

Implementation of a Reduced-Order Model for the Navier-Stokes Equations using the POD-Galerkin Method

a project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Jens Polzin

January 2022

approved by

Dr. Joseph Schulz
Faculty Advisor



San José State
UNIVERSITY

© 2021
Jens Polzin
ALL RIGHTS RESERVED

ABSTRACT

Implementation of a Reduced-Order Model for the Navier-Stokes Equations using the POD-Galerkin Method

Jens Polzin

The Proper Orthogonal Decomposition (POD)-Galerkin method was applied to the finite volume approximation of the Navier-Stokes Equations. For validation, the lid-driven cavity flow test case was selected. An incompressible, 3-dimensional flow simulation was performed using Ansys Fluent. Flow state data obtained from this simulation was then used to calculate the dominant modes of the system. Inserting these modes into the Galerkin projection of the Navier-Stokes equations generated a linear system of equations capable of predicting the flow behavior within the system. The implemented technique decreased the calculation time by several orders of magnitude while also introducing an instability to the system. Reduced-Order Model (ROM) results immediately start to deviate noticeably from the numerical results, highlighting the need for further validation of the ROM program.

Acknowledgments

I would like to thank Joseph Schulz for his counsel and encouragement. His mentorship and assistance were extraordinarily valuable to me within this project's scope and beyond. I am also grateful for my friends and fellow students at San José State, with whom I have burnt the midnight oil countless times. Finally, I would like to thank my family for their continuous love and support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Literature Review	2
1.2.1	Numerical Simulation	2
1.2.2	Reduced-Order Modeling	5
1.3	Project Proposal	7
2	Flow Simulation	8
2.1	Governing Equations	8
2.1.1	Incompressible Navier-Stokes Equations	8
2.1.2	Incompressible RANS Equations	8
2.1.3	Additional Equations	9
2.2	Finite-Volume Methods	10
2.2.1	Inviscid Flux Calculation	11
2.2.2	Viscous Flux Calculation	13
2.2.3	Discretization in Time	13
2.2.4	Gradient Reconstruction	13
2.3	Turbulence Model	15
3	Test Data Generation	17
3.1	Simulation Inputs	17
3.2	Mesh	19
3.3	Wall Function	23
3.4	Simulation Results	24
3.4.1	Laminar Flow Case	25
3.4.2	Turbulent Flow Case	28
4	Reduced-Order Model	31
4.1	Proper Orthogonal Decomposition	31
4.2	Basis Calculation	32
4.3	Galerkin-Projection	34
5	Model Analysis	35
6	Conclusion and Outlook	41

List of Figures

1.1	Gradient reconstruction scheme classifications	3
3.1	Isometric view of the simulated cavity	17
3.2	Imposed velocity distribution on top of the cavity	19
3.3	U-velocity profile with linear and log-law approximation. Channel flow, DNS-data, $Re_\tau=395$ [1]	20
3.4	Mesh for the laminar flow case	22
3.5	Mesh for the turbulent flow case	23
3.6	Cavity flow at $\Delta t = 5$ for the laminar flow case	25
3.7	Flow quantities at $\Delta t = 5$ for the laminar flow case	26
3.8	Pressure contours for the laminar flow case at different positions	26
3.9	Volume-averaged turbulent kinetic energy over time for the laminar flow case	27
3.10	Flow quantities at the central cross-section in y-direction for the laminar flow case	27
3.11	Cavity flow at $\Delta t = 5$ for the turbulent flow case	28
3.12	Flow quantities at $\Delta t = 5$ for the turbulent flow case	29
3.13	Pressure contours for the turbulent flow case at different positions	29
3.14	Flow quantities at the central cross-section in y-direction for the turbulent flow case	30
5.1	POD reconstruction properties of the laminar flow case	35
5.2	Numerical results at different time steps	36
5.3	Cell velocity profile over time	37
5.4	POD mode velocity profiles at the central cross section in XZ direction	38
5.5	Temporal coefficients of the POD-Galerkin ROM	39
5.6	ROM cell velocity profile over time	40
5.7	RMSE over time	40

Symbols

Symbol	Definition	Units (SI)
a	Temporal ROM coefficient	----
B	ROM matrix	----
c_f	Skin friction coefficient	----
C	ROM matrix	----
\mathbf{g}	Numerical flux	----
GR	Growth rate	----
k	Turbulent kinetic energy	$\text{m}^2 \text{s}^{-2}$
p	Pressure	Pa
q	Heat flux	W m^{-2}
R	Specific gas constant	$\text{J kg}^{-1} \text{K}$
Re	Reynolds number	----
S	Sutherland temperature	K
t	Time	s
T	Temperature	K
\mathbf{u}	Velocity vector	m s^{-1}
u	Velocity in x-direction	m s^{-1}
v	Velocity in y-direction	m s^{-1}
w	Velocity in z-direction	m s^{-1}
y^+	Non-dimensional distance from the wall	----
Greek Symbols		
δ	Boundary layer thickness	[m]
ε	Dissipation rate of turbulent kinetic energy	$\text{m}^2 \text{s}^{-3}$
κ	Thermal conductivity	$\text{W m}^{-1} \text{K}$
μ	Dynamic viscosity	Pas
ν	Kinematic viscosity	m s^{-2}
ρ	Density	kg m^{-3}
τ	Shear stress	N m^{-2}
ϕ	Spatial ROM mode	----
Subscripts		
$()_{cc}$	Cell-center	----
$()_w$	Wall	----
$()_\tau$	Turbulent	----

Acronyms		
AUSM	Advection Upstream Splitting Method	-----
CFD	Computational Fluid Dynamics	-----
CVRC	Continuously Variable Resonance Combustor	-----
DNS	Direct Numerical Simulation	-----
FDS	Flux-Difference Splitting	-----
FOM	Full-Order Model	-----
FV	Finite-Volume	-----
FVM	Finite Volume Method	-----
FVS	Flux-Vector Splitting	-----
LSPG	Least-squares Petrov Galerkin	-----
PDE	Partial Differential Equation	-----
POD	Proper Orthogonal Decomposition	-----
RANS	Reynolds-averaged Navier-Stokes Equations	-----
RMSE	Root Mean Square Error	-----
ROM	Reduced-Order Model	-----
SST	Menter Shear Stress Transport	-----
SVD	Singular Value Decomposition	-----
TKE	Turbulent kinetic energy	-----
URANS	Unsteady Reynolds-averaged Navier-Stokes Equations	-----
WLSQ	Weighted Least-Squares	-----

1 Introduction

1.1 Motivation

The research field of Computational Fluid Dynamics (CFD) is directly linked to the computational resources available. Therefore, it has seen rapid advancements over the past decades. The overall growth in computational power is quantified and emphasized in Moore's law, stating that the transistor count of state-of-the-art integrated circuits doubles roughly every two years. This observation has been accurate for decades and is projected to remain valid for years to come [2]. New simulation capabilities have become quickly accessible in the recent past in correspondence with this trend. In the early 1970s, calculating a transonic potential flow solution around an airfoil depicted the limit of what was possible. By 1975, computing this solution over a swept wing was technically feasible. In 1982, the problem was solved for an entire aircraft [3]. This trend continued and enabled the simulation of evermore complex and even reacting flow problems. In addition to the applications, the usage of CFD has also changed. Previously used as a research tool only available to institutions with access to immense computational resources, it has become a design tool commonly employed in industrial development processes. Nowadays, small-scale simulations are often run on common computers or laptops to validate or review design choices. It is also to be noted that this development has been backed by the increased availability of open-source CFD software.

Yet, CFD simulations remain demanding. Obtaining accurate solutions for complex flows is often only possible using simplification models that result in errors of varying extend. A complete Direct Numerical Simulation (DNS) is technically infeasible for most flow cases. This complexity is inherent to CFD, which formulates multiscale, multiphysics problems. It combines several scientific disciplines (e.g., thermodynamics, gas dynamics, chemistry) into a single problem expressed on multiple length scales. In a turbulent flow, the largest scale is the *integral length scale*. A direct energy cascade then transports energy down to *Kolmogorov length* scales. While eddies below this scale are dampened by viscous effects of the fluid, even smaller length scales can be of importance for reacting flows. Combustion zones can be thinner than the *Kolmogorov length*, and combustion itself happens on a molecular length scale.

As a result, CFD has become a critical resource for numerous engineering applications while remaining exceedingly challenging at the same time. This contrast is especially apparent in aerospace applications, which rely on highly optimized designs for safety and efficiency while often subjected to complex flow behavior. Hence, a reduction of the problem complexity is critical to support full CFD functionality at a manageable computational expense. This fact often justifies sacrificing accuracy to a degree to achieve a better simulation performance.

1.2 Literature Review

Since this project uses a combination of different research fields, a literature review can be divided into two subsections. The first will cover the numerical simulation that generates the baseline and provides training data for the ROM. The second subsection will discuss literature connected to the ROM itself.

1.2.1 Numerical Simulation

All numerical simulations for the project have been conducted using tools of the Ansys software suite, especially the CFD module Ansys Fluent 2021 R2¹. The software is based on a plethora of scientific literature and centuries of fluid dynamics research. Essential papers are reviewed in the following paragraphs.

Physical flow equations composing the core of a CFD problem were formulated centuries ago. The Euler equations describing adiabatic and inviscid flow were phrased in 1755 [4]. A set of coupled differential equations describing the viscous flow of Newtonian fluids was composed in 1822 and are commonly known today as the Navier-Stokes equations [5].

Since turbulent flows exhibit a seemingly random and chaotic behavior, averaged flow values are often of higher interest than the fluctuating values themselves. Osborne Reynolds proposed separating a fluctuating variable u into its average, and a fluctuating part in 1895 [6], known today as *Reynolds decomposition*. The average of the fluctuating variable part is 0.

$$u = \bar{u} + u' \quad (1.1)$$

$$\overline{u'} = 0 \quad (1.2)$$

After applying this decomposition to the Navier-Stokes equations, they are referred to as the Reynolds-averaged Navier-Stokes Equations (RANS). For non-stationary processes, the so-called Unsteady Reynolds-averaged Navier-Stokes Equations (URANS) should be used, where the derivatives with respect to time are not equal to zero.

A density-based average of some variables is often beneficial when compressible fluid behavior, strong temperature gradients, or combustion occurs. This approach was first proposed by Favre in 1965 [7] and is hence referred to as the *Favre-average*.

$$u = \tilde{u} + u'' \quad (1.3)$$

$$\tilde{u} = \frac{\overline{\rho u}}{\bar{\rho}} \quad (1.4)$$

$$\overline{\rho u''} = 0 \quad (1.5)$$

¹<https://www.ansys.com/products/fluids/ansys-fluent>

Since the analytical solution to the resulting Partial Differential Equations (PDEs) is rarely known or calculable, numerical methods are commonly employed. Ansys Fluent was used to set up a Finite Volume Method (FVM) for this paper. The literature on these methods is extensive. An overview is provided by the books of Munz and Westermann [8], MacCormack [9], and Lomax, Pulliam, and Zingg [10] or Magoules [11].

A gradient reconstruction scheme needs to be selected when setting up the simulation. Information about the gradient values of state variables at different grid positions is used for linear upwind differencing schemes and non-orthogonal correctors. Figure 1.1 shows different gradient reconstruction schemes for the *C-gradient* at the cell-center and the *F-gradient* at the center of the cell face. It is possible to formulate a node-based and a cell-based configuration for every gradient reconstruction scheme shown. A cell-based version of the weighted least-squares schemes was also used to calculate the gradients needed for the Galerkin projection.

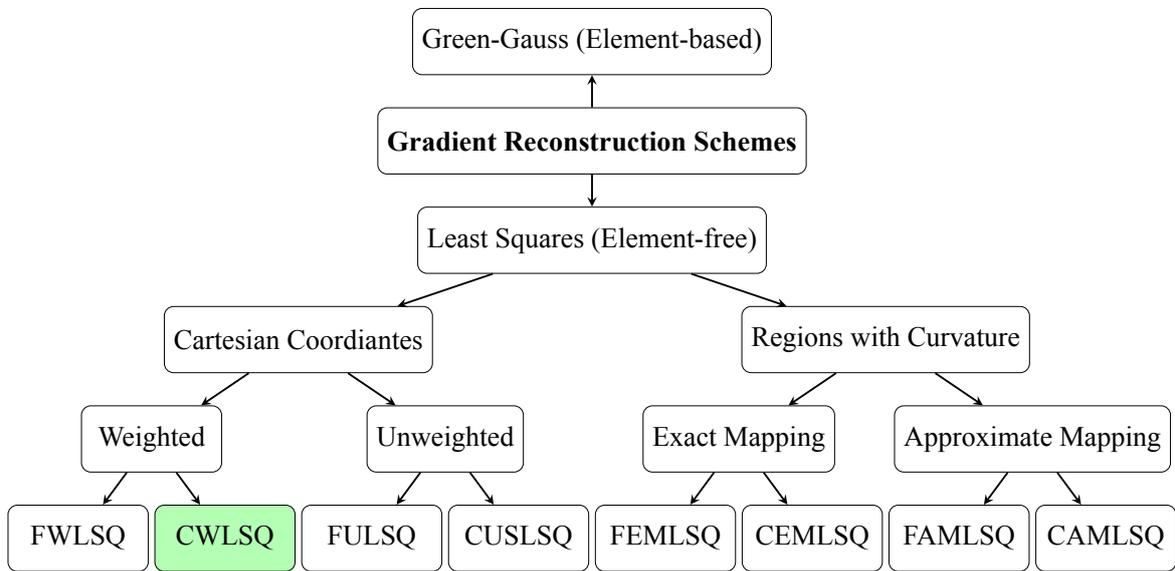


Figure 1.1: Gradient reconstruction scheme classifications

Diskin and Thomas [12] describe different gradient reconstruction schemes in detail and analyze their accuracy for different structured high aspect ratio grids. They concluded that a node-centered Weighted Least-Squares (WLSQ) scheme produces a very accurate *C-gradient* reconstruction. Furthermore, they showed that the order of accuracy of gradient reconstruction schemes depends on the selected mesh type. This result is supported by the findings of Syrakos et al. [13]. They write that the assumed second-order accuracy of Green-Gauss and WLSQ for every grid is a misconception and only holds for structured grids. Theoretical analysis revealed first-order accuracy for unstructured grids for a cell-centered *CWLSQ* scheme and possible zeroth order of accuracy for a cell-based Green-Gauss *C-gradient* scheme. This result was confirmed by numerical simulations using an in-house CFD code as well as OpenFOAM.

The flux across the cell faces in a FVM can be separated into convective and viscous flux. A numerical flux scheme is needed to discretize the convective flux. The Advection Upstream Splitting Method (AUSM) has been selected for all simulations conducted within the scope of this project. The method was first introduced by Liou and Steffen in 1993 [14]. Its objective was to mimic the Flux-Difference Splitting (FDS) schemes in accuracy while achieving an order of operations per grid point similar to Flux-Vector Splitting (FVS) methods. While FDS methods like Roe- [15] or Osher-splitting [16] have shown a high accuracy in the past, both are of second-order. This order is due to the necessary differentiation of fluxes for the Roe-splitting. Osher-splitting requires the calculation of the intermediate states that connect the states between grid points, which is computationally expensive. While FVS methods like Steger-Warming [17] and Van-Leer splitting [18] are simple and computationally inexpensive, their accuracy is significantly reduced by numerical diffusion. Liou and Steffen propose a scheme that separates the pressure term from the convective flux. It recognizes that this term is dependent on the acoustic wave speeds, unlike the convective terms, which are governed by the flow velocity. By basing the scheme on physics, a stable first-order method was created, which has been shown to produce results similar in accuracy to the Roe-scheme. Furthermore, the scheme has been shown to calculate accurate results even in some fringe cases the Roe-scheme can not model accurately, like the 2D supersonic flow over a blunt cylinder. In 1995, Liou presented the modified AUSM+ scheme [19] which proved to be more reliable than the original scheme.

Nine additional components are needed to solve the RANS in three dimensions, six for the symmetric Reynolds-stress tensor and three for the turbulent heat flux vector. This closure is performed using simplifying turbulence models. One capable model is the Menter Shear Stress Transport (SST). It was first described in 1994 by F. R. Menter [20]. By blending the two, it addresses problems arising in either the $k - \epsilon$ or the $k - \omega$ models. It suggests using the $k - \omega$ model in near-wall regions and a $k - \epsilon$ model for the rest of the flow with a smooth transition in between. Furthermore, the SST model addresses inaccuracies arising in both the $k - \omega$ and the $k - \epsilon$ model. They tend to overestimate the wall shear stress, which leads to inaccurate flow separation estimation. The SST model, therefore, introduces a shear stress limiter, which is implemented in the calculation of the eddy viscosity. Due to its low computational cost and high accuracy, it is one of the most popular turbulence models currently used for CFD simulations.

It is known that viscous flows generate a boundary layer near walls or solid objects inside the flow. Since its first identification by Prandtl [21], this boundary layer has been studied intensively [22, 23]. It is caused by the *no-slip condition* and has significant effects on the overall flow behavior. While its dimensions in the wall-normal direction are small, it contains steep temperature and velocity gradients. This behavior has to be considered when generating the mesh for a viscous flow simulation. Any mesh that fails to capture these near-wall gradients will cause the simulation to become inaccurate and may introduce instabilities or divergence. The two most important methods for boundary layer treatment within modern CFD simulations are summarized by Bredberg [24]. The report details integrating and wall-function approaches and discusses the respective derivations. The former approach resolves

boundary layer behavior by utilizing an appropriate low-Reynolds-number turbulence model and a substantial mesh refinement near the wall. The wall-function approach works using a coarser mesh and so-called wall functions, which mimic the near-wall flow behavior. For this project, a wall-function approach was selected. The mesh was generated accordingly. The adaptive wall function method utilized was first proposed by Knopp et al. [25].

The flow over a lid-driven cubical cavity was selected as an appropriate test case. It has been studied extensively both experimentally [26] and using DNS [27]. Prasad and Koseff used polystyrene beads and two-dimensional laser-Doppler anemometer measurements to experimentally collect velocity data for different Reynolds number and spanwise aspect ratios of the cavity along the horizontal and vertical centerlines. Leriche and Gavrilakis used a DNS to examine the flow inside the cavity in detail. They imposed a velocity profile on top of the cavity flow to mimic flow experiments more realistically. Their paper notes the long time scales exhibited by the flow. The simulation revealed a mostly laminar flow with most of its total kinetic energy near the lid. Energy from this near-lid flow is then transmitted onto a downflowing jet that splits into two upwards streaming jets near the bottom of the cavity.

1.2.2 Reduced-Order Modeling

POD, also known as principal component analysis or Karhunen-Loève decomposition, was initially applied to fluid dynamics in 1967 by Lumley [28]. The paper describes the vector field partition of a turbulent fluid motion into separate functions. Each resulting function contains a part of the flow's kinetic energy. Lumley noted that the method was previously independently suggested by multiple scientists [29]. The application of POD in fluid dynamics has been heavily researched since and covered by numerous papers and articles [30, 31].

POD uses so-called snapshot data of the system at different points in time. The snapshots are made up of each mesh cell's state variables, for example, density and velocity data. These snapshots are concatenated and used to calculate a covariance matrix. The eigenvectors of this matrix are the dominant modes of the system, as described by Weiss [32]. The eigenvalues reflect the Turbulent kinetic energy (TKE) captured by the respective eigenvectors.

For most CFD applications, the total number of state variables far exceeds the number of snapshots used. Hence, Sirovich suggested in 1987 to exchange the temporal and spatial data while calculating the POD basis [33]. This method can accelerate the computation in cases where the number of grid points is larger than the number of snapshots used to construct the basis. This procedure is known as the method of snapshots or the Snapshot POD-method.

The POD-Galerkin method assumes that these modes are only a function of space and can be linearly combined with temporal coefficients to approximate the original system. It is described in detail by Kunisch and Volkwein [34] and Iliescu and Wang [35]. These linear combinations can then be inserted into the Navier-Stokes or Reynolds-Averaged Navier Stokes equations, as described by Sirovich [33] and Lorenzi et al. [36].

The ROM obtained through application of the POD-Galerkin method to the Navier-Stokes equations often quickly become unstable. This issue is examined in the work of Akhtar et al. [37]. Different stabilization techniques, supremizer enrichment and a Poisson equation approach, are compared by Stabile et al. [38]. Supremizer enrichment relies on the inf-sup condition and is described by Rozza and Veroy [39]. The second approach for ROM stabilization changes the incompressibility condition with the Poisson pressure equation. The application of this approach for finite volume methods was proposed in [40] by Stabile et al. The paper concludes that the first approach seems to produce superior results for pressure fields. In contrast, the second was able to model velocity fields more accurately. Computationally, the second approach was less demanding while both managed to produce a significant speed-up compared to the Full-Order Model (FOM). For long-term integrations, the supremizer enrichment technique seems to be less stable.

The application of the POD-Galerkin method to the compressible Navier-Stokes Equations is discussed in detail by Rowley et al. [41]. It differs from the incompressible Navier-Stokes equation treatment since the pressure and other variables influence the flow behavior. Therefore, additional conservation equations and a different choice for the inner product are required.

Carlberg et al. showed that ROM stability could be increased by changing the projection [42]. They minimized the residual by applying a Petrov-Galerkin projection to the time discretized FOM and then using a weighted least-squares norm. Their work proves that the error bounds for the resulting Least-squares Petrov Galerkin (LSPG) method are either equal to or smaller than the error bounds of the Galerkin projection in the backward Euler case if the original function is Lipschitz continuous. Furthermore, it acknowledges how the two approaches are equivalent under certain conditions, for example, when the time step Δt approaches zero. Their resulting ROM has proven to be more accurate and stable in the examined test case than the POD-Galerkin ROM, given that a proper time step was chosen. Since the resulting computation time for the non-linear least-squares minimization could exceed even the calculation time of the FOM, the authors propose using the gappy POD method, developed by Everson and Sirovich [43]. They reduce the problem to a linear least-squares minimization by sampling rows from the basis matrix and filling in the gaps. A Gauss-Newton with Approximated Tensors ROM was built using a greedy method, which sped up computation significantly.

Huang et al. [44] stated that reacting flows are especially sensitive to resolution and become, in many cases, numerically unstable. The paper examines the root causes of this emerging instability. Three possible origins were proposed, the first being conservation errors of the ROM. The second is a loss of dissipation caused by the truncation of high-energy modes in the ROM. The last proposed cause was the existence of unphysical local phenomena. Their investigation shows that only the latter has a measurable effect on ROM stability. In their calculations, the occurrence of temperatures below 0 K was observed, after which the ROM tends to explode. It is concluded that this occurrence can be attributed to the Gibbs phenomenon, which is commonly present in reacting flow simulations due to the steep temperature gradients. Temperature constraints were imposed on the ROM as a countermeasure. This action significantly improved the stability of the model. In addition, the paper notes that an expected stability im-

provement using the LSPG method, proposed by Carlberg et al. [42] was not observed in this context.

Since these models use the dominant modes calculated through a solution of the FOM, only state behavior within the space spanned by these modes can be captured. ROMs therefore tend to produce inaccurate results when solving hyperbolic PDE, which often characterize advection-based behavior. This limitation is addressed in another paper by Carlberg [45]. A method for increasing the reduced-order basis without reverting to the FOM is proposed. This method constructs a tree for state variables by examining the proportionality and anti-proportionality to each other and applying k-means clustering. It introduces an error approximation method, which is run for all top-level tree nodes in regular time intervals to assess the model's performance. Poorly performing nodes are split up if the total error exceeds a controllable threshold. This refinement broadens the ROM basis and allows a controllable model accuracy without requiring a FOM solve.

In addition to analytical methods, machine learning techniques can be employed to build ROMs. An example for this is described by Wang et al. [46]. The paper uses an artificial neural network to develop a non-intrusive ROM for unsteady flows. These models are, unlike conventional intrusive ROMs, completely decoupled the online stage and the training stage. The model was developed for a quasi-one-dimensional flow through the Continuously Variable Resonance Combustor (CVRC) at Purdue University (Indiana, U.S.) [47]. The resulting model generated robust results at a low computational cost.

1.3 Project Proposal

This project shall detail the application of the POD-Galerkin method to the incompressible Navier-Stokes equations. The resulting ROM shall then be trained using flow information from a well-understood test case, generated using a commercial CFD solver. A program shall be written which imports flow data, applies the POD-Galerkin method, and outputs solutions at different time steps. After verifying the solutions, and the performance of the ROM shall be assessed. First steps for the extension of the ROM construction program towards applicability to the RANS equations shall be made to accommodate future work.

2 Flow Simulation

The following chapter details the basis for the conducted flow simulation. A finite-volume method was used to numerically solve the PDEs describing the flow. A theoretical framework for extension towards turbulent flow is outlined after detailing the incompressible Navier-Stokes equations and the selected numerical methods. This includes the RANS equations and the SST turbulence model.

2.1 Governing Equations

Systems of parabolic PDEs, based on the conservation of mass momentum and energy, are used to characterize the flow behavior in a CFD simulation. These conservation equations are detailed in the following subsections. Additional equations needed to achieve closure of the RANS equations are detailed in section 2.3.

2.1.1 Incompressible Navier-Stokes Equations

The incompressible Navier-Stokes equations read as follows:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \frac{1}{\rho} \nabla p \quad (2.1)$$

where \mathbf{u} denotes the velocity vector, containing the flow velocities in x-, y- and z- direction. Incompressibility indicates that the divergence of the velocity vector is equal to zero, $\nabla \cdot \mathbf{u} = 0$. This relationship is known as the continuity equation. It can be seen that the Navier-Stokes equations contain a convection term $-(\mathbf{u} \cdot \nabla) \mathbf{u}$, a diffusion term $(\nu \nabla^2 \mathbf{u})$, and a pressure term $(\frac{1}{\rho} \nabla p)$. The resulting system of equations conserves momentum in every direction in space and mass.

2.1.2 Incompressible RANS Equations

Applying the Reynolds decomposition to the mass conservation equation introduced in 2.1.1 leads to the following relationship:

$$\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} + \frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{\partial w'}{\partial z} = \nabla \bar{\mathbf{u}} + \nabla \mathbf{u}' = 0 \quad (2.2)$$

By time-averaging this equation, the fluctuating terms become zero. Therefore, $\nabla \bar{\mathbf{u}} = 0$, and, as a consequence, $\nabla \mathbf{u}' = 0$. When applying *Reynolds-averaging* to equation 2.1, the following momentum equations can be derived:

$$\frac{\partial \bar{\mathbf{u}}_i}{\partial t} + \bar{\mathbf{u}}_j \frac{\partial \bar{\mathbf{u}}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{\mathbf{u}}_i}{\partial x_i \partial x_j} - \frac{\partial \overline{\mathbf{u}'_i \mathbf{u}'_j}}{\partial x_j} \quad (2.3)$$

where the shear tensor $S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ and δ_{ij} denotes the Kronecker-Delta. The equations are written down using *Einstein notation* for the sake of brevity.

2.1.3 Additional Equations

Furthermore, equations for the shear stress τ within the fluid, the heat flux q , and an equation of state are needed to define the system properly. In 1682, the shear stress for Newtonian fluids was described by the Newtonian fluid hypothesis. The heat flux can be calculated using Fourier's law of heat conduction from 1822. For most technical applications, the ideal gas law, first mentioned in 1834, can be used as an equation of state. Since the flow simulations within this paper are conducted for an incompressible fluid, this equation does not apply. Instead, the NIST real gas models implemented in Ansys were employed.

To calculate the shear stress τ , Sir George Stokes assumed isotropic flow, a linear dependency of the stress tensor of the strain rate tensor, and a divergence of τ equal to 0. The resulting stress tensor reads

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \frac{\partial u_k}{\partial x_k} \quad (2.4)$$

where δ_{ij} , again, denotes the Kronecker delta and λ the second coefficient of viscosity. A common assumption for λ is $-2/3\mu$ [48]. In addition, this shear stress calculation requires a model to evaluate the dynamic viscosity μ . Within the scope of this paper, a constant dynamic viscosity for water of $\mu = 0.001002 \text{ N m}^{-2} \text{ s}$ was assumed. For compressible gases, a temperature-dependent viscosity model can be deployed. Sutherland's law is frequently used. It was first described by William Sutherland in 1893 [49] and states the following:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + S}{T + S} \quad (2.5)$$

The respective parameters for standard air are listed in table 2.1 for later reference.

Table 2.1: Sutherland’s law coefficients for air [49]

Parameter	[Unit]	Value
μ_{ref}	$[\frac{kg}{ms}]$	1.716×10^{-5}
T_{ref}	[K]	273.15
S	[K]	110.4

If the temperature is considered, Fourier’s law for conductive heat transfer can be used to calculate the heat flux q in most cases.

$$q = -\kappa \nabla T \quad (2.6)$$

The thermal conductivity is herein denoted as κ . Constant Prandtl numbers for air of 0.72 (laminar), and 0.90 (turbulent) were assumed. Using the definition of the Prandtl number $Pr = \frac{c_p \mu}{\kappa}$, the thermal conductivity can then be calculated. For the specific heat capacity at constant pressure c_p , a calorically perfect gas was assumed. The value of c_p therefore stays constant at $4184 \text{ J kg}^{-1} \text{ K}^{-1}$ for water and $1004 \text{ J kg}^{-1} \text{ K}^{-1}$ for dry air.

If the ideal gas law can be used as an appropriate equation of state, it writes as follows:

$$p = \rho RT \quad (2.7)$$

where R denotes the specific gas constant, defined as $R = \mathbb{R}/M$. \mathbb{R} symbolizes the universal gas constant while M is the molecular weight of the gas. For dry air, the specific gas constant reads $287.058 \text{ J kg}^{-1} \text{ K}^{-1}$.

2.2 Finite-Volume Methods

Finite-Volume (FV) methods are the most common class of numerical techniques for the solution of hyperbolic PDEs and are also commonly employed for the solution of parabolic PDEs. They are the *de facto* standard for the numerical solution of non-linear problems and are hence frequently used in the field of CFD.

The foundation of any FV method is the separation of the flow domain into a finite number of discrete elements, commonly referred to as cells. Every cell features a respective value for each flow property in question, i.e., flow velocity in x-, y-, or z-direction or static pressure. The following conservation equation can be formulated, if the absence of sources or drains and a fixed domain are assumed:

$$\mathbf{u}_t + \nabla \cdot \mathbf{f}(\mathbf{u}) = 0 \quad (2.8)$$

where \mathbf{u} describes the cell state vector containing the calculated flow parameters. The index t denotes the partial derivative with respect to time, $u_t = \frac{\partial \mathbf{u}}{\partial t}$. The matrix \mathbf{f} denotes the flux across the cell borders. The divergence of the flux $\nabla \cdot \mathbf{f}(\mathbf{u})$ is also referred to as the residual $R(\mathbf{u})$. Equation 2.8 can be written for every cell in integral form:

$$\frac{d}{dt} \int_{\Omega_i} \mathbf{u} dV = - \int_{\partial\Omega_i} \mathbf{f}(\mathbf{u}) \cdot \mathbf{n} dS \quad (2.9)$$

The left-hand side describes the integral over the entire volume of cell i , and the right-hand side describes the surface integral along its borders. A matrix consisting of unit vectors orthogonal to the cell surfaces directed outward is denoted by \mathbf{n} . The abbreviation of \mathbf{u}_i^n for the mean values of state vector \mathbf{u} in cell i at time t_n is frequently used¹. The vector \mathbf{u}_i at time t_{n+1} can be calculated by integrating the flux over time.

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{1}{|\Omega_i|} \int_{t_n}^{t_{n+1}} \int_{\partial\Omega_i} \mathbf{f}(\mathbf{u}) \cdot \mathbf{n} dS dt \quad (2.10)$$

After assuming that cell i is constricted by a number of m edges E , this conservation equation can be written as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{1}{|\Omega_i|} \int_{t_n}^{t_{n+1}} \sum_{k=1}^m \int_{E_k} \mathbf{f}(\mathbf{u}) \cdot \mathbf{n} dS dt \quad (2.11)$$

The values for the flux \mathbf{f} at the faces are interpolated from the state vector values at the neighboring cell centers. When a face is only shared between two cells and the same interpolation scheme is used for both cells, it can be inferred that conservation of the flow quantities on the discrete level is achieved. This reconstructed flux is also known as the numerical flux and will be written as \mathbf{g} .

The flux calculation is commonly split into the reconstruction of the inviscid flux, meaning the convective and pressure term flux, and the viscous flux, caused by the diffusion term in the Navier-Stokes equations.

e

2.2.1 Inviscid Flux Calculation

To calculate the inviscid flux \mathbf{g}_f at every face of a cell, a number of numerical schemes can be employed. The selected AUSM scheme separates F_f into convective terms \mathbf{g}_C and pressure terms \mathbf{g}_p [14]. Using the conservation equation, described in detail section 2.1, the following can be formulated for the flux in x-direction for a 3-dimensional problem:

¹ $\mathbf{u}_i^n \equiv \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{u}(\mathbf{x}, t_n) d\mathbf{x}$ where $\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ for a 3-dimensional domain

$$\mathbf{g}_I = \mathbf{g}_C + \mathbf{g}_P = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho H \end{pmatrix} u + \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.12)$$

At the cell face, the convective numerical flux writes as

$$\mathbf{g}_C = Ma_{1/2} \begin{pmatrix} \rho a \\ \rho au \\ \rho av \\ \rho aw \\ \rho aH \end{pmatrix}_{L/R} \quad (2.13)$$

The index L/R denotes the selection of the left cell state values if $M_{1/2} \geq 0$ and the right cell state values otherwise. The advective velocity $M_{1/2}$ is defined as

$$M_{1/2} = M_L^+ M_R^- \quad (2.14)$$

Liou and Steffen selected Van Leer Splitting [50] for the split Mach number M^\pm :

$$M^\pm = \begin{cases} \pm \frac{1}{4} (M \pm 1)^2, & \text{if } \|M\| \leq 1; \\ \frac{1}{2} (M \pm \|M\|), & \text{otherwise.} \end{cases} \quad (2.15)$$

The pressure term p at the cell face is calculated using the following equation:

$$p_{1/2} = p_L^+ + p_R^- \quad (2.16)$$

A polynomial expansion of the characteristic velocities weighs the pressure splitting. Liou and Steffen propose two possible polynomial expansions: one first-order, one second-order. The second-order expansion writes as

$$p^\pm = \begin{cases} \frac{p}{4} (M \pm 1)^2 (2 \mp M), & \text{if } \|M\| \leq 1; \\ \frac{p}{2} (M \pm \|M\|) / M, & \text{otherwise.} \end{cases} \quad (2.17)$$

The same method is also used to estimate the inviscid fluxes in y-direction and z-direction.

A modified version of this scheme, AUSM+, is implemented in ANSYS and was used for the simulations.

2.2.2 Viscous Flux Calculation

The viscous flux \mathbf{g}^v is less dominant than the convective flux in typical flows. It is defined as

$$\mathbf{g}^v = \mu_{tot} \begin{bmatrix} \cdot \\ \frac{\partial x}{\partial x_i} \\ \frac{\partial y}{\partial x_i} \\ \frac{\partial z}{\partial x_i} \end{bmatrix} \quad (2.18)$$

Where μ_{tot} is the total viscosity, $\mu_{tot} = \mu_{dyn} + \mu_{urb}$. The viscous flux is driven by diffusion. Due to the random nature of diffusivity, it is commonly calculated using a central differencing scheme.

2.2.3 Discretization in Time

Since the flow is unsteady, the equations must also be discretized in time. A second-order implicit scheme was chosen. Implicit methods are stable for every time step size and are, unlike explicit ones, capable of resolving transient incompressible flows. The second-order discretization in time is expressed in the following update equation in Ansys Fluent:

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} = R(u) \quad (2.19)$$

u is a scalar quantity of the flow. The exponents denote the time level and $R(u)$ the residual. This equation is solved iteratively at every time step until the residual is reduced below a preset threshold.

2.2.4 Gradient Reconstruction

A WLSQ method was employed to reconstruct the gradients of the flow properties. The scheme is based on the principle of least-squares minimization. The state variable ϕ is examined at the node \mathbf{P} and using its neighboring nodes \mathbf{N}_f with coordinates (x_0, y_0, z_0) and (x_f, y_f, z_f) , respectively. For each neighboring point, the *Taylor expansion* at node \mathbf{P} is then constructed.

$$\phi(\mathbf{N}_f) - \phi(\mathbf{P}) = \nabla\phi(\mathbf{P}) \cdot (\mathbf{N}_f - \mathbf{P}) + O(h^2) \quad (2.20)$$

The following notation is introduced:

$$\Delta x_i \equiv x_i - x_0 \quad (2.21)$$

$$\Delta y_i \equiv y_i - y_0 \quad (2.22)$$

$$\Delta z_i \equiv z_i - z_0 \quad (2.23)$$

$$\Delta\phi_i \equiv \phi(\mathbf{N}_i) - \phi(\mathbf{P}) \quad (2.24)$$

The higher-order terms $O(h^2)$ from equation 2.20 are then neglected, and the expansions are rearranged to

$$\begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \vdots \\ \Delta\phi_F \end{bmatrix} = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \Delta x_2 & \Delta y_2 & \Delta z_2 \\ \vdots & \vdots & \vdots \\ \Delta x_F & \Delta y_F & \Delta z_F \end{bmatrix} \cdot [\nabla^{ls}\phi(\mathbf{P})] \quad (2.25)$$

which forms a linear system of equations of the type

$$b = \mathbf{A}x \quad (2.26)$$

This system has no exact solution due to the absence of higher-order terms. It has more linearly independent equations than unknowns and is therefore overconstrained. Algebraically, the vector b does not lie in the column space of matrix \mathbf{A} . It is possible to minimize the error $e \equiv b - \mathbf{A}x$ by finding a solution in which its projection onto the column space of \mathbf{A} is zero. In this case, it is perpendicular to every column of \mathbf{A} .

$$\mathbf{A}^T(b - \mathbf{A}x) = 0 \quad (2.27)$$

This case minimizes the sum of the squared components of the error e . Least-squares gradient schemes can be augmented by a diagonal weighting matrix \mathbf{W} To take the different distances to the nodes into account. This weighting is vital if the cell contains skewed or stretched cells, for example, near a no-slip wall.

$$\mathbf{W} = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_f \end{bmatrix} \quad (2.28)$$

$$(2.29)$$

The solution to equation 2.25 after multiplying both sides with the weighting matrix is

$$x = \nabla^{ls}\phi(\mathbf{P}) = (\mathbf{A}^T \mathbf{W}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^T \mathbf{W} b \quad (2.30)$$

The weights making up the matrix \mathbf{W} are usually of the form $w_f = (\Delta r_f)^{-q}$ where r_f is the distance between nodes.

$$\Delta r_f = \|\mathbf{N}_f - \mathbf{P}\| \quad (2.31)$$

The user can select the exponent q at his will. Common picks are $q = 1$ and $q = 2$. Syrakos et al. also noted advantages of using $q = 2/3$ for specific grid types [13].

2.3 Turbulence Model

In this case, the SST model was selected as a turbulence model.

The transport equation for the TKE k and the specific dissipation ω in the SST write as follows:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_{turb}) \frac{\partial k}{\partial x_j} \right] \quad (2.32)$$

$$\begin{aligned} \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} &= \frac{\gamma}{\nu_{turb}} \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_{turb}) \frac{\partial \omega}{\partial x_j} \right] \\ &+ 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{aligned} \quad (2.33)$$

with $\tau_{ij} = \mu_{turb} \left(2S_{ij} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}$ and $S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$. The blending function F_1 incorporates the cell centroid distance to the nearest wall d and is defined as $F_1 = \tanh(arg_1^4)$ where

$$arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right), \frac{4\rho \sigma_{\omega 2} k}{CD_{k\omega} d^2} \right] \quad (2.34)$$

$$CD_{k\omega} = \max \left(2\rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \right) \quad (2.35)$$

The empirical constants σ_k , σ_ω , and β are blended in accordance with the blending function F_1 .

$$\phi = F_1 \phi_1 + (1 - F_1) \phi_2 \quad (2.36)$$

Respective empirical values for ϕ_1 and ϕ_2 can be found in table 2.2. The constant β^* is assumed to be 0.09 in the SST model.

ϕ_1	σ_{k1}	0.85
	$\sigma_{\omega 1}$	0.65
	β_1	0.075
ϕ_2	σ_{k2}	1.00
	$\sigma_{\omega 2}$	0.856
	β_2	0.0828

Table 2.2: Empirical constants in the SST model

Furthermore, both the $k - \omega$ and the $k - \varepsilon$ model tend to overestimate the wall shear stress, which leads to inaccurate flow separation estimation. The SST model, therefore, introduces a shear stress limiter, which is implemented in the calculation of the eddy viscosity μ_{turb} .

$$\mu_{turb} = \frac{\rho a_1 k}{\max(a_1 \omega, SF_2)} \quad (2.37)$$

where a_1 is a SST closure constant and equal to 0.31. This calculation uses the blending function F_2 which is defined as

$$F_2 = \tanh(arg_2^4) \quad (2.38)$$

$$arg_2 = \max\left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega}\right) \quad (2.39)$$

3 Test Data Generation

The selected test case to generate detailed flow data is a lid-driven flow within a cubical cavity. The numerical simulation is discussed in detail in this section. After elaborating on simulation inputs like the lid geometry and flow properties, the simulation mesh is discussed. A brief overview of the results is shown, and an *a posteriori* mesh check is performed to verify that the chosen mesh refinement does not introduce errors in the simulation.

3.1 Simulation Inputs

The cavity was designed with a spanwise aspect ratio of 1:1. Its depth and width are 150 mm, each. An isometric view of the cavity is shown in figure 3.1.

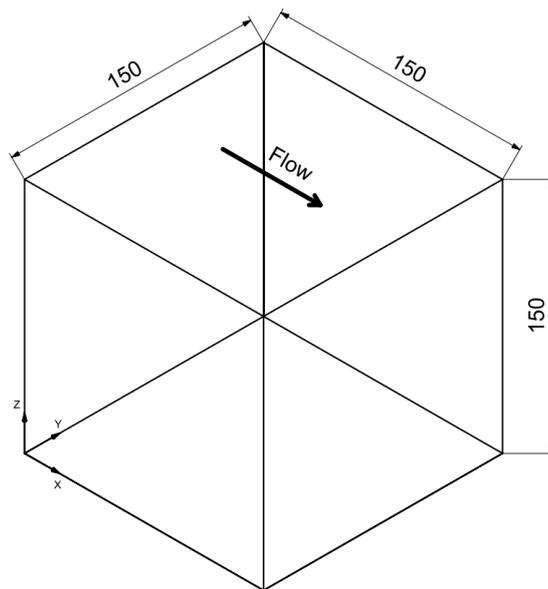


Figure 3.1: Isometric view of the simulated cavity

Experiments by Koseff et al. were conducted for this geometry using isothermal water flow [26]. Within the scope of this simulation, water is treated as an incompressible fluid. Simulation input parameters can be found in table 3.1.

Temperature [K]	293.15
Pressure [Pa]	101 325
Density [kg/m^3]	998.2071
Dynamic viscosity [Pa · s]	0.001002

Table 3.1: Input parameters of the fluid simulation

Many lid-driven cavity flow simulations choose a constant flow velocity in x-direction when defining the Dirichlet boundary conditions imposed on the cavity. Leriche et al. note that this is physically inconceivable and propose a polynomial velocity distribution of degree 36 for the u -velocity instead [27]. This distribution was selected based on a series of 2-dimensional flow simulations conducted prior. The flow simulation performed within this project's scope uses an identical flow velocity distribution to enable a direct comparison of results. It is given by

$$u(x,y) = u_0 \left(1 - \left(\frac{2x}{h} - 1\right)^n\right)^2 \left(1 - \left(\frac{2y}{h} - 1\right)^n\right)^2 \quad (3.1)$$

where $n = 18$ for a polynomial distribution of degree 36, $h = 150$ mm and u_0 is the maximum flow velocity. The average flow velocity on top of the lid can be calculated by integrating the function and dividing by the total lid area:

$$\bar{u} = \frac{1}{(0.15)^2} \int_0^{0.15} \int_0^{0.15} u_0 \left(1 - \left(\frac{2x}{0.15} - 1\right)^{18}\right)^2 \left(1 - \left(\frac{2y}{0.15} - 1\right)^{18}\right)^2 dx dy \approx 0.8496 \cdot u_0 \quad (3.2)$$

The flow simulation was conducted for two different maximum Reynolds numbers at the lid, $Re_{max} = 3200$ and $Re_{max} = 12000$. These two simulations represent an essentially laminar flow regime and a turbulent one. Therefore, the average Reynolds numbers on top of the cavity are approximately 2718 and 10196, respectively. The maximum velocities in x-direction imposed on the cavity are

$$u_{01} = \frac{Re_{max1} \cdot \mu}{h \cdot \rho} \approx 0.021414 \text{ m/s} \quad (3.3)$$

$$u_{02} = \frac{Re_{max2} \cdot \mu}{h \cdot \rho} \approx 0.080304 \text{ m/s} \quad (3.4)$$

and the average velocities in the x-direction are

$$\bar{u}_1 = \frac{Re \cdot \mu}{h \cdot \rho} \approx 0.018189 \text{ m/s} \quad (3.5)$$

$$\bar{u}_2 = \frac{Re \cdot \mu}{h \cdot \rho} \approx 0.068230 \text{ m/s} \quad (3.6)$$

The u -velocity distribution on top of the cavity for the laminar flow simulation is shown in figure 3.2.

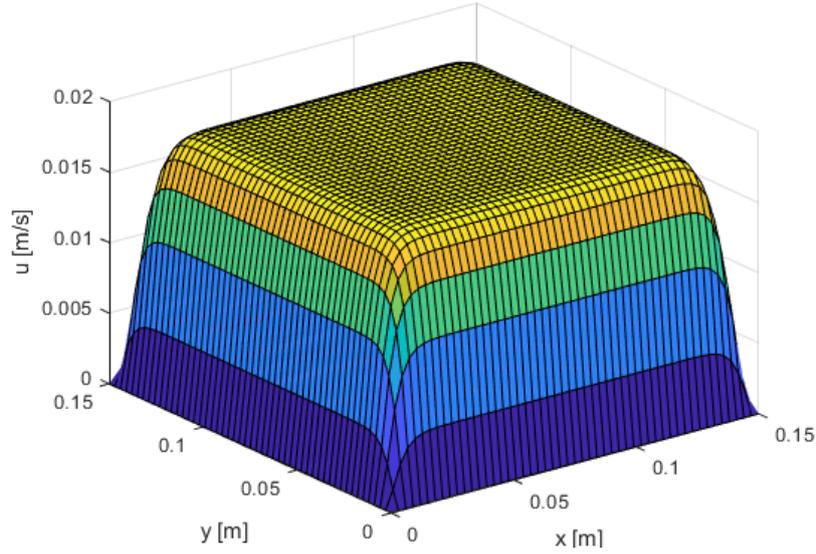


Figure 3.2: Imposed velocity distribution on top of the cavity

The test cases were simulated for ten seconds. Only five seconds were used to generate flow data for training the ROM. The remaining five seconds were used for validation of the ROM data results. To iterate through time, a two-step iteration scheme was chosen. It allows for a relatively high accuracy even when using a large outer step size. A Δt of 0.05 s was selected. The maximum number of inner iterations per time step was set to 15.

3.2 Mesh

To accurately simulate viscous flows near a wall, it is necessary to account for the *boundary layer*. It increases the production of turbulence and dampens wall-normal flow velocities. Research has shown that the boundary layer can be divided into three sections [51]. It is common to describe the boundary layer in terms of y^+ . This parameter denotes the wall-normal distance normalized using the friction velocity u_τ . It is defined as

$$y^+ = y \cdot u_\tau / \nu = y \cdot u_\tau \cdot \rho / \mu \quad (3.7)$$

where $u_\tau = \sqrt{\tau_w / \rho}$. The wall shear stress τ_w is defined as $\tau_w = \frac{1}{2} \rho u^2 c_f$. Near the wall, at , the friction velocity is dominated by viscous flow effects. This viscous sub-layer occurs at distances around $0 < y^+ < 5$. At those distances, the normalized velocity profile in x-direction $u^+ = u / u_\tau$ can be approximated by a linear function. In the so-called inertial sub-layer, turbulent effects dominate the flow behavior. It is located at distances of $20 < y^+ < 200$. The

velocity u^+ can be approximated by a logarithmic function. This can also be seen in figure 3.3.

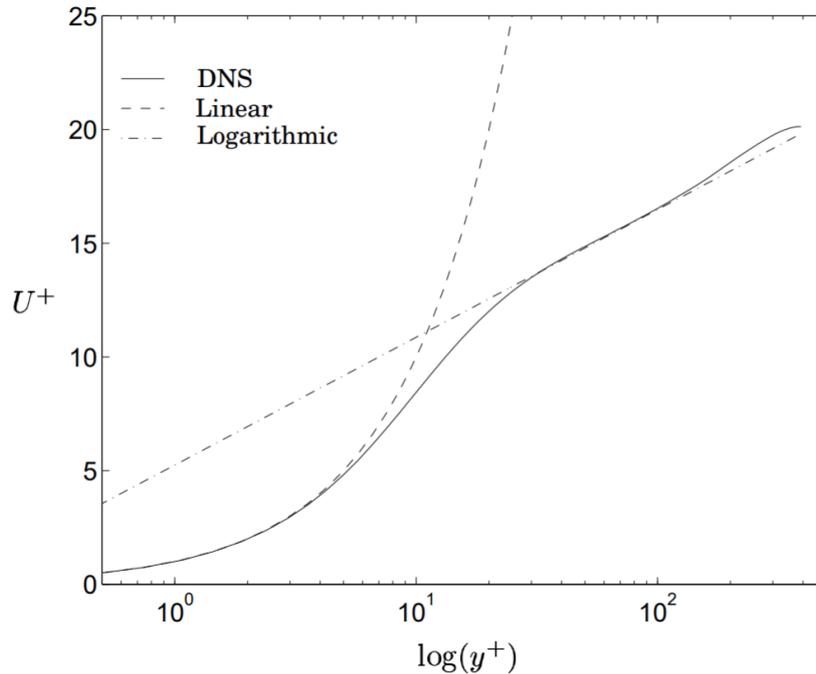


Figure 3.3: U-velocity profile with linear and log-law approximation. Channel flow, DNS-data, $Re_\tau=395$ [1]

The buffer layer is between those two sub-layers, where both viscous and turbulent effects influence the flow behavior. In this layer, the highest amount of turbulence is produced. Because different turbulence models utilize additional turbulence production terms and the flow behavior inside the buffer layer is generally challenging to model, the centroid of the mesh cell closest to the wall should not be placed within this region. Instead, the following approaches are used:

- Placing the wall-nearest cell centroid in the viscous sub-layer and incorporate a low-Reynolds-number model (wall-resolved approach)
- Placing the wall-nearest cell centroid in the inertial layer and use a high-Reynolds-number model (wall function approach)

Because wall-resolved flow simulations typically become impractical at high Reynolds numbers, these flows then require a wall function approach. In this case, a wall-resolved approach, often called the integration method, was selected. It is to note that the sub-layer ranges are given for flow over a flat plate and can vary due to pressure gradients, wall curvature, or flow separation. Therefore, it is common practice to select a y^+ value of 1 when generating a mesh for a wall-resolved simulation¹. To calculate an initial estimate for the skin friction coefficient

¹<https://www.cfd-online.com/Forums/main/111896-why-y-1-a.html>

c_f , an equation proposed by Schlichting [23] was used. It is based on empirical data and relates c_f to the Reynolds number of a fully turbulent flow over a flat plate.

$$c_f = (2 \cdot \log_{10}(Re) - 6.5)^{-2.3} \quad (3.8)$$

A wall-function approach with an initial value of $y^+ = 100$ was selected. The height of the first mesh cell height at a wall was calculated using the previously introduced relationships, the input data from 3.1, and a characteristic length of 150 mm:

$$c_f \approx 4.6990 \quad (3.9)$$

$$\tau_w \approx 1.0755 \text{ N/m}^2 \quad (3.10)$$

$$u_\tau \approx 0.03282 \text{ m/s} \quad (3.11)$$

$$y_{cc1} = \frac{y^+ \mu}{u_\tau \rho} \approx 3.0581 \cdot 10^{-3} \text{ m} \quad (3.12)$$

The centroid is located at half the cell height for a hexahedral mesh cell. The resulting in an initial cell height is therefore

$$y_{c1} = 2 * y_{cc1} \approx 6.11626 \cdot 10^{-3} \text{ m} \quad (3.13)$$

For the turbulent flow case, this process is analogous and results in the following geometry value:

$$c_{f2} \approx 0.3124 \quad (3.14)$$

$$\tau_{w2} \approx 1.0055 \text{ N/m}^2 \quad (3.15)$$

$$u_{\tau2} \approx 0.031739 \text{ m/s} \quad (3.16)$$

$$y_{cc2} = \frac{y^+ \mu}{u_{\tau2} \rho} \approx 3.1627 \cdot 10^{-3} \text{ m} \quad (3.17)$$

$$y_{c2} = 2 * y_{cc2} \approx 6.3254 \cdot 10^{-3} \text{ m} \quad (3.18)$$

Its maximum thickness δ is calculated to ensure that the mesh can resolve the boundary layer accurately. For the laminar flow, this value can be approximated using a solution derived from the Blasius equations, as described by Schetz and Bowersox [22].

$$\delta_1 \approx 5.0 \frac{x}{\sqrt{Re}} = 5.0 \frac{0.15 \text{ m}}{\sqrt{3200}} \approx 0.013258 \text{ m} \quad (3.19)$$

For the turbulent case, an approximation for the boundary layer thickness over a flat plate in turbulent flow was employed. This relationship is given by Schlichting [23].

$$\delta_2 \approx 0.37 \frac{x}{Re^{1/5}} = 0.37 \frac{0.15 \text{ m}}{12000^{1/5}} \approx 0.008481 \text{ m} \quad (3.20)$$

$n = 7$ layers were placed within the boundary layer for both of these meshes. The growth rate GR can be calculated using the boundary layer thickness and the first cell height, the growth rate GR can be calculated.

$$GR_1 = \left(\frac{\delta_1}{y_{c1}} \right)^{\frac{1}{n}} \approx 1.11680 \quad (3.21)$$

$$GR_2 = \left(\frac{\delta_2}{y_{c2}} \right)^{\frac{1}{n}} \approx 1.04278 \quad (3.22)$$

A mesh with a regular cell size of 8 mm was selected for both flow cases. The mesh for the laminar flow case can be seen in figure 3.4. Two cylindrical refinement zones near the top side of the cavity were added. Both zones extend to a radius of 75 mm and were placed normal to the main flow direction. Within the refinement zones, a standard cell height of 0.0025 mm was defined. These zones were added to increase the accuracy of the simulation since the largest pressure gradients are expected within these domains. The resulting mesh consists of 149552 hexahedral cells spanning between 156392 nodes.

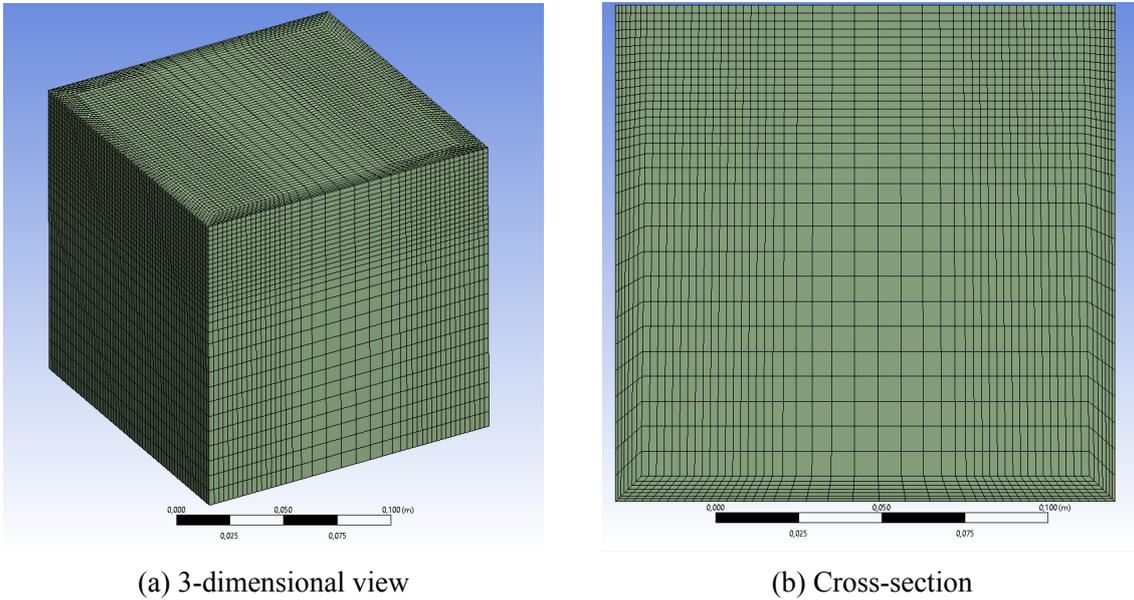


Figure 3.4: Mesh for the laminar flow case

The mesh for the turbulent flow case is shown in figure 3.5. It is similar to the laminar mesh, but features a thinner boundary layer and lower wall-adjacent cell height.

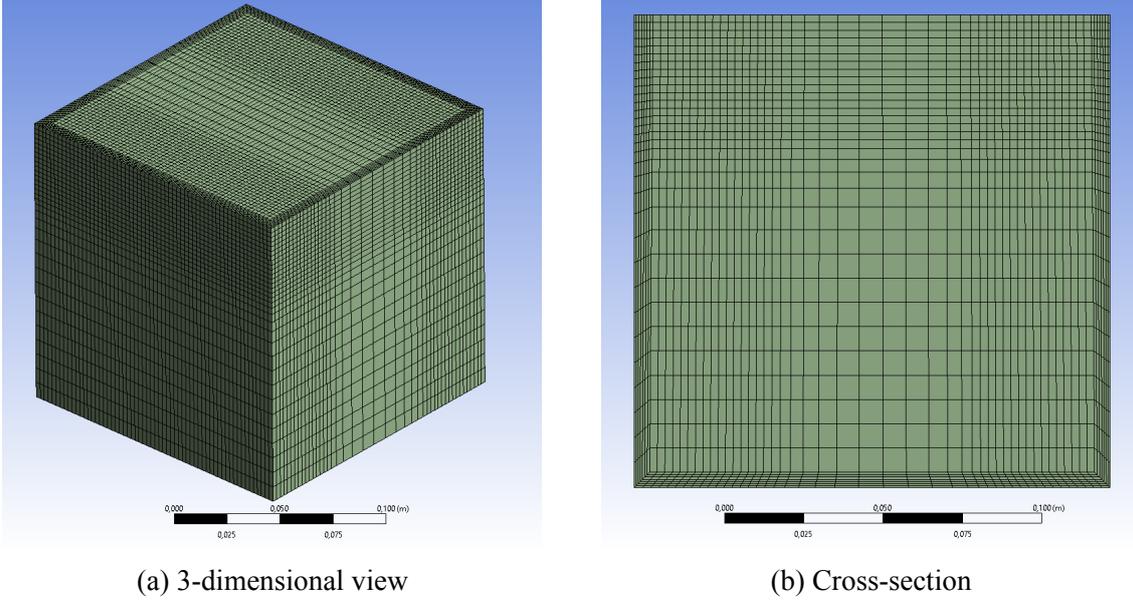


Figure 3.5: Mesh for the turbulent flow case

3.3 Wall Function

The adaptive wall function was selected for the simulation. It has been validated by Carlson et al. [52] by comparison with theoretical and experimental data and produced very accurate results. The method by Knopp et al. [25] is based on the work of Reichardt [53] and Spalding [54]. It uses a hyperbolic tangent function to blend the *log-law* equation for the friction velocity and the *law-of-the-wall* equation by Reichardt. The former can be written as

$$u_{\tau}|_{Log} = \frac{1}{\kappa} \ln(y^+) + 5.1 \quad (3.23)$$

Herein, κ denotes the von Kármán constant, here set to 0.41. The latter equation takes the form of

$$u_{\tau}|_{Reichardt} = \frac{\ln(1 + 0.4y^+)}{\kappa} + 7.8 \left(1 - e^{-y^+/11} - \frac{y^+}{11} e^{-y^+/3} \right) \quad (3.24)$$

Using the blending function ϕ_{b1} , the modified Reichardt equation writes as follows:

$$u_{\tau}|_{Reichardt,m} = (1 - \phi_{b1})u_{\tau}|_{Reichardt} + \phi_{b1}u_{\tau}|_{Log} \quad (3.25)$$

with

$$\phi_{b1} = \tanh(\arg^4) \quad (3.26)$$

$$\arg = \frac{y^+}{27} \quad (3.27)$$

In a second step, this function is then blended with the friction velocity function by Spalding in a similar fashion. For the SST turbulence model, this primarily affects the results in the buffer layer.

$$u_\tau|_{Spalding} = u^+ + e^{-\kappa 5.2} \left(e^{\kappa u^+} - 1 - \kappa u^+ - \frac{(\kappa u^+)^2}{2} - \frac{(\kappa u^+)^3}{6} \right) \quad (3.28)$$

$$u_\tau|_{SST} = (1 - \phi_{SST})u_\tau|_{Spalding} + \phi_{SST}u_\tau|_{Reichardt,m} \quad (3.29)$$

$$\phi_{SST} = \tanh(arg^2) \quad (3.30)$$

$$arg = \frac{y^+}{50} \quad (3.31)$$

Knopp et al. also propose an alternative calculation method for the, the specific dissipation rate ω near the wall:

$$\omega = \phi \omega_{b1} + (1 - \phi) \omega_{b2} \quad (3.32)$$

$$\phi = \tanh(arg^4), \quad arg = \frac{y^+}{10} \quad (3.33)$$

$$\omega_{b1} = \omega_{vis} + \omega_{log}, \quad \omega_{b2} = \left(\omega_{vis}^{1.2} + \omega_{log}^{1.2} \right)^{1/1.2} \quad (3.34)$$

$$\omega_{vis} = \frac{6\nu}{\beta_1 y^2}, \quad \omega_{log} = \frac{u_\tau}{\sqrt{\beta^* \kappa y}} \quad (3.35)$$

The parameters β_1 and β^* can be found in section 2.3.

3.4 Simulation Results

The simulation results for the laminar and the turbulent flow case are detailed in the following subsections.

3.4.1 Laminar Flow Case

The laminar, lid-driven cavity simulation resulted in the following findings: The streamlines, plotted at the central cross-section in figure 3.6a, confirmed the expected flow behavior. A large-scale circulation flow forms in the cavity. The plot of the flow direction vector field, shown in figure 3.6b, reveals two low-velocity re-circulation zones, or secondary eddies, at the bottom of the lid.

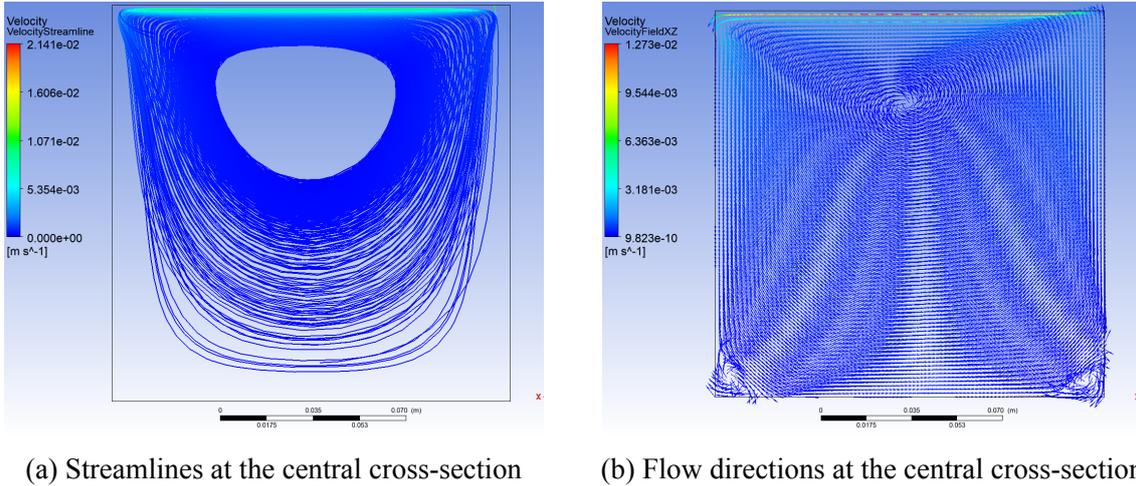
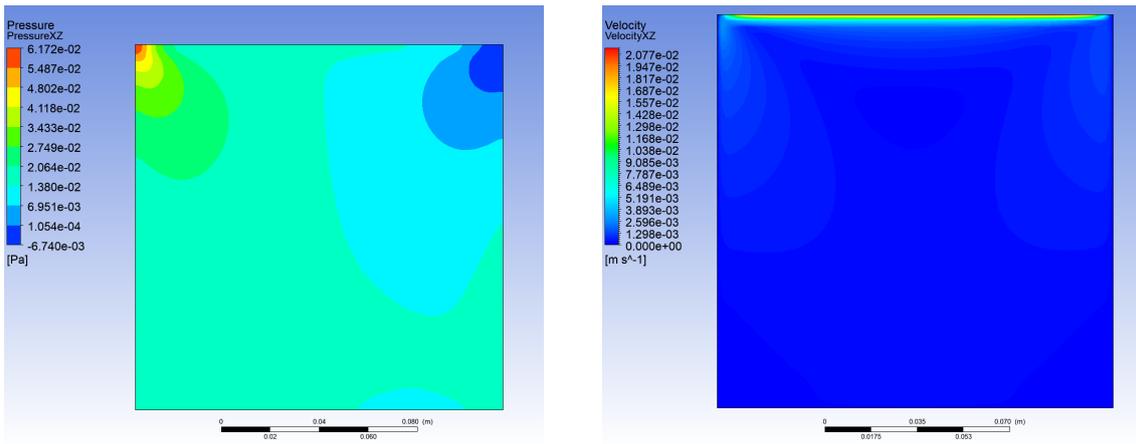


Figure 3.6: Cavity flow at $\Delta t = 5$ for the laminar flow case

The pressure and velocity profiles at the central cross-section at $\Delta t = 5$ are plotted in figure 3.7. The minor pressure variations have their maximums near the upper x-normal edges of the cavity. The plotted pressure is the gauge pressure, i.e. the pressure zero-referenced by the normal pressure of 101 325 Pa. The maximum flow velocities can, as expected, be found in the main flow near the lid in the x-direction. Figure 3.7a shows how only a very minor portion of the lid flow momentum is transported down into the cavity in the laminar flow case. The pressure profile is plotted at different cross-sections along the x-axis is plotted in figure 3.8. The pressure profiles within the boundary layers display smoother contours due to the increased mesh resolution. The figure confirms the pressure profile indicated by figure 3.7b and visualizes the 3-dimensional nature of these pressure peak zones, as well as a certain symmetry within the cavity.

In general, the system behaves very laminar. The initial TKE present in the cavity is transported outwards and quickly dissipates at the walls of the cavity, as shown in figures 3.10a and 3.10b, respectively. The TKE remaining after five seconds is located at the center of the cavity, see figure 3.10a.



(a) Pressure profile at the central cross-section (b) Velocity profile at the central cross-section

Figure 3.7: Flow quantities at $\Delta t = 5$ for the laminar flow case

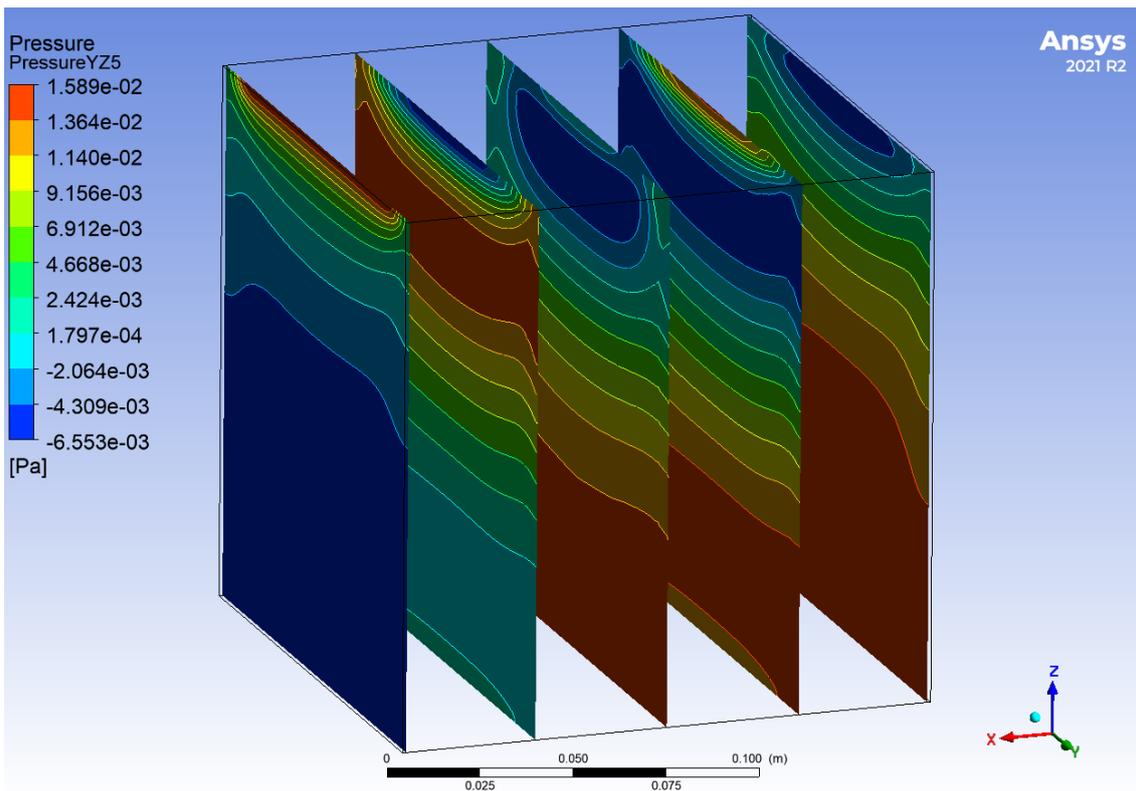


Figure 3.8: Pressure contours for the laminar flow case at different positions

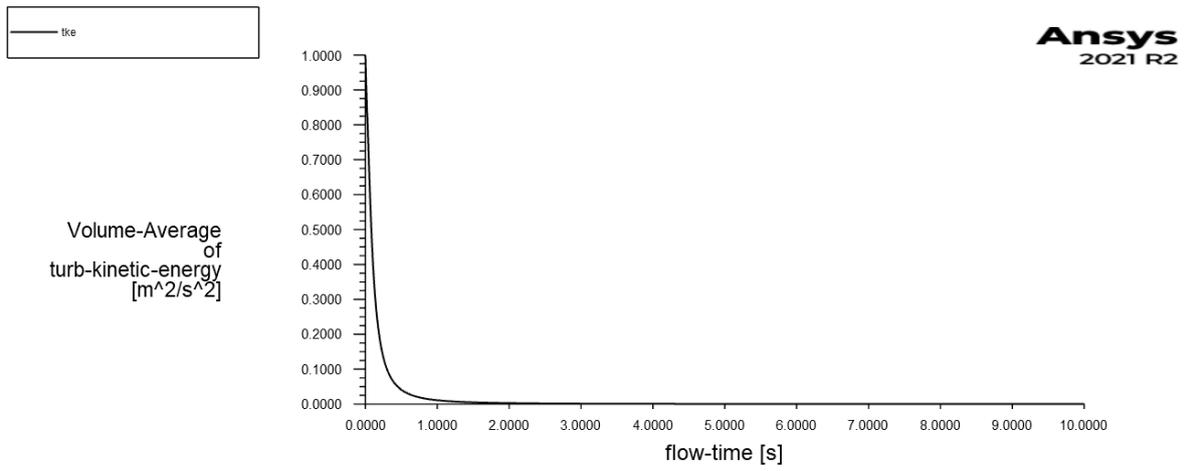


Figure 3.9: Volume-averaged turbulent kinetic energy over time for the laminar flow case

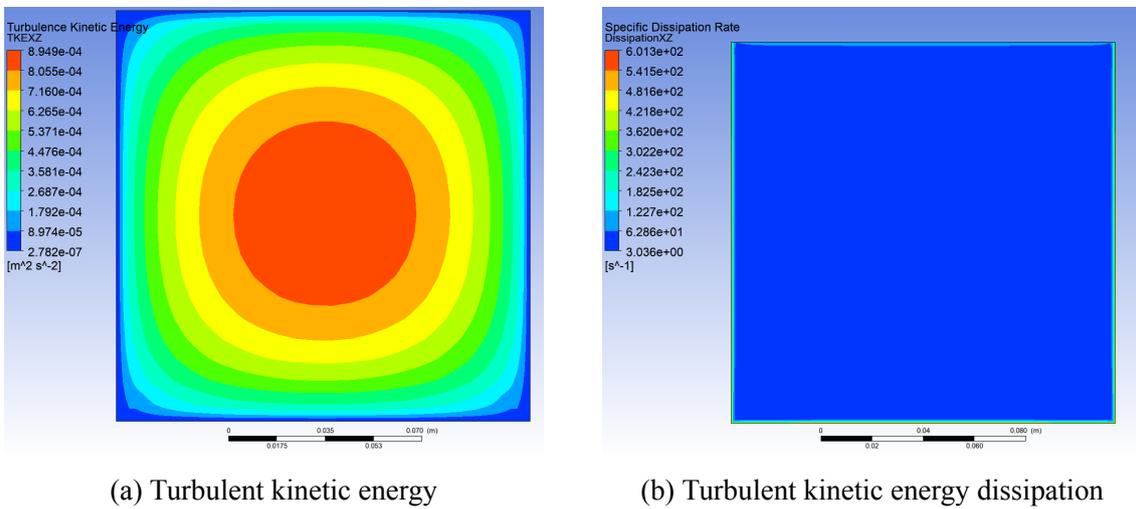


Figure 3.10: Flow quantities at the central cross-section in y-direction for the laminar flow case

3.4.2 Turbulent Flow Case

The turbulent, lid-driven cavity simulation produced the following results: The streamlines confirm that the large-scale circulation flow, which appeared in the laminar flow case, is still present. They are plotted at the central cross-section in figure 3.11 a. The center of this cavity-dominating vortex is moved downstream in the turbulent flow case. The plot of the flow direction vector field, shown in figure 3.11 b, supports this claim and reveals the presence of two secondary eddies at the bottom of the lid. These eddies are significantly smaller than the ones occurring in the laminar flow case.

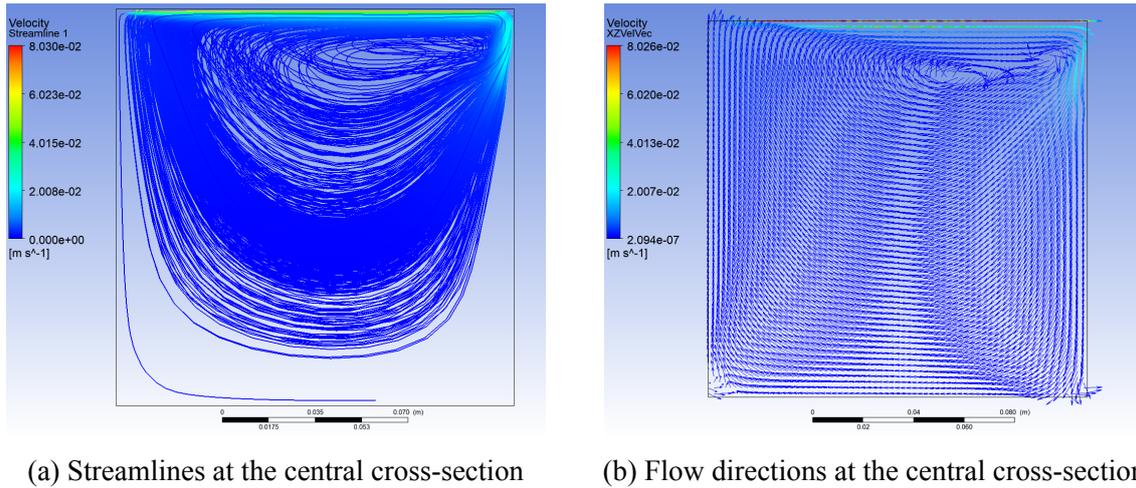
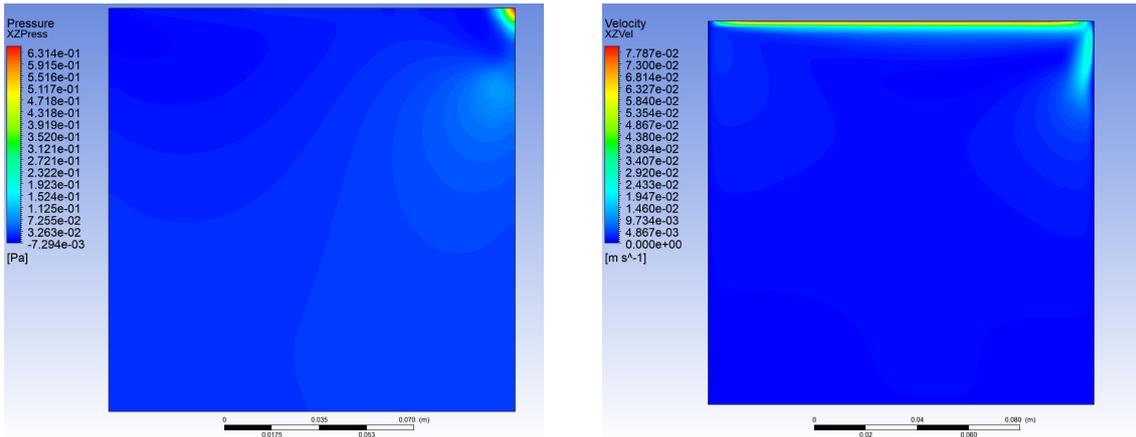


Figure 3.11: Cavity flow at $\Delta t = 5$ for the turbulent flow case

The contour plot of the gauge pressures present at the central cross-section in y-direction shows the formation of a high-pressure region near the lid's trailing edge. It is shown in figure 3.12b. The peak pressure is one order of magnitude above the peak pressure calculated for the laminar flow case. Besides this region, the cavity is subject to smaller variations in pressure. At a lower height, near the backside of the cavity in the x-direction, another pressure peak can be detected. This maximum is located below a zone of negative gauge pressure. Figure 3.12a shows the velocity distribution at the central cross-section. It is noticeable that higher flow velocities are present in the cavity compared to the laminar flow case. The momentum introduced into the flow via the lid is transported downwards in a high-velocity jet. The jet travels through the region of negative gauge pressure and separates from the cavity wall. It is dispersed at the height of the lower pressure peak.



(a) Pressure contours at the central cross-section (b) Velocity contours at the central cross-section

Figure 3.12: Flow quantities at $\Delta t = 5$ for the turbulent flow case

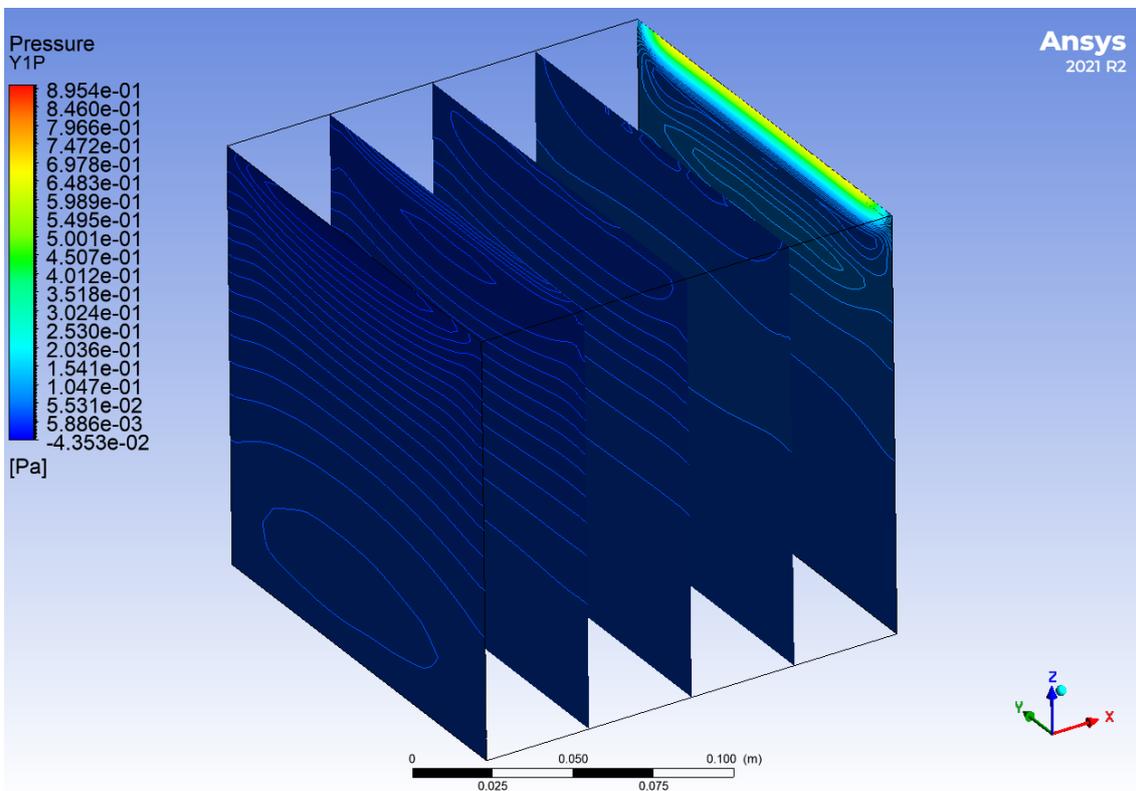
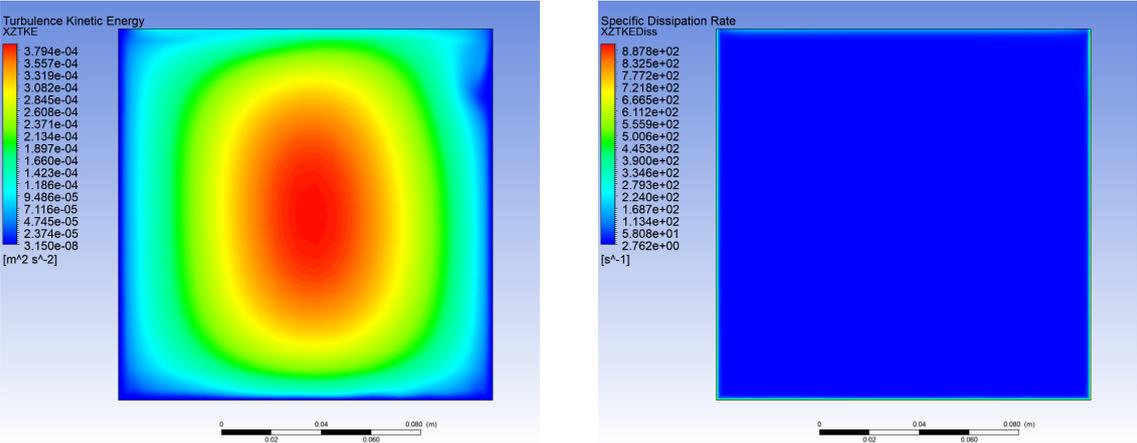


Figure 3.13: Pressure contours for the turbulent flow case at different positions

Figure 3.13 shows pressure contour plots at different cross-sections within the cavity. It indicates symmetry in the y-direction and shows the extent of the pressure peak near the trailing wall and the zone of negative gauge pressure above it.

Similar to the laminar flow case, the highest amount of TKE is present in the center of the cavity. This distribution can be seen in figure 3.14a. Additionally, an increased concentration of TKE can be found near the leading edge of the lid. The energy is low where the downwards high-velocity jet separates from the cavity wall. Figure 3.14b shows how the TKE dissipates due to the no-slip conditions at the cavity walls. This dissipation causes the TKE present in the cavity to be lowest in near-wall regions. Additional dissipation is present close to the constant velocity lid-flow.



(a) Turbulent kinetic energy

(b) Turbulent kinetic energy dissipation

Figure 3.14: Flow quantities at the central cross-section in y-direction for the turbulent flow case

4 Reduced-Order Model

A reduced-order model can be built using the information generated in the so-called *offline stage*, calculated using a finite-volume method within this project. The following chapter details the basis selected for this ROM. After describing an alternative to the direct POD basis calculation, the Galerkin projection is applied to the Navier-Stokes equations.

4.1 Proper Orthogonal Decomposition

Several different techniques are available to construct a ROM basis, for example, dynamic mode decomposition, eigenfunctions, or the POD. The latter approach was selected for this project because it can capture the highest amount of TKE using the fewest number of modes possible. The decomposition assumes that flow behavior can be separated into a sum of products of spatial functions and temporal coefficients. For a flow quantity q , this can be written as

$$q(x,t) \approx q_r(x,t) = \sum_{k=1}^{\infty} \Phi_k(x) a_k(t) \quad (4.1)$$

where Φ denotes the spatial POD modes. The word *proper* in POD implies that the maximum amount of energy of the system is captured by the first n modes. The *orthogonal* indicates that the POD modes are orthogonal to each other, meaning that the integral with respect to space of the product of modes n and m is 1 if $n = m$ and 0 in every other case.

Snapshots of data at different points in time are used to calculate the POD modes of a system. These snapshots have to be selected in observance of the system's timescales that need to be resolved. Sub-scale behavior can not be captured by the POD model. For N_s snapshots, the snapshot data can be written as

$$q_n \equiv q(x, t_n), \quad n = 1, \dots, N_s \quad (4.2)$$

After selecting appropriate snapshots, the data is rearranged into matrix U . For this, the vectors of snapshot information are concatenated into a row vector. The resulting snapshot vectors are then stacked on top of each other. If m data points are observed at n points in time, matrix U is of size $n \times m$. The dominant modes of this system can then be found by calculating the *covariance matrix* C , defined as

$$C = U^T U \quad (4.3)$$

It can be seen that the size of the *covariance matrix* is of size $m \times m$. Each diagonal matrix element of C denotes the data variance at the respective data point while the off-diagonal elements quantify the statistical correlation ρ between two points. For data points i and j , it is defined as $\rho = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}}$. A ρ -value of 1 implies a perfect correlation between two points, while a value of 0 means there is no correlation at all. A value of negative 1 indicates a perfect anti-correlation. Since the matrix C is produced by multiplying a matrix and its transpose, it is naturally symmetric. The dominant modes of the *covariance matrix* are those that maximize the variance values of each axis, written on the matrix diagonal. Since the modes are orthogonal to each other, there is no correlation. Every off-diagonal element is, therefore, zero.

It is known from linear algebra that the eigenvectors of a symmetric matrix form an orthogonal basis in which said matrix can be diagonalized:

$$C = \Xi \Lambda \Xi^{-1} \quad (4.4)$$

where Λ is a diagonal matrix with the eigenvalues of C as its entries and

$$\Xi = [\xi_1, \xi_2, \dots, \xi_n] \quad (4.5)$$

The eigenvectors of the correlation matrix are denoted with ξ_i while n is its dimensions. The eigenvalues indicate the total energy represented by the corresponding eigenvector. The further left eigenvectors are stored in the eigenvector matrix Ξ , the more dominant they are. This property is caused by assorting the eigenvalues by value. The POD basis function can then be constructed as follows:

$$\phi_i(x) = \sum_{n=1}^{N_s} \xi_{i,n}, \quad i = 1, \dots, N_r \quad (4.6)$$

N_r denotes the number of modes used to create the ROM.

4.2 Basis Calculation

Since the size of the *correlation matrix* is $m \times m$, the eigenvalue calculation becomes infeasible for large spatial domains. The Singular Value Decomposition (SVD) can be used to mitigate this problem. It factorizes the snapshot matrix U into

$$U = L \Sigma R^T \quad (4.7)$$

Calculating the *correlation matrix* using this equality leads to

$$U^T U = (L \Sigma R^T)^T (L \Sigma R^T) \quad (4.8)$$

$$= (R \Sigma^T L^T) (L \Sigma R^T) \quad (4.9)$$

$$= (R \Sigma^T \Sigma R^T) \quad (4.10)$$

When comparing this relationship to equation 4.4, it can be seen that $\Lambda = \Sigma^T \Sigma$, the diagonal matrix of the system's eigenvalues. $\Xi = R$, a matrix composed of the system's eigenvectors. The temporal correlation can be calculated as follows:

$$C_s = U U^T = (L \Sigma R^T) (L \Sigma R^T)^T \quad (4.11)$$

$$= (L^T \Sigma R) (R \Sigma^T L^T) \quad (4.12)$$

$$= (L \Sigma \Sigma^T L^T) \quad (4.13)$$

Where L is the eigenvector matrix. It can be shown that the non-zero elements of the diagonal matrices $\Sigma^T \Sigma$ and $\Sigma \Sigma^T$ are identical. The eigenvectors of the spatial *correlation matrix* can therefore be calculated using the eigenvalues and vectors of the temporal *correlation matrix*.

$$R^T = \Sigma^{-1} L^{-1} U \quad (4.14)$$

Since the eigenvectors are orthogonal to each other, $L^{-1} = L^T$. The inverse of the Σ is a diagonal matrix with the reciprocal entries of Σ .

$$R = (\Sigma^{-1} L^T U)^T \quad (4.15)$$

An equivalent POD basis to 4.6 can therefore be found by

$$\phi_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{n=1}^{N_s} \zeta_{i,n} u_n(x), \quad i = 1, \dots, N_r \quad (4.16)$$

The eigenvectors of the temporal correlation matrix are written as ζ_i . The n th row of the snapshot matrix is denoted by u_n . Since the number of snapshots N_s is often far smaller than the number of data points in the domain for typical flow calculations, this technique can notably accelerate the POD basis calculation.

4.3 Galerkin-Projection

To predict the system's behavior, equation 4.1 is inserted into the incompressible Navier-Stokes equations. Since a is only a function of t and ϕ is only a function of x , the following system results:

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = 0 \quad (4.17)$$

$$\sum_{i=1}^{N_r} \phi_i a_{i_t} = \nu\Delta \sum_{i=1}^{N_r} \phi_i a_i - \left(\sum_{i=1}^{N_r} \phi_i a_i \cdot \nabla \right) \sum_{i=1}^{N_r} \phi_i a_i - \nabla p \quad (4.18)$$

The number of modes used to construct the ROM is denoted by N_r . The system 4.17 is therefore approximated by the system 4.18. For every temporal coefficient a_i , the following linear system can be constructed:

$$\phi_i a_{i_t} = \nu\Delta\phi_i a_i - (\phi_i a_i \cdot \nabla) \sum_{j=1}^{N_r} \phi_j a_j - \nabla p \quad (4.19)$$

Since the POD modes are orthogonal to each other by design, meaning $\phi_i \phi_j = 0$ for every $i \neq j$, the system simplifies to

$$\phi_i a_{i_t} = \nu\Delta\phi_i a_i - (\phi_i a_i \cdot \nabla)\phi_i a_i - \nabla p \quad (4.20)$$

$$a_{i_t} = \frac{\phi_i^T}{|\phi_i^2|} \nu\Delta\phi_i a_i - \frac{\phi_i^T}{|\phi_i^2|} (\phi_i a_i \cdot \nabla)\phi_i a_i - \frac{\phi_i^T}{|\phi_i^2|} \nabla p \quad (4.21)$$

In most cases, the pressure term can be neglected [34–36], $\nabla p = 0$. Hence, to extrapolate the flow behavior into the future, the following initial value problem needs to be solved:

$$a_{i_t} = \dot{a}_i = \nu B_i a_i - a_i C_i a_i \quad (4.22)$$

$$B_i = \frac{\phi_i^T}{|\phi_i^2|} \Delta\phi_i \quad (4.23)$$

$$C_i = \frac{\phi_i^T}{|\phi_i^2|} (\phi_i \cdot \nabla)\phi_i \quad (4.24)$$

The flow velocity information at the last calculated time step of the *offline phase* is used to calculate an initial value for a . This data is denoted u_{tf} .

$$a0_i = \frac{\phi_i}{|\phi_i|^2} u_{tf} \quad (4.25)$$

5 Model Analysis

In the following chapter, the results of the POD-Galerkin method are discussed and compared to the numerically generated results.

The normalized eigenvalues corresponding to the POD modes are shown in figure 5.1a. It can be seen that the energy corresponding to the respective POD mode rapidly declines. The truncation error for the reconstruction is below 10^{-12} when 77 modes are used. Only 46 modes are needed to achieve a truncation error below 10^{-9} . Additional truncation errors for reconstructions using N_r modes are listed in table 5.1. The truncation error is plotted over the number of modes used for the reconstruction in figure 5.1b.

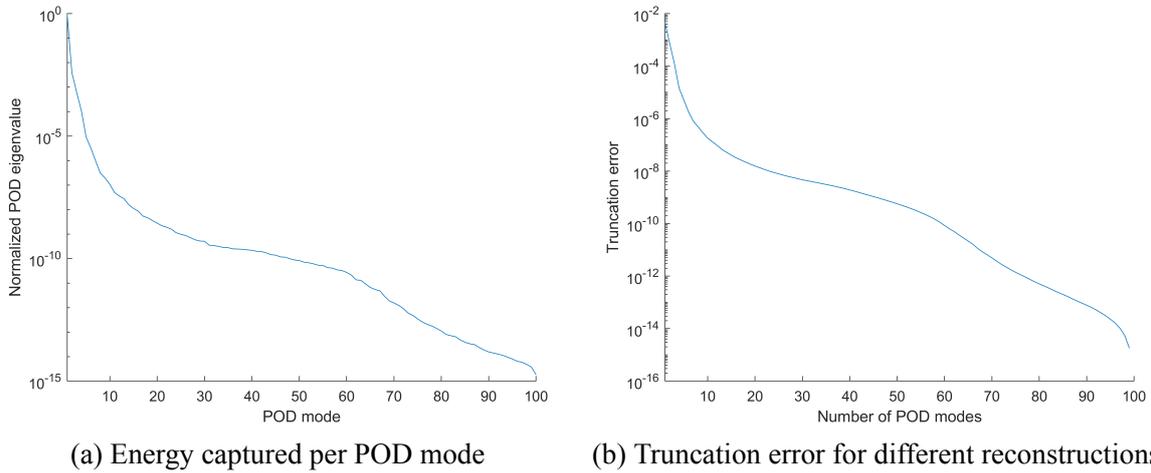


Figure 5.1: POD reconstruction properties of the laminar flow case

N_r	2	4	5	7	12	24
e_{trunc}	7.44e-04	1.43e-5	5.07e-6	7.98e-7	9.61e-8	8.88e-9
N_r	46	60	68	77	89	98
e_{trunc}	9.56e-10	8.54e-11	8.38e-12	9.32e-13	9.34e-14	5.57e-15

Table 5.1: Truncation errors for different reconstruction

The results for the POD-Galerkin method employed within this project's scope were calculated using the first 24 most dominant modes of the system. The truncation error should therefore be close to 10^{-8} . The ROM was run over a total period of five seconds. It was

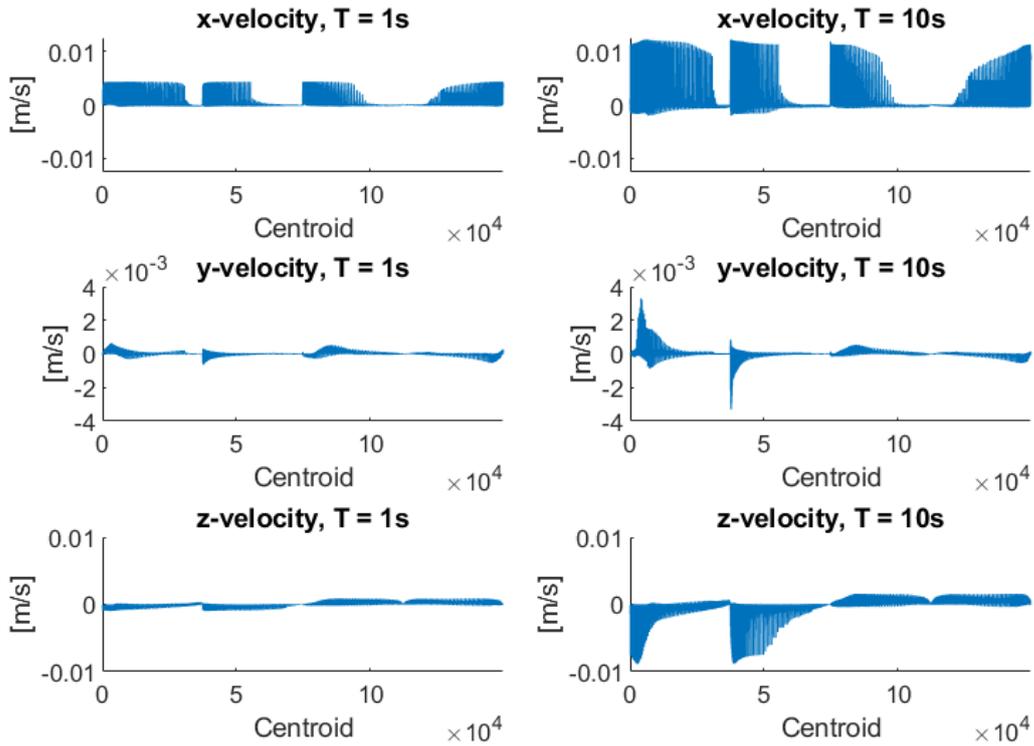


Figure 5.2: Numerical results at different time steps

constructed using 100 snapshots with intervals of 0.05 seconds between them. The selected time step size for the ROM was therefore 0.05 seconds, too. Since only information with this temporal resolution was used, the model fails to capture flow phenomena below that scale. The error hence increases, as shown by Carlberg et al. [42].

The numerical simulation shows that the velocity profiles in every direction become more pronounced with time. Figure 5.2 plots the velocities at every centroid in the domain for two different times. The graphs show that the velocities in x- and z-direction exceed the velocity in the y-direction by order of magnitude. Since the three left plots show the profiles at $t = 1s$, likely, the flow has not fully developed yet, and the result is highly dependent on the selected initialization. As time progresses, the dominant flows in the system become more distinct, and the simulation results become more robust.

An *online stage* of 5 seconds was compared against results of a numerical simulation to validate the ROM. Since the model was trained on data generated for 5 seconds of flow time, the temporal domain for the *online stage* is between 5 and 10 seconds. Numerical results for this stage are shown in figure 5.3. The figure plots the velocities in x-, y-, and z-direction for every centroid within the domain concatenated into one velocity profile. This growth of this velocity profile is visualized by overlapping the data generated at different steps in time.

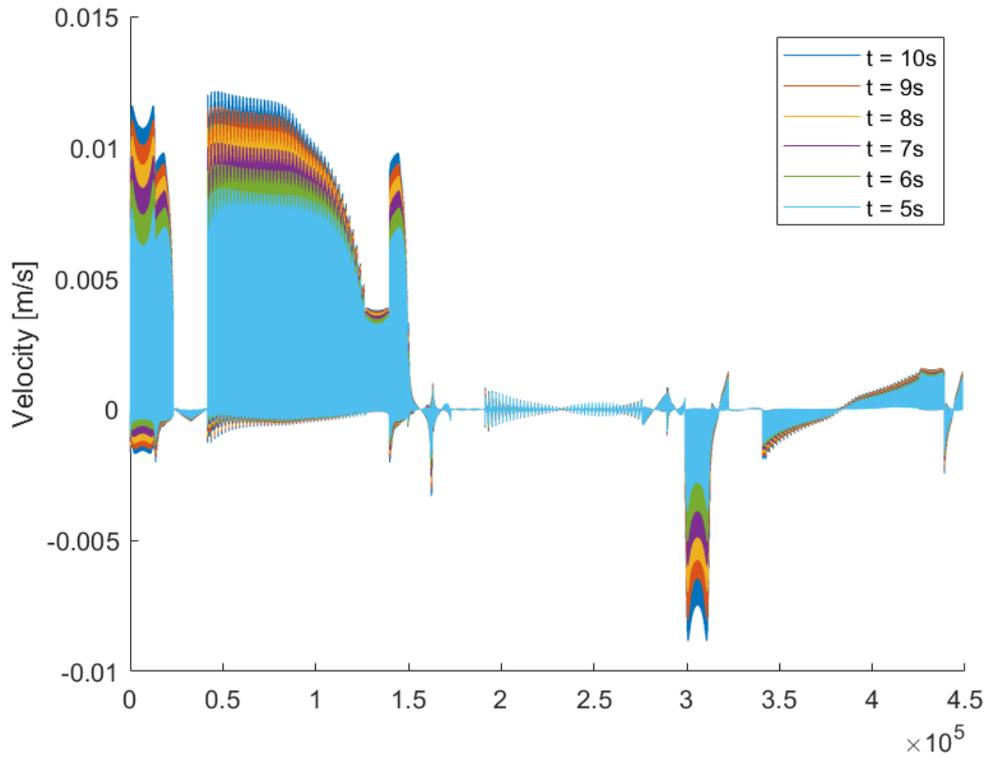
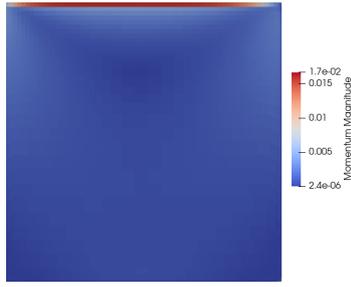
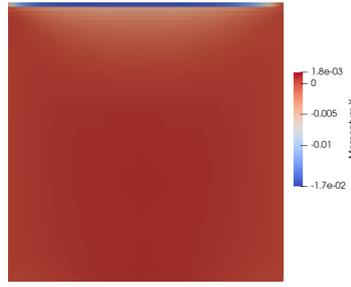


Figure 5.3: Cell velocity profile over time

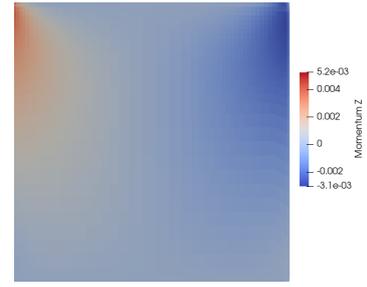
The velocity profiles generated by the first four dominant POD modes are shown in figure 5.4. The velocity in y -direction was not shown since its magnitude is far below the magnitudes of the x - and z -velocities. The first two modes seem to mostly model the flow behavior near the lid and the walls normal to the flow direction. The third mode captures the large circulation flow through the cavity. Since none of the first few dominant modes show the two reflow regions at the bottom of the cavity, the momentum carried by these flows can be estimated to be fairly insignificant. The less dominant a mode is, the smaller the scale of the flow behavior captured by it seems to be. The non-dominant POD modes capture small fluctuations in the flow to further approach the accuracy of the FOM. The model implemented within the scope of this project should reach a FOM accuracy at $N_r = 100$ if assumption 4.1 holds, since 100 snapshots in time were used to construct it.



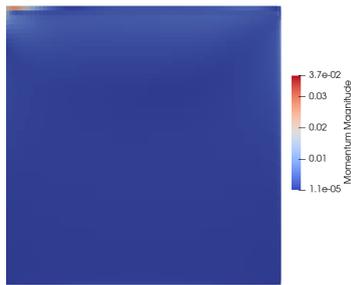
(a) Magnitudes of POD mode 1



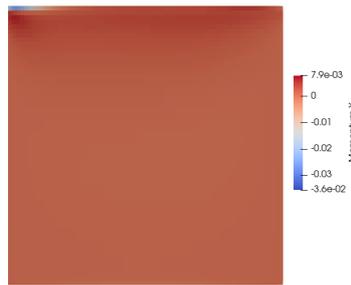
(b) x-Velocity of POD mode 1



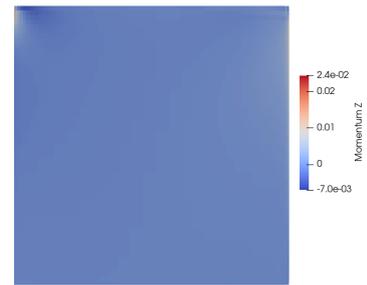
(c) z-Velocity of POD mode 1



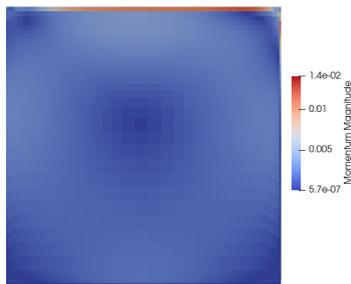
(d) Magnitudes of POD mode 2



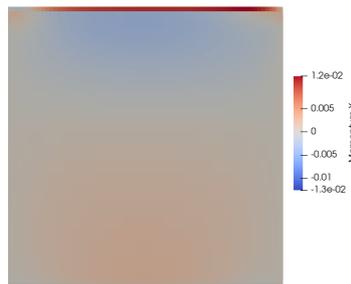
(e) x-Velocity of POD mode 2



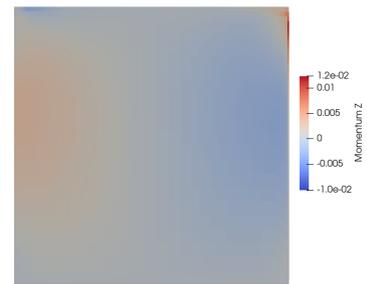
(f) z-Velocity of POD mode 2



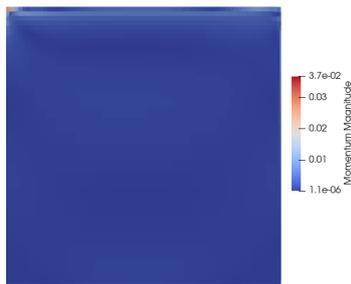
(g) Magnitudes of POD mode 3



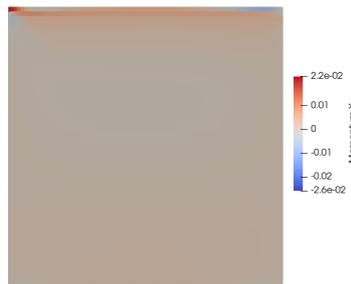
(h) x-Velocity of POD mode 3



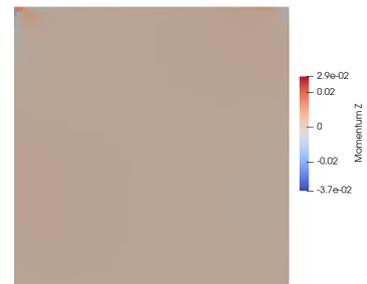
(i) z-Velocity of POD mode 3



(j) Magnitudes of POD mode 4



(k) x-Velocity of POD mode 4



(l) z-Velocity of POD mode 4

Figure 5.4: POD mode velocity profiles at the central cross section in XZ direction

Figure 5.5 shows the five a coefficients corresponding to the most dominant modes of the system plotted over 5 seconds. It can be seen that only the first coefficient experiences notable change. The profile of the coefficient suggests that the system will quickly become unstable. This suspicion is further reinforced by the plot of the Root Mean Square Error (RMSE) over time, shown in figure 5.7.

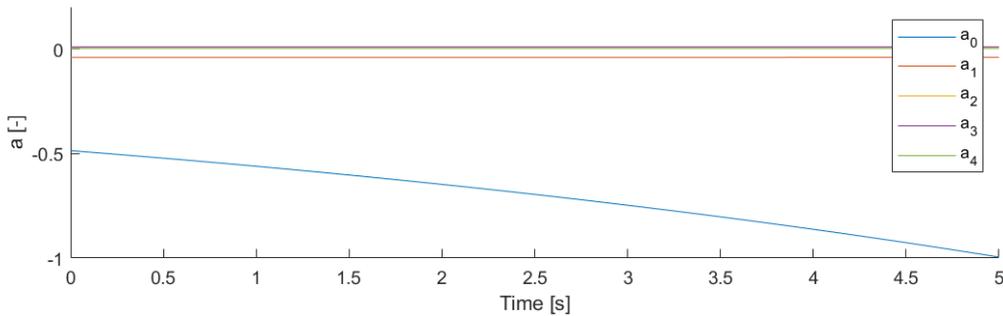


Figure 5.5: Temporal coefficients of the POD-Galerkin ROM

This deviation from the expected system behavior can also be seen in figure 5.6. When compared to figure 5.3, it can be seen that the Galerkin projection captured critical characteristics of the flow while quickly starting to deviate from the numerical solution.

The calculation time of the ROM using the POD-Galerkin method is significantly lower than performing a numerical simulation. While the FV method took hours to run on standard laptop hardware, the model was computed in less than four minutes. The majority of this time was dedicated to Input/Output. Furthermore, the program can be optimized to require only a fraction of its current runtime by removing output functions devoted to the validation and debugging of the program.

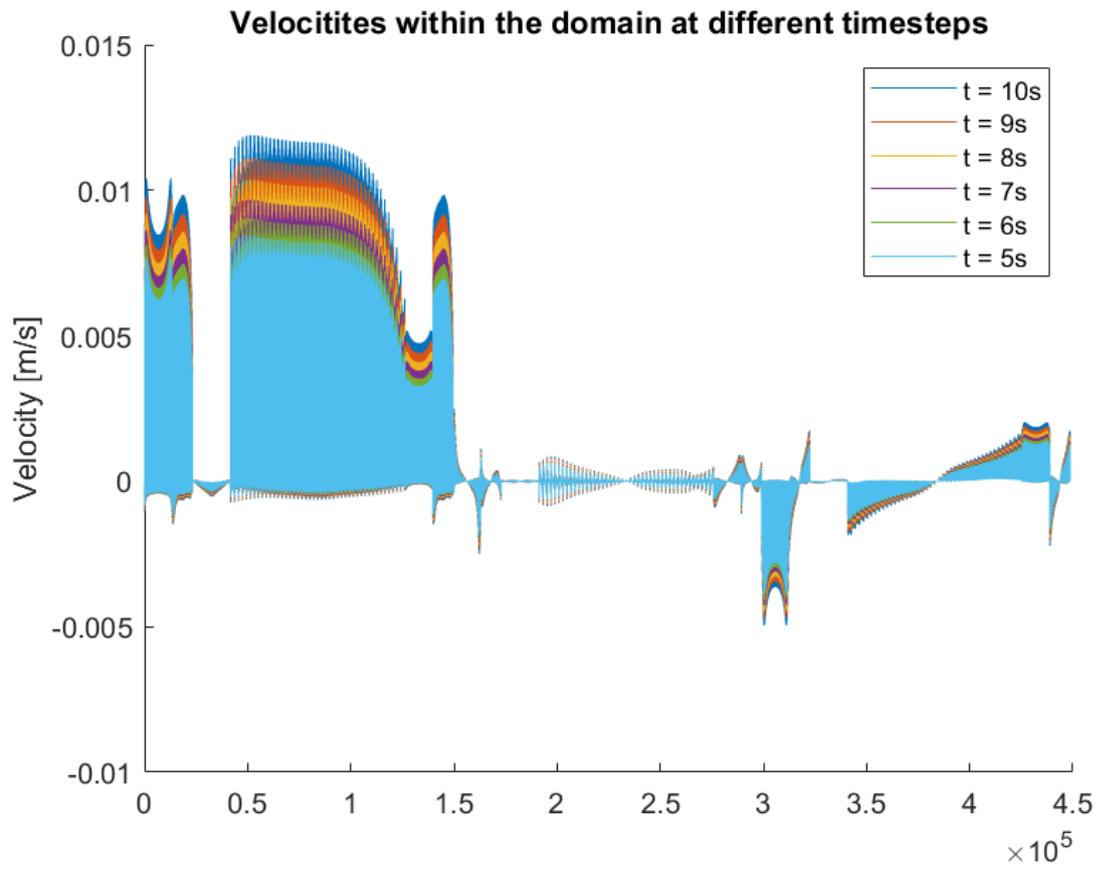


Figure 5.6: ROM cell velocity profile over time

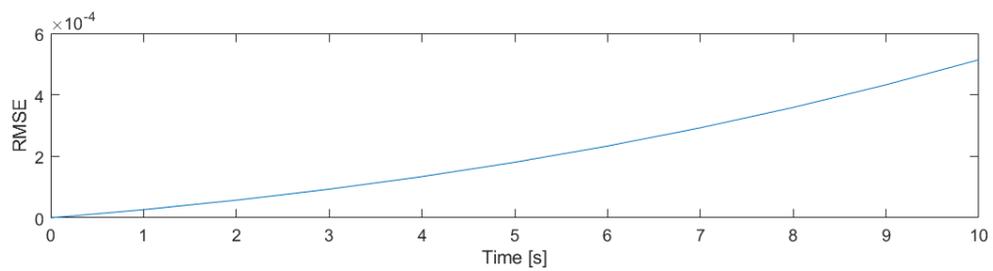


Figure 5.7: RMSE over time

6 Conclusion and Outlook

The ROM was able to capture the dominant flow behavior extracted from the numerical data and make accurate predictions into the future. The well-known instabilities occurring after a short time using the POD-Galerkin method were observed and quantified using the RMSE. The POD modes were also used to analyze the cavity flow and isolate different flow phenomena within the cavity.

The accuracy of the model, and potentially its stability, could be further increased by using more than 24 POD modes to construct it. Due to hardware limitations, this mode limit was not exceeded within this project's scope. Using more modes increases the storage space in RAM required drastically. Further program optimization could mitigate this limitation without requiring large amounts of RAM.

While the model computed orders of magnitude faster than the numerical simulation, it can be further optimized. The embedding of a ROM solver into CFD software will remove the input/output bottleneck and has the potential to reduce computation time by over 90%. This combination has been done for CFD codes like "AERO-F", developed by the Farhat Research Group at Stanford University.

The results computed by the ROM program require further verification. The flow velocities start to deviate immediately from the numerical ones at a faster pace than the truncation error, calculated using the POD eigenvalues, suggests. Furthermore, the occurring instabilities are caused by unrestricted growth of the a coefficients instead of fluctuations, shedding doubt upon the results. Additional unexpected predictions calculated using the model can be observed. A Galerkin-projection based on 24 POD modes should capture all the dominant flow characteristics. Figures 5.3 and 5.6 suggest that some regions are modeled poorly. A possible explanation for this phenomenon is inherent to the method applied. POD-Galerkin is known to model behavior that has not been previously observed within the domain very inaccurately. It fails to recreate propagating shocks modeled by the Burger's equation and similar transport phenomena.

After further verification of the ROM, additional measures should be implemented to increase its robustness. These measures could include changing the projection type, implementing adaptive h-refinement, or using supremizer enrichment for stabilization. The model should also be adapted to mimic the RANS equations. A possible verification case could be the lid-driven cavity with a significantly higher lid velocity.

Bibliography

- [1] Moser, R. D., Kim, J., and Mansour, N. N., “Direct numerical simulation of turbulent channel flow up to $Re \tau = 590$,” *Physics of fluids*, Vol. 11, No. 4, 1999, pp. 943–945.
- [2] Krzanich, B., “Brian Krzanich: Our Strategy and The Future of Intel,” <https://newsroom.intel.com/editorials/brian-krzanich-our-strategy-and-the-future-of-intel/#gs.334tsx>, Accessed: 2021-06-04.
- [3] Bristeau, M., MO, B., et al., “Transonic flow simulations by finite elements and least square methods,” 1980.
- [4] Euler, L., “Principes généraux du mouvement des fluides,” *Mémoires de l’Académie des Sciences de Berlin*, 1757, pp. 274–315.
- [5] “Memoires de L’Academie Royale des Sciences de L’Institut de France,” *VI*, 1827, p. 389.
- [6] Reynolds, O., “IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion,” *Philosophical transactions of the royal society of london.(a.)*, , No. 186, 1895, pp. 123–164.
- [7] Favre, A., “Equations des gaz turbulents compressibles,” *J. de Mécanique*, Vol. 4, No. 3, 1965.
- [8] Munz, C.-D., and Westermann, T., *Numerische Behandlung gewöhnlicher und partieller Differenzialgleichungen*, Springer, 2006.
- [9] MacCormack, R. W., *Numerical computation of compressible and viscous flow*, American Institute of Aeronautics and Astronautics, Inc., 2014.
- [10] Lomax, H., Pulliam, T. H., Zingg, D. W., Pulliam, T. H., and Zingg, D. W., *Fundamentals of computational fluid dynamics*, Vol. 246, Springer, 2001.
- [11] Magoulès, F., *Computational fluid dynamics*, CRC Press, 2011.
- [12] Diskin, B., and Thomas, J. L., “Accuracy of gradient reconstruction on grids with high aspect ratio,” *NIA report*, Vol. 12, 2008, p. 2008.
- [13] Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., and Tsamopoulos, J., “A critical analysis of some popular methods for the discretisation of the gradient operator in finite volume methods,” *Physics of Fluids*, Vol. 29, No. 12, 2017, p. 127103.

- [14] Liou, M.-S., and Steffen Jr, C. J., “A new flux splitting scheme,” *Journal of Computational physics*, Vol. 107, No. 1, 1993, pp. 23–39.
- [15] Roe, P. L., “Approximate Riemann solvers, parameter vectors, and difference schemes,” *Journal of computational physics*, Vol. 43, No. 2, 1981, pp. 357–372.
- [16] Lai, R., and Osher, S., “A splitting method for orthogonality constrained problems,” *Journal of Scientific Computing*, Vol. 58, No. 2, 2014, pp. 431–449.
- [17] Steger, J. L., and Warming, R., “Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods,” *Journal of computational physics*, Vol. 40, No. 2, 1981, pp. 263–293.
- [18] Van Leer, B., “Flux-vector splitting for the Euler equation,” *Upwind and high-resolution schemes*, Springer, 1997, pp. 80–89.
- [19] Liou, M.-S., “A sequel to ausm: Ausm+,” *Journal of computational Physics*, Vol. 129, No. 2, 1996, pp. 364–382.
- [20] Menter, F. R., “Two-equation eddy-viscosity turbulence models for engineering applications,” *AIAA journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.
- [21] Tollmien, W., Schlichting, H., Görtler, H., and Riegels, F., “Über Flüssigkeitsbewegung bei sehr kleiner Reibung,” *Ludwig Prandtl Gesammelte Abhandlungen*, Springer, 1961, pp. 575–584.
- [22] Schetz, J. A., and Bowersox, R. D., *Boundary layer analysis*, American Institute of Aeronautics and Astronautics, 2011.
- [23] Schlichting, H., and Gersten, K., *Boundary-layer theory*, Springer, 2016.
- [24] Bredberg, J., *On the Wall Boundary Condition for Turbulence Models*, Internal Report 00/4, Chalmers University of Technology, Göteborg, Sweden, 2000.
- [25] Knopp, T., Alrutz, T., and Schwamborn, D., “A grid and flow adaptive wall-function method for RANS turbulence modelling,” *Journal of Computational Physics*, Vol. 220, No. 1, 2006, pp. 19–40.
- [26] Prasad, A. K., and Koseff, J. R., “Reynolds number and end-wall effects on a lid-driven cavity flow,” *Physics of Fluids A: Fluid Dynamics*, Vol. 1, No. 2, 1989, pp. 208–218.
- [27] Leriche, E., and Gavrilakis, S., “Direct numerical simulation of the flow in a lid-driven cubical cavity,” *Physics of Fluids*, Vol. 12, No. 6, 2000, pp. 1363–1376.
- [28] Lumley, J. L., “The structure of inhomogeneous turbulent flows,” *Atmospheric turbulence and radio wave propagation*, 1967.

- [29] Berkooz, G., Holmes, P., and Lumley, J. L., “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.
- [30] Taira, K., Brunton, S. L., Dawson, S. T., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., “Modal analysis of fluid flows: An overview,” *Aiaa Journal*, Vol. 55, No. 12, 2017, pp. 4013–4041.
- [31] Chatterjee, A., “An introduction to the proper orthogonal decomposition,” *Current science*, 2000, pp. 808–817.
- [32] Weiss, J., “A tutorial on the proper orthogonal decomposition,” *AIAA Aviation 2019 Forum*, 2019, p. 3333.
- [33] Sirovich, L., “Turbulence and the dynamics of coherent structures. I. Coherent structures,” *Quarterly of applied mathematics*, Vol. 45, No. 3, 1987, pp. 561–571.
- [34] Kunisch, K., and Volkwein, S., “Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics,” *SIAM Journal on Numerical analysis*, Vol. 40, No. 2, 2002, pp. 492–515.
- [35] Iliescu, T., and Wang, Z., “Variational multiscale proper orthogonal decomposition: Navier-stokes equations,” *Numerical Methods for Partial Differential Equations*, Vol. 30, No. 2, 2014, pp. 641–663.
- [36] Lorenzi, S., Cammi, A., Luzzi, L., and Rozza, G., “POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 311, 2016, pp. 151–179.
- [37] Akhtar, I., Nayfeh, A. H., and Ribbens, C. J., “On the stability and extension of reduced-order Galerkin models in incompressible flows,” *Theoretical and Computational Fluid Dynamics*, Vol. 23, No. 3, 2009, pp. 213–237.
- [38] Stabile, G., and Rozza, G., “Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations,” *Computers & Fluids*, Vol. 173, 2018, pp. 273–284.
- [39] Rozza, G., and Veroy, K., “On the stability of the reduced basis method for Stokes equations in parametrized domains,” *Computer methods in applied mechanics and engineering*, Vol. 196, No. 7, 2007, pp. 1244–1260.
- [40] Stabile, G., Hijazi, S., Mola, A., Lorenzi, S., and Rozza, G., “POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder,” *arXiv preprint arXiv:1701.03424*, 2017.
- [41] Rowley, C. W., “Model reduction for fluids, using balanced proper orthogonal decomposition,” *International Journal of Bifurcation and Chaos*, Vol. 15, No. 03, 2005, pp. 997–1013.

- [42] Carlberg, K., Barone, M., and Antil, H., “Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction,” *Journal of Computational Physics*, Vol. 330, 2017, pp. 693–734.
- [43] Everson, R., and Sirovich, L., “Karhunen–Loeve procedure for gappy data,” *JOSA A*, Vol. 12, No. 8, 1995, pp. 1657–1664.
- [44] Huang, C., Duraisamy, K., and Merkle, C. L., “Investigations and improvement of robustness of reduced-order models of reacting flow,” *AIAA Journal*, Vol. 57, No. 12, 2019, pp. 5377–5389.
- [45] Carlberg, K., “Adaptive h-refinement for reduced-order models,” *International Journal for Numerical Methods in Engineering*, Vol. 102, No. 5, 2015, pp. 1192–1210.
- [46] Wang, Q., Hesthaven, J. S., and Ray, D., “Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem,” *Journal of computational physics*, Vol. 384, 2019, pp. 289–307.
- [47] Yu, Y., Koeglmeier, S., Sisco, J., and Anderson, W., “Combustion instability of gaseous fuels in a continuously variable resonance chamber (CVRC),” *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2008, p. 4657.
- [48] Ferziger, J. H., Perić, M., and Street, R. L., *Numerische Strömungsmechanik*, Vol. 1, Springer, 2008.
- [49] Sutherland, W., “The viscosity of gases and molecular force,” *Philosophical magazine S. 5*, 36, 1893, pp. 507–531.
- [50] Leer, B. V., *Lecture Notes in Physics*, Vol. 170, Springer-Verlag, New York/Berlin, 1982.
- [51] Tennekes, H., and Lumley, J. L., *A first course in turbulence*, MIT press, 2018.
- [52] Carlson, J.-R., Vatsa, V. N., and White, J., “Validation of a Node-Centered Wall Function Model for the Unstructured Flow Code FUN3D,” *AIAA, 22nd CFD Conference*, 2015.
- [53] Reichardt, H., “Vollständige Darstellung der turbulenten Geschwindigkeitsverteilung in glatten Leitungen,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, Vol. 31, No. 7, 1951, pp. 208–219.
- [54] Spalding, D., “A single formula for the “law of the wall”,” *J. Appl. Mech.*, Vol. 28, No. 3, 1961.