation of tangential variation at different span for the static pressure at the TRS inlet was carried out. Since the TRS represent vanes located on the same physical TRS, both a comparison of software and designs could be investigated.

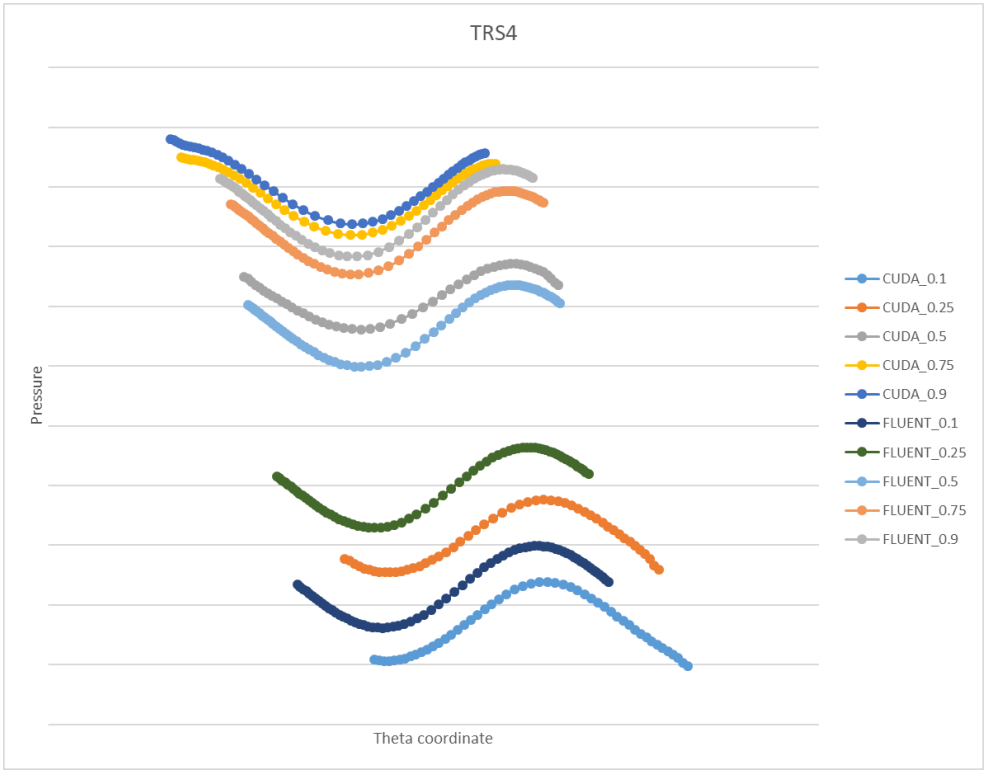


Figure 32 Pressure at tangential span for TRS4.

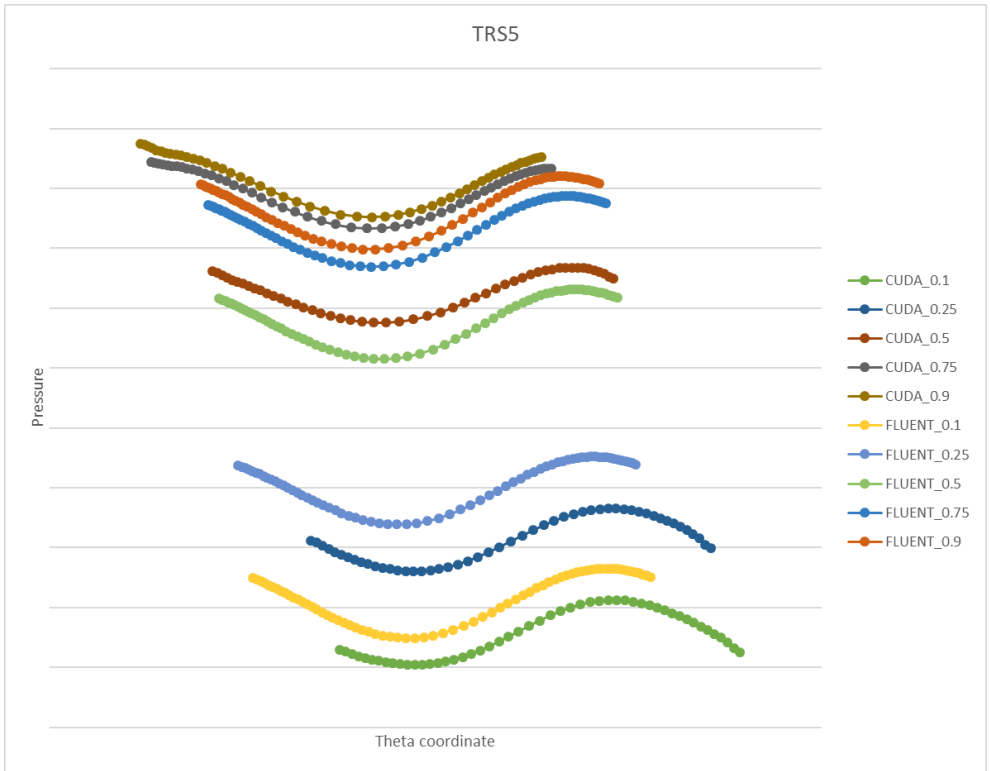


Figure 33 Pressure at tangential span for TRS5.

In Figure 32 and Figure 33 the pressure at tangential span is visualized for TRS4&5. In the figure the results can be shifted in direction of the theta coordinate, this is because the coordinate is periodic and not starting at the same value for the two software. The starting theta coordinate is defined in different ways in the software. The shape of the graphs are similar, but there is a difference in predicted pressure. CUDA predicts lower pressure at lower span and higher pressure at higher span, this is true for both geometries of the TRS. It is difficult to draw any further conclusions without values though.

Results for the two software are visualized in Figure 34 and Figure 35. Several trends can be recognized in both software. Predictions of which TRS have higher pressure at a specific span seems to agree between the software. The prediction that the pressure is similar for the higher spans seems to agree as well. The jump in pressure from span 0.25 to 0.5 may be more visible in CUDA, but even this result is visible in Fluent.

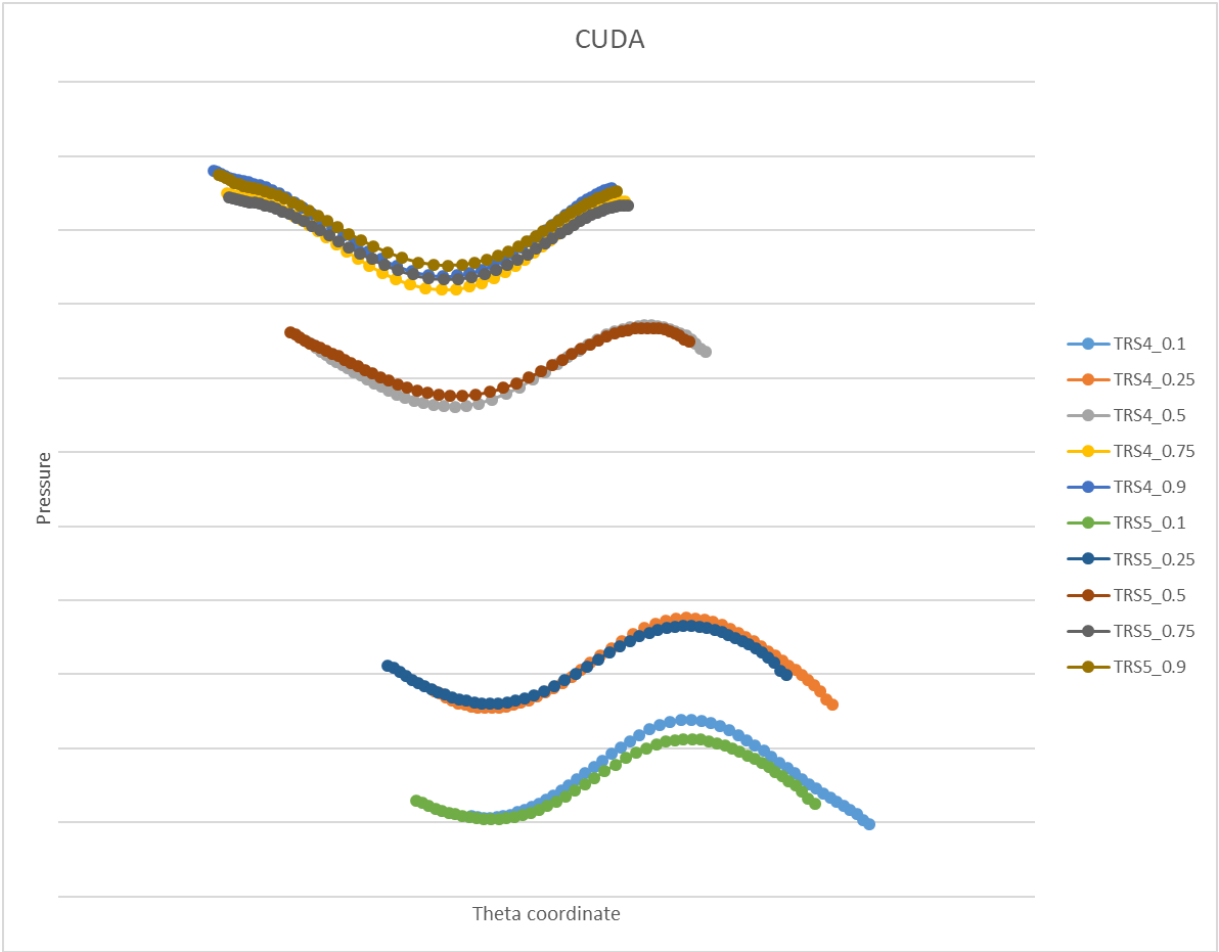


Figure 34 Pressure at tangential span for CUDA.

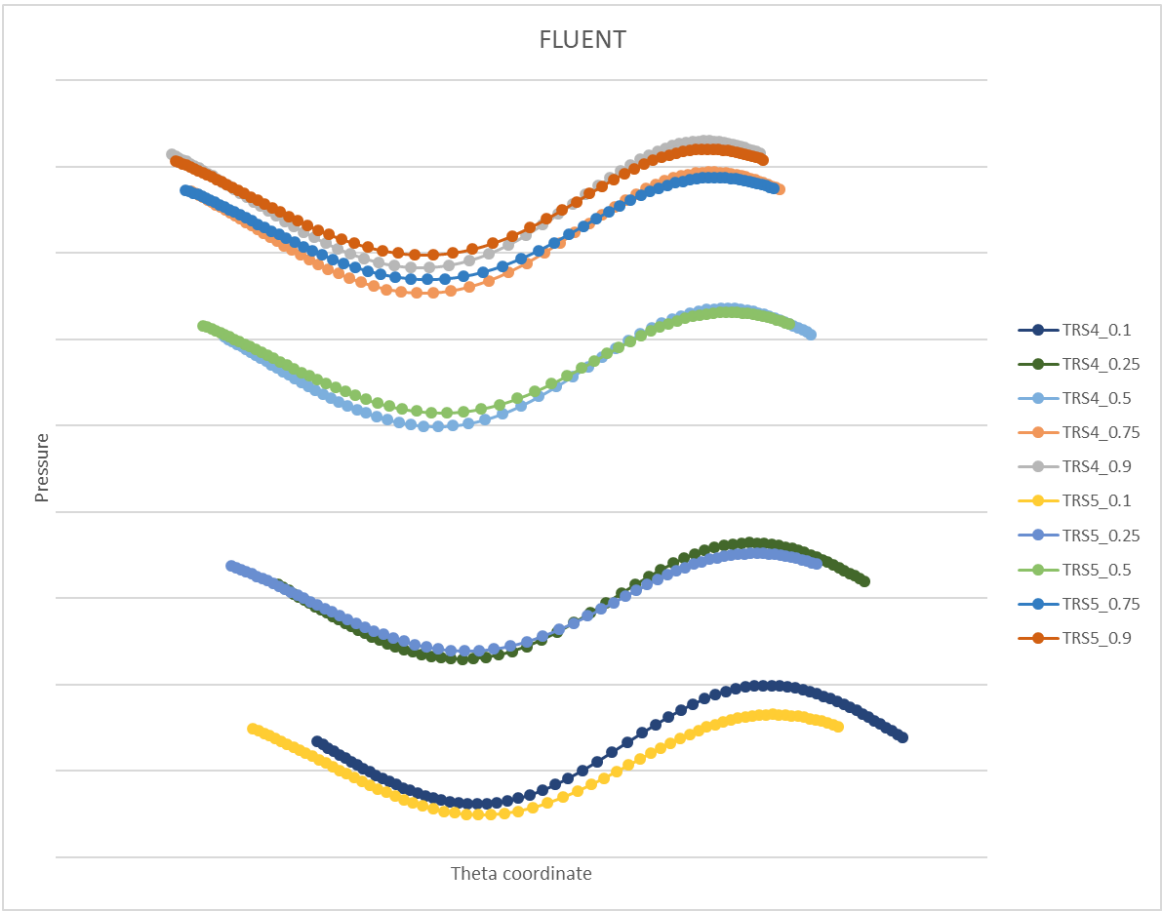


Figure 35 Pressure at tangential span for Fluent.