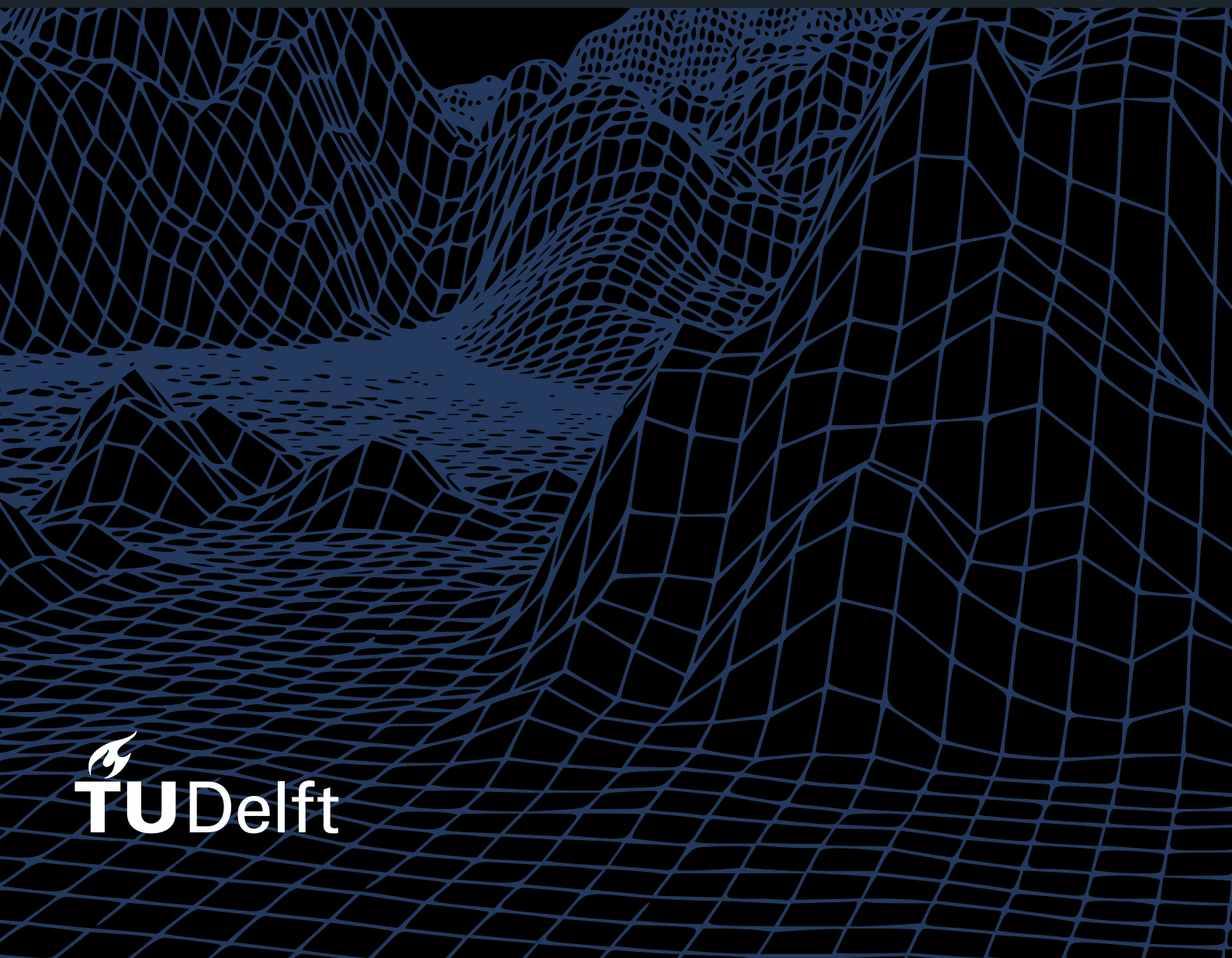
Comparing MSEM and conventional CFD for incompressible 2D Navier-Stokes

Thesis report

Jakub Plutecki



TU Delft

Comparing MSEM and conventional CFD for incompressible 2D Navier-Stokes

Thesis report

by

Jakub Plutecki

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on November 19, 2024 at 09:00

Thesis committee:

Chair:	Dr. S.J. Hulshoff
Supervisor:	Dr. M.I. Gerritsma
External examiner:	Dr. A. Palha
	Dr. Y. Zhang
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	June 2023 - November 2024
Student number:	4838513

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology

TU Delft

Preface

The following thesis report has been a result of several months of research, exploration, and scholarly inquiry in the field of the Mimetic Spectral Element Method. The thesis is a final project required to obtain a Master of Science degree from the Aerospace Faculty at Delft University of Technology.

I want to acknowledge the invaluable support and guidance of my thesis supervisor Marc Gerritsma, the creator of the phyem package Yi Zhang and Artur Palha for some insightful discussions. Their constructive criticism, feedback and scholarly contributions have substantially improved the quality and depth of this thesis. I am extremely grateful for their mentorship and support, which resulted in the successful completion of this thesis.

Writing this thesis has been a challenging yet rewarding journey and it would not have been possible without the support of my family. I would like to express my deepest gratitude to my parents and brother for their constant encouragement, patience and understanding throughout this process.

Jakub Plutecki
Delft, November 2024

Contents

Nomenclature

List of Abbreviations

AMG	Algebraic MultiGrid
BC	Boundary Conditions
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
DoF	Degrees of Freedom
DPIV	Digital Particle Imaging Velocimetry
FVM	Finite Volume Method
MEEVC	Mass, Energy, Enstrophy and Vorticity Conserving
MSEM	Mimetic Spectral Element Method
MSEPY	Mimetic Spectral Element in Python
PDE	Partial Differential Equations
RANS	Reynolds Averaged Navier-Stokes
RMS	Root Mean Square
RMSE	Root Mean Square Error
SPH	Smoothed Particle Hydrodynamics

List of Symbols

ν	Dynamic viscosity
ω	Vorticity
ε	Error
Δt	Timestep
Γ	Boundary section
$\wedge$	Wedge product
$\mathbb{R}$	Real coordinate space
$\mathcal{M}$	Manifold
$\mathcal{T}$	Trace operator
∇	Gradient operator
Ω	Domain
π_h	Projection operator
Ψ	Stream function
ρ	Fluid density
$\star$	Hodge star operator
θ	Angular coordinate
$\times$	Cross product
ξ	Spatial coordinate
a	Wake center position
b	Wake width
c_d	Drag coefficient
c_l	Lift coefficient
D	Cylinder diameter
d	Exterior derivitive
d^*	Codifferantial
h	Step height
I	Reconstruction operator
K	Number of mesh elements
l	Wake length
N	Polynomial degree
P	Total pressure
p	Static pressure
R	Reduction operator
Re	Reynolds number

St	Strouhal number	u_y	y velocity
t	Time	x	x-coordinate
u	Velocity	X_n	Reattachment position
u_x	x velocity	y	y-coordinate

List of Figures

List of Tables

Abstract

Computational Fluid Dynamics (CFD) has become important in designing aerospace and transport products. It allows for predicting the key flow properties. The CFD techniques were developed in the late 20th century. Mimetic schemes are relatively new in the realm of CFD, although they have been applied to a wide range of Partial Differential equations (PDE) problems. Some attempts have been made to draw comparisons between Finite Volume Methods and Finite Element Methods along with other conventional CFD techniques but not much attention has been paid towards the comparison of the Mimetic Spectral Element Method (MSEM) with these most conventional techniques. Further analysis is required in order to make judgments regarding the effectiveness, stability and accuracy of MSEM against conventional methods.

The purpose of this report is to present the findings and outcomes of 2D incompressible Navier-Stokes flow simulation techniques. Then, the comparison between the MSEM and conventional CFD will be made. The mass, energy, enstrophy and vorticity conserving (MEEVC) was chosen as a MSEM solver. MEEVC employs mixed finite element discretization which allows for the preservation of mass, energy, enstrophy and vorticity. The phyem package for Python created by Yi Zhang et al. is used to carry out MEEVC. Because a non-linear system needs to be solved, a Newton linearization is implemented. ANSYS Fluent represents the conventional CFD due to its built-in 2D solver. A coupled solver is selected as it solves for both pressure and velocity simultaneously compared to usually used segregated methods.

Four test cases were conducted using both software, and the results were compared with the literature for validation. They were selected in order to capture different flow conditions, including internal and external, steady and unsteady flows. The study validates both methods against the literature and analytical solutions. Then, a cross-case study is performed in order to answer the research questions.

1

Introduction

The use of Computational Fluid Dynamics (CFD) has become necessary in the design and evaluation of products manufactured in the aerospace and transport sectors, [1]. The CFD helps in the early design phases of product development, [2]. The key aspects of CFD are to study the properties of the fluids, to numerically approximate how the fluids will move with time and forces applied and how the movement would affect the surroundings. The computing techniques and physical models are becoming more and more advanced, [3]. They rely on the Navier–Stokes equations (continuity, momentum and energy equations). The development of computational fluid dynamics has been based quite significantly on the numerical formulations brought forth in the 1970s and 1980s, [4, 5, 6]. The conventional software packages for design include ANSYS CFX, ANSYS Fluent, Dassault Systemes XFlow and OpenFOAM. Finite Volume, Finite Element, Finite Difference, Lattice Boltzmann and Spectral Element are the most commonly used spatial discretization methods in computational fluid dynamics.

The Spectral Element Method was introduced in 1980s by Patera, [7]. It combined the finite element method with the accuracy of spectral techniques. A high-order Lagrangian interpolant is used for representation of velocity at each element through the Chebyshev grid. The discrete versions of differential operators gradient, divergence and curl were proposed by Hyman and Shashkov, [8]. Later, the support-operator method was used to construct second-order aforementioned operators [9] as natural discrete operators from [8] were not sufficient.

In ”Principles of Mimetic Discretizations of Differential Operators”, Pavel Bochev introduces a method for discretizing differential operators, [10] These mimetic operators are reduction and reconstruction. The framework is compatible with finite element/volume/difference methods. Direct and conforming methods are equivalent to regular reconstruction operators. This allows for the possibility of performing error analysis on direct mimetic methods, [10].

In the context of numerical methods and discretizations, ”mimetic” refers to the approach of imitating or reproducing specific mathematical properties. Mimetic discretization methods preserve the symmetries inherent in the underlying physical system [11]. These methods adhere to the same conservation laws as their continuous counterparts. Differential geometry and algebraic topology are core aspects of the mimetic discretizations, [11].

Recently, the mimetic spectral element method in CFD analysis has gained much attention. Tonti found the geometric aspects within physical theories comprising differential geometry and algebraic topology [12]. He explains the connection between physical variables in partial differential equations (PDEs) and geometric elements in a discrete scheme to construct a structure-preserving method. Another area of application was computational and geophysical fluid dynamics, [13]. Such approximations are supported by high-degree polynomials which approximate the

solution within each element in the MSEM. It should provide a better representation of the derivative and gradient operators compared to traditional numerical methods like Finite Volume Method and Finite Element Method. Also, the MSEM, unlike other CFD methods as described in [14] is less dependent on mesh so it may be implemented in arbitrarily shaped domains. One of the main advancements incorporated into the MSEMs are the discrete analogues of continuous differential operators that, in most cases, act similarly to gradient, divergence and curl operators, [15]. The correspondence of k-forms and k-cochains is built via mimetic operators like reduction, reconstruction and projection. Then the mass, energy, enstrophy and vorticity conserving (MEEVC) was proposed in [16] by Artur Palha and Marc Gerritsma. It is a mimetic spectral element combined with mixed finite element discretization designed for solving the 2D incompressible Navier-Stokes equations. It was modified in [17] by Yi Zhang to incorporate no-slip boundary conditions while satisfying the initial vorticity conservation law.

The MSEM offers versatile solutions for physical problems expressed in terms of partial differential equations (PDEs). The examples include: Stokes flow [18], diffusion [19, 20], advection [21, 22], 3D linear elasticity problems [23], geophysical flow [13, 24], electromagnetism [25, 26], Navier-Stokes [16, 17], shallow water flows [27, 28] and viscoelastic flows [29].

Comparison studies are essential in assessing the performance and accuracy of new computational methods and tools in fluid dynamics. To illustrate, the work of Liu [30] presents a comparison of a number of CFD codes aimed at the simulation of gas-solid fluidized bed hydrodynamics, thus shedding some light on how well these tools represent complex flow patterns. In a different study conducted by Herzog [31], a comprehensive comparison of the Large Eddy Simulation (LES) model and the Reynolds Averaged Navier Stokes (RANS) model for RCCI combustion was conducted. Stringer performed a comparative analysis of the commercial package CFX and the open-source package OpenFOAM in the simulation of flow over a circular cylinder, [32]. Jeong compared the two methods commonly employed in commercial CFD: Finite Volume Method (FVM) and Finite Element Method (FEM), [33]. Jet impingement tests were performed by López in OpenFOAM and ANSYS Fluent, [34]. These studies stress the need to assess and compare the differences in the functions and behaviour of the distinct methodologies. However, one interesting aspect has yet to be covered. No direct comparison was made between mimetic schemes and conventional CFD. Such comparisons are necessary to check whether mimetic schemes offer superior accuracy and stability in the simulation of Navier-Stokes equations.

The purpose of this report is to present the results of the two methods for flow simulation. The thesis aims to comprehensively explore and analyze the Mimetic Spectral Element Method and compare it to conventionally used CFD software. The study will be conducted for the incompressible 2D Navier-Stokes. A set of benchmark test cases are chosen to cover a range of flow scenarios. The data obtained from both MSEM and conventional CFD will be validated against available experimental, numerical data from the literature as well as with analytical solutions if those exist.

Four test cases were selected: Taylor Green vortex, lid-driven cavity flow, backward-facing step and flow around a cylinder. The Taylor-Green vortex is often treated as a standard case which involves the study of the time-evolving dissolution of vortex structures in a non-viscous or viscous fluid. The lid-driven cavity flow is often performed to check the precision of the CFD solvers applied to the shearing motion of the fluid located in a closed domain. The backward-facing step flow involves an abrupt increase in the width of the channel and gives rise to separation, reattachment and recirculation regions. The backward-facing step flow finds lots of everyday applications such as airflow over high angles of airfoils, airflow separation behind vehicles and airflow in the entry, condenser and combustion chamber of engines, [35, 36, 37]. Finally, the flow around a cylinder is one of the elementary problems in the field of external aerodynamics

where accurate prediction of the boundary layer development, wake and vortex shedding in the wake of a bluff body is the main challenge making it ideal for simulating obstructed flows and testing turbulence models in unsteady situations, [38].

1.1. Research Formulation

In the thesis, the investigation includes the research objective as well as supportive research questions. They are as follows:

Research Objective

A Comparative Study of Mimetic Spectral Element Method and Conventional CFD Techniques for incompressible 2D Navier-Stokes.

Research Question 1

How does the accuracy of the Mimetic Spectral Element Method (MSEM) compare to conventional Computational Fluid Dynamics (CFD) techniques?

Research Question 2

How do both methods compare to literature/experimental data?

Research Question 3

How do the Mimetic Spectral Element Method and conventional CFD methods perform across different flow types (e.g. internal, external, steady, unsteady)?

1.2. Structure of the Report

The structure of the report is as follows. Firstly, in Chapter 2 the general theory of the MSEM is presented as well as the python package used for simulations - phyem. For comparison, the conventional CFD methods are described in Chapter 3. The numerical results are presented in Chapter 4. The main focus of the chapter is to compare MSEM and ANSYS FLUENT results for several test cases. The report is concluded in Chapter 5.

2

Mimetic Spectral Element Method

The Mimetic Spectral Element Method (MSEM) method imitates essential characteristics of mathematical and physical systems, such as conservation laws, duality and self-adjointness of differential operators. As the scope of this research is the 2-dimensional simulation, the approach for 2D will be presented in this chapter. It simplyfies the representation and discretization of the domain.

In this chapter, the flow of the phyem simulation package is presented, [39]. It was created by Andrea Brugnoli, Ramy Rashad, Stefano Stramigioli, Yi Zhang. The mass, energy, enstrophy and vorticity conserving (MEEVC) was initially proposed in [16] and modified in [17]. It is a mimetic spectral element solver designed to solve the 2D incompressible Navier-Stokes equations. The phyem package will be used for MEEVC simulations of the flow to compare with commercially available options.

Currently, there are two ways to simulate something. The first option is to use commercial software that works like a black box. This means we only need to understand the general concepts of pre-processing and post-processing. The second option is to use open-access libraries to set up the simulation. With this approach, we need to work out the math on paper (the abstract stage) and then use the library's functionalities accordingly. The phyem aims to deliver the third option which allows the user to work out the abstract part of the problem within the code itself, [39].

The PhyEM package enables LEGO-like multi-physics simulation by representing equations, weak formulations and discretization at a program level. It utilizes a differential forms-based mathematical representation. Its simplified flow is seen in Figure 2.1, [39].

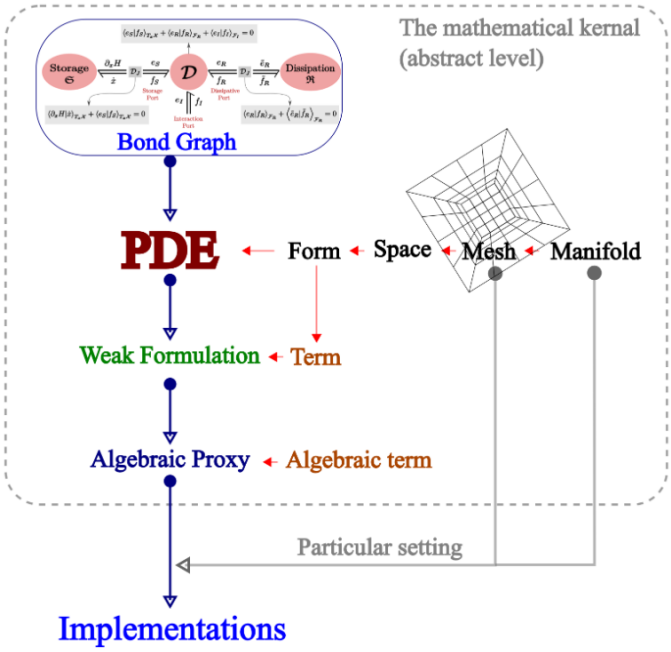


Figure 2.1: Simulation flow with the use of phyem package, [39].

2.1. Basis functions

Basis functions are a set of functions that form a basis for a vector space or function space. A basis is a set of linearly independent functions.

Nodal functions, also known as nodal basis functions or nodal shape functions, are used for representing and interpolating values within an element or mesh. Let's consider an interval $[-1, 1] \in \mathbb{R}$ and the nodes are then equal to:

$$-1 < \xi_0 < \xi_1 < ... < \xi_{n-1} < \xi_n < 1 \tag{2.1}$$

where nodes follow the Legendre-Gauss-Lobatto (LGL) grid, [40]. The LGL grid includes the endpoints of the interval. The Lagrange polynomial is the smallest degree polynomial that passes through a particular set of points. It is defined as:

$$h_i\left(\xi_j\right) = \delta_j^i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2.2}$$

with δ_j^i being the Kronecker delta function. It then follows:

$$h_i\left(\xi_j\right) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{\xi - \xi_j}{\xi_i - \xi_j} \tag{2.3}$$

The Lagrange polynomial basis for $n = 5$ nodes using the Gauss-Lobatto grid is presented in Figure 2.2. The function values for all nodes are preserved as only information from the particular node is used. Value from node 1 at $\xi = -1$, from node 3 at $\xi = 0$ etc. However, in between nodal points, the values are interpolated with information from all nodes. When trying to interpolate

a set of equidistant data points, using high-degree polynomials can result in inaccuracies and large oscillations in the interpolated function, particularly near the edges of the data interval.

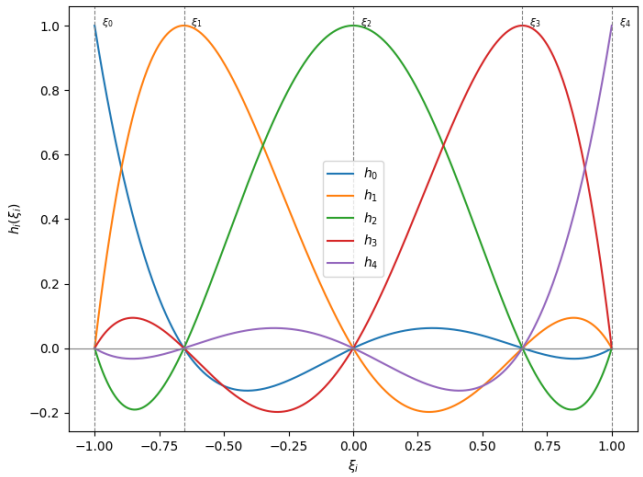


Figure 2.2: Nodal Lagrange polynomial basis for $n = 5$.

Edge basis functions are used to discretize functions along the edges or boundaries of elements in a mesh. Let's start with a definition of the basis functions. There exist n 1-cochains, in contrast to the $n + 1$ 0-cochains as n edges are located between the $n + 1$ nodes.

$$e_i(\xi) = -\sum_{k=0}^{i-1} \frac{dh_k(\xi)}{d\xi} \mathrm{d}\xi = -\sum_{k=0}^{i-1} \mathrm{d}h_k(\xi) \tag{2.4}$$

where $h_k(\xi)$ are the nodal functions and $dh_k(\xi)$ is an exterior derivative of $h_k(\xi)$. The basis functions, denoted as $e_i(\xi)$, can be interpreted as polynomial indicator functions which satisfy: [11]

$$\int_{\xi_{j-1}}^{\xi_j} e_i(\xi) = \delta_j^i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2.5}$$

with δ_j^i being the Kronecker delta function.

In Figure 2.3 edge functions for $n = 5$ are shown. On the contrary to the nodal functions in Figure 2.2, the information is not conserved on the nodes. Rather the integral of $e_i(\xi)$ is preserved.

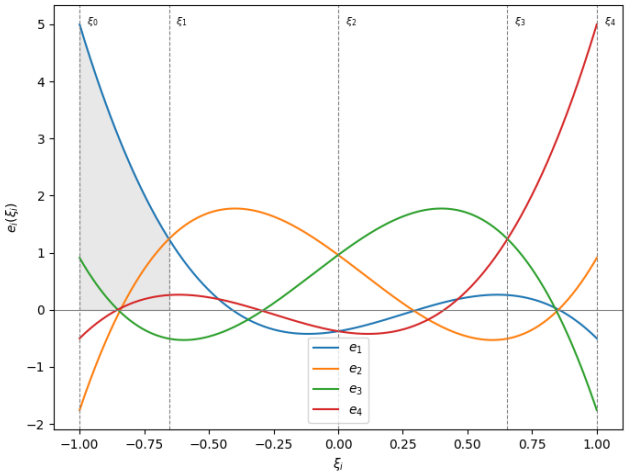


Figure 2.3: Edge functions basis for $n = 5$.

For edge functions, the Kronecker delta is not as clearly seen as for nodal functions as it has to satisfy the integral property (Equation 2.5). For example of edge function e_1, the integral over the range (ξ_0, ξ_1) equals to 1 which is indicated with shaded area in the Figure 2.3. The integral is equal to 0 for other segments.

2.2. Mimetic operators

In order to establish a connection between differential forms and cochains, three operators will be introduced. In the framework of mimetic methods, two key operators are introduced: the reduction operator $\mathcal{R}$ and the reconstruction operator $\mathcal{I}$. They give the projection operator as $\pi = \mathcal{I} \cdot \mathcal{R}$. It can be seen in Figure 2.4, [13].

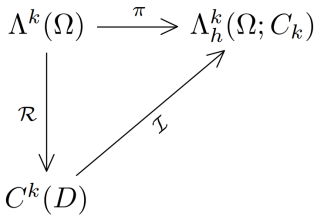


Figure 2.4: Illustration of mimetic operators, [13].

2.2.1. Reduction Operator

The reduction operator, denoted by $\mathcal{R} : \Lambda^k(\Omega) \to C^k(D)$, maps differential forms into cochains. It is also known as the De Rham map. The operator is defined by: [13]

$$\langle \mathcal{R}a^{(k)}, \tau_{(k)} \rangle = \int_{\tau^{(k)}} a^{(k)}, \quad \forall \tau^{(k)} \in C_k(D), \tag{2.6}$$

where $\tau_{(k)}$ is a k-chain in the complex $C_k(D)$ and $a^{(k)}$ is a k-form in $\Lambda^k(\Omega)$. It downsizes the continuous differential form to its discrete counterpart. When taking the exterior derivative at the continuous level and then reducing it through discretization, it is equivalent to first discretizing

and then applying the coboundary operator. This principle holds true for all differential forms, [14].

2.2.2. Reconstruction Operator

The reconstruction operator, denoted by $\mathcal{I} : C^k(D) \to \Lambda_h^k(\Omega; C_k)$ maps cochains back into differential forms. It is the inverse of the reduction operator $\mathcal{R}$. The reconstructed differential forms belong to the discrete space $\Lambda_h^k(\Omega; C_k)$, which is a subset of the continuous space $\Lambda^k(\Omega)$, [13].

$\mathcal{I}$ is the right inverse of $\mathcal{R}$, meaning that $\mathcal{R}\mathcal{I} = \text{Id}$ on $C^k(D)$. $\mathcal{I}$ also commutes with differentiation, ensuring that the exterior derivative of the reconstructed form follows:

$$d\mathcal{I} = \mathcal{I}\delta \text{ on } C^k(D). \tag{2.7}$$

2.2.3. Projection Operator

The projection operator π_h is defined as the composition of the reduction and reconstruction operators, $\pi_h = \mathcal{I}\mathcal{R}$. It provides a way to map a continuous k-form to its discrete approximation: [13]

$$a_h^{(k)} = \pi_h a^{(k)} = \mathcal{I}\mathcal{R}a^{(k)}, \quad a_h^{(k)} \in \Lambda_h^k(\Omega; C_k). \tag{2.8}$$

2.3. Space & Form

In order to begin the simulation, the embedding space dimension has to be set. It can be either one-, two- or three-dimensional. For the sake of this thesis, the 2D analysis will be performed. Then, both manifold and mesh are set. At the moment, it is impossible to visualize the manifold or mesh as everything is still at the abstract level, [41]. The size, shape and amount of the cells are carried in later parts of the simulation (see Section 2.7).

After presetting the simulation, both space and form can be defined. Space can be both outer- or inner-oriented as established in [42]. A form is simply an element of a space. Thus, it is rational to define a form through a space, [41]. In 2D Navier-Stokes simulation forms are as follows:

- Velocity $\vec{u}_h \in U_h \subset H(div; \Omega)$
- Vorticity $\omega_h \in \mathcal{W}_h \subset H(curl; \Omega)$
- Pressure $P_h \in Q_h \subset L^2(\Omega)$

With forms created, one can implement different operators to build more complex non-root forms. These operators are presented in Table 2.1, [41]. a and b represent forms where the operator is being implemented.

Table 2.1: Form operators in phyem.

Operator	Symbol	Useage
Exterior derivative	d	a.exterior_derivative()
Codifferential	$d^\star$	a.codifferential()
Time derivative	$\frac{\partial}{\partial t}$	a.time_derivative()
Wedge product	$\wedge$	phyem.wedge(a, b)
Inner product	$\langle \cdot, \cdot \rangle$	phyem.inner(a, b)
Hodge	$\star$	phyem.Hodge(a)
Trace	$\mathcal{T}$	phyem.trace(a)

2.4. De Rham complex

An incidence matrix is used to describe the action of the exterior derivative d. Together with the basis functions, need to be defined in a way that, when combined, creates a discrete version of the De Rham complex as seen in Figure 2.5, [42].

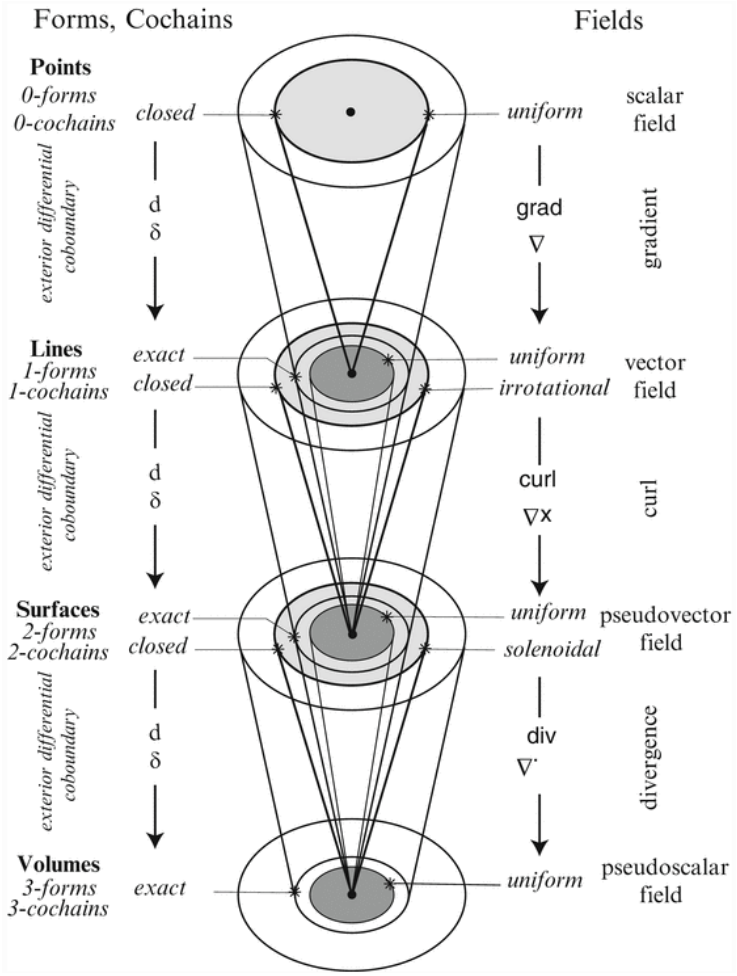


Figure 2.5: Generalized De Rham complex, [42].

In differential geometry, the exterior derivative operator d, encompasses the vector operations of gradient ∇, divergence $\nabla\cdot$ and curl $\nabla\times$. A general form of De Rham complex on a manifold Ω can be written as:

$$0 \xrightarrow{d} \Gamma^0(\Omega) \xrightarrow{d} \Gamma^1(\Omega) \xrightarrow{d} \Gamma^2(\Omega) \xrightarrow{d} ... \xrightarrow{d} \Gamma^k(\Omega) \xrightarrow{d} 0 \tag{2.9}$$

where $\Gamma^k(\Omega)$ represents the space of k-forms on the manifold Ω. d is the exterior derivative mapping k-forms to (k+1)-forms.

In $\mathbb{R}^1$ it reduces to:

$$0 \longrightarrow H^1(\Omega) \xrightarrow{\nabla} L^2(\Omega) \longrightarrow 0. \tag{2.10}$$

And for $\mathbb{R}^2$ it equals to: [10]

$$0 \longrightarrow H^1(\Omega) \xrightarrow{\nabla} H(\mathsf{rot};\Omega) \xrightarrow{\nabla\times} L^2(\Omega) \longrightarrow 0, \tag{2.11}$$

$$0 \longrightarrow H(\mathsf{curl};\Omega) \xrightarrow{\nabla\times} H(\mathsf{div};\Omega) \xrightarrow{\nabla\cdot} L^2(\Omega) \longrightarrow 0. \tag{2.12}$$

The discrete De Rham complex for $\mathbb{R}^2$ can be seen in Equation 2.14. With the Hodge matrices necessary to convert from primal to dual mesh. $C(\Omega)$, $D(\Omega)$, $S(\Omega)$, $G(\Omega)$ and $R(\Omega)$ are used to express finite dimensional conforming function spaces that are subsets of Sobolev spaces, [17].

$$\begin{gathered} C(\Omega) \subset H(\mathsf{curl};\Omega), \quad D(\Omega) \subset H(\mathsf{div};\Omega), \quad S(\Omega) \subset L^2(\Omega), \\ G(\Omega) \subset H^1(\Omega), \quad R(\Omega) \subset H(\mathsf{rot};\Omega), \end{gathered} \tag{2.13}$$

$$\begin{aligned} \mathbb{R} \hookrightarrow C(\Omega) \xrightarrow{\nabla\times} D(\Omega) \xrightarrow{\nabla\cdot} S(\Omega) \to 0, \\ \mathbb{R} \hookrightarrow G(\Omega) \xrightarrow{\nabla} R(\Omega) \xrightarrow{\nabla\times} S(\Omega) \to 0. \end{aligned} \tag{2.14}$$

2.5. Mixed weak formulation

When the space and form are created, the partial differential equations can be constructed. It is done with the command phyem.pde, [41].

Let's consider viscous incompressible Navier-Stokes equations

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nabla \cdot (\nu \nabla \vec{u}) + \nabla p = \vec{f}, \tag{2.15a}$$

$$\nabla \cdot \vec{u} = 0, \tag{2.15b}$$

where ∂_t is time derivative, $\vec{u}$ is velocity, ν is fluid dynamic (absolute) viscosity, p is a static pressure and $\vec{f} = 0$ is a body force per unit mass as no external forces are applied to the system.

The Equation 2.15 can be rewritten in Velocity-Vorticity-Pressure (VVP) formulation

$$\partial_t \vec{u} + \omega \times \vec{u} + \nu \nabla \times \omega + \nabla P = 0, \tag{2.16a}$$

$$\omega - \nabla \times \vec{u} = 0, \tag{2.16b}$$

$$\nabla \cdot \vec{u} = 0. \tag{2.16c}$$

The static pressure p is substituted with total pressure $P = p + \frac{1}{2}\vec{u}^2$ for incompressible flow. The $\omega = \nabla \times \vec{u}$ represents the vorticity.

The unknowns have to be set for the equations. These are velocity u, vorticity ω and total pressure $P = p + \frac{1}{2}\vec{u} \cdot \vec{u}$ with static pressure p. Then boundary conditions are set with phyem.pde.bc.define_bc. Only two types, essential and natural, of boundary conditions are implemented, [41].

Two pairs of boundary conditions are prescribed [17]

$$\begin{aligned} \vec{u} \cdot n &= \hat{u}_\perp \qquad \text{on } \Gamma_\perp \\ P &= \hat{P} \qquad \text{on } \Gamma_{\hat{P}}, \end{aligned} \tag{2.17}$$

$$\begin{aligned} \omega &= \hat{\omega} \qquad \text{on } \Gamma_{\hat{\omega}}, \\ \vec{u} \times n &= \hat{u}_{\parallel} \qquad \text{on} \Gamma_{\parallel}, \end{aligned} \tag{2.18}$$

with unit outward norm vector n and pairs of boundaries $\Gamma_{\hat{\omega}} \cup \Gamma_{\parallel} = \Gamma_{\perp} \cup \Gamma_{\hat{P}} = \partial\Omega$. The boundaries are disjoint in each pair i.e. $\Gamma_{\hat{\omega}} \cap \Gamma_{\parallel} = \Gamma_{\perp} \cap \Gamma_{\hat{P}} = \emptyset$.

The trace operator $\mathcal{T}$ restricts a function to a boundary section $\Gamma \subset \partial\Omega$. It acts on $\omega \in H(\mathsf{curl}; \Omega)$, $\varphi \in H^1(\Omega)$, $\vec{u} \in H(\mathsf{div}; \Omega)$ and $\sigma \in H(\mathsf{rot}; \Omega)$. It is defined as follows: [17]

$$\mathcal{T}\omega = \omega|_{\Gamma}, \quad \mathcal{T}\varphi = \varphi|_{\Gamma}, \tag{2.19}$$

$$\mathcal{T}\vec{u} = \vec{u} \cdot n|_{\Gamma}, \quad \mathcal{T}\sigma = \sigma \times n|_{\Gamma}. \tag{2.20}$$

with trace spaces:

$$\mathcal{T}H(\mathsf{curl}; \Omega, \Gamma) := \{\mathcal{T}\omega \mid \omega \in H(\mathsf{curl}; \Omega)\}, \tag{2.21}$$

$$\mathcal{T}H(\mathsf{div}; \Omega, \Gamma) := \{\mathcal{T}\vec{u} \mid \vec{u} \in H(\mathsf{div}; \Omega)\}, \tag{2.22}$$

$$H^{1/2}(\Omega, \Gamma) := \{\mathcal{T}\varphi \mid \varphi \in H^1(\Omega)\}, \tag{2.23}$$

$$\mathcal{T}H(\mathsf{rot}; \Omega, \Gamma) := \{\mathcal{T}\sigma \mid \sigma \in H(\mathsf{rot}; \Omega)\}. \tag{2.24}$$

After establishing boundary conditions, The weak formulation can be set up. One can integrate by parts, the equation which results in the following. In $\Omega \subset \mathbb{R}^2$, unknowns $(u_h, \omega_h, P_h) \in D(\Omega) \times C(\Omega) \times S(\Omega)$, such that [17]

$$\begin{cases} \underbrace{\langle \partial_t u_h, \tilde{v}_h \rangle_{\Omega}}_{0-0} + \underbrace{a(\omega_h, u_h, \tilde{v}_h)_{\Omega}}_{0-1} + \underbrace{\nu \langle \nabla \times \omega_h, \tilde{v}_h \rangle_{\Omega}}_{0-2} - \underbrace{\langle P_h, \nabla \cdot \tilde{v}_h \rangle_{\Omega}}_{0-3} + \underbrace{\langle \hat{P} | \mathcal{T}\tilde{v}_h \rangle_{\Gamma_{\hat{P}}}}_{0-4} = 0 \quad \forall \tilde{v}_h \in D_0(\Omega, \Gamma_{\perp}), \\ \underbrace{\langle \omega_h, \tilde{w}_h \rangle_{\Omega}}_{1-0} - \underbrace{\langle u_h, \nabla \times \tilde{w}_h \rangle_{\Omega}}_{1-1} + \underbrace{\langle \hat{u}_{\parallel} | \mathcal{T}\tilde{w}_h \rangle_{\Gamma_{\parallel}}}_{1-2} = 0 \quad \forall \tilde{w} \in C_0(\Omega, \Gamma_{\hat{\omega}}), \\ \underbrace{\langle \nabla \cdot u_h, \tilde{q}_h \rangle_{\Omega}}_{2-0} = 0 \quad \forall \tilde{q}_h \in S(\Omega). \end{cases} \tag{2.25}$$

where indices below the element of the equation show the numbering of each part. $\hat{P} \in H^{1/2}(\Omega, \Gamma_{\hat{P}})$ and $\hat{u}_{\parallel} \in H(\mathsf{rot}; \Gamma, \Gamma_{\parallel})$ being the natural BC while $\mathcal{T}u_h = \hat{u}_{\perp} \in \mathcal{T}D(\Omega, \Gamma_{\perp})$ and $\mathcal{T}\omega_h = \hat{\omega} \in \mathcal{T}C(\Omega, \Gamma_{\hat{\omega}})$ are the essential BC. The initial condition $(u_h^0, \omega_h^0) \in D(\Omega) \times C(\Omega)$. No external force f is present in the equations as it is assumed to be zero for all test cases.

Element $1 - 2$ can be reduced for the case of the natural BC for tangential velocity being 0 all over the boundary. Moreover, when only normal velocity is prescribed on the boundary and pressure is not, it cancels the $0 - 4$ element from the equation. This results in:

$$\begin{cases} \underbrace{\langle \partial_t u_h, \tilde{v}_h \rangle_{\Omega}}_{0-0} + \underbrace{a(\omega_h, u_h, \tilde{v}_h)_{\Omega}}_{0-1} + \underbrace{\nu \langle \nabla \times \omega_h, \tilde{v}_h \rangle_{\Omega}}_{0-2} - \underbrace{\langle P_h, \nabla \cdot \tilde{v}_h \rangle_{\Omega}}_{0-3} = 0 \quad \forall \tilde{v}_h \in D_0(\Omega, \Gamma_{\perp}), \\ \underbrace{\langle \omega_h, \tilde{w}_h \rangle_{\Omega}}_{1-0} - \underbrace{\langle u_h, \nabla \times \tilde{w}_h \rangle_{\Omega}}_{1-1} = 0 \quad \forall \tilde{w} \in C_0(\Omega, \Gamma_{\hat{\omega}}), \\ \underbrace{\langle \nabla \cdot u_h, \tilde{q}_h \rangle_{\Omega}}_{2-0} = 0 \quad \forall \tilde{q}_h \in S(\Omega). \end{cases} \tag{2.26}$$

A mixed finite element formulation is used to spatial discretize the Navier–Stokes equations. It means that each of the unknowns has a separate discrete degree of freedom (DoF). As it is an

important step in normalizing the mesh when comparing with conventional CFD because it can tell the size of the matrix system one is solving for (at each timestep). The DoF is calculated in the following manner

$$P_{\text{DoF}} = K_x \cdot K_y \cdot N^2 \tag{2.27a}$$
$$\omega_{\text{DoF}} = (K_x + 1) \cdot (K_y + 1) \cdot N^2 \tag{2.27b}$$
$$u_{\text{DoF}} = (K_x \cdot N + 1) \cdot (K_y \cdot N) + (K_x \cdot N) \cdot (K_y \cdot N + 1) \tag{2.27c}$$

for an orthogonal mesh with $K_x \times K_y$ grid. The total DoF is sum of P_{DoF}, $\omega_{\text{DoF}}, u_{\text{DoF}}$. For more complex geometries, one can just divide it into segments and calculate DoF for each.

2.6. Temporal discretization

While MSEM is primarily designed for spatial discretization, it is usually combined with various time integration schemes to solve time-dependent problems. Effective time integration schemes are crucial for maintaining conservation properties, [43]. The primary focus of this thesis is on incompressible 2D Navier-Stokes equations.

First, an abstract time sequence has to be set up. Then, the time intervals Δt can be defined. For the temporal discretization, the classic implicit midpoint method is used, [44].

$$\left\langle \frac{u_h^k - u_h^{k-1}}{\Delta t}, \tilde{v}_h \right\rangle_\Omega + a\left(\frac{\omega_h^{k-1} + \omega_h^k}{2}, \frac{u_h^{k-1} + u_h^k}{2}, \tilde{v}_h \right)_\Omega + \nu \left\langle \nabla \times \frac{\omega_h^{k-1} + \omega_h^k}{2}, \tilde{v}_h \right\rangle_\Omega -$$
$$\left\langle P_h^{k-\frac{1}{2}}, \nabla \cdot \tilde{v}_h \right\rangle_\Omega = -\left\langle \hat{P}^{k-\frac{1}{2}} | \mathcal{T} \tilde{v}_h \right\rangle_{\Gamma_{\hat{P}}} \quad \forall \tilde{v}_h \in D_0(\Omega, \Gamma_\perp), \tag{2.28a}$$
$$\left\langle \omega_h^k, \tilde{w}_h \right\rangle_\Omega - \left\langle u_h^k, \nabla \times \tilde{w}_h \right\rangle_\Omega = -\left\langle \hat{u}_\parallel^k | \mathcal{T} \tilde{w}_h \right\rangle_{\Gamma_\parallel} \quad \forall \tilde{w} \in C_0(\Omega, \Gamma_{\hat{\omega}}), \tag{2.28b}$$
$$\left\langle \nabla \cdot u_h^k, \tilde{q}_h \right\rangle_\Omega = 0 \quad \forall \tilde{q}_h \in S(\Omega) \tag{2.28c}$$

with

$$\Delta t = t_k - t_{k-1} > 0 \quad \text{for } k \in \{1, 2, 3, ...\} \tag{2.29}$$

and unknowns being (u_h^k, ω_h^k and $P_h^{k-\frac{1}{2}}$) $\in D(\Omega) \times C(\Omega) \times S(\Omega)$.

The law of conservation of mass is an essential principle of nature that is being upheld at every moment in time, [17]. The incompressibility indicates that the density is constant. Then, the discrete weak formulation of Equation 2.16c strongly enforces

$$\nabla \cdot u_h^k = 0 \tag{2.30}$$

all across Ω.

Then spatial discretization on an abstract mesh is achieved by simply specifying the finite degrees to finite-dimensional forms with phyem.space.finite(degree).

Time sequences can be defined in two ways. Either it is constant throughout the simulation or it is described by a function. This function is defined by a user and e.g. might be dependent on the velocity residual. After defining the time sequence, one can prescribe initial and boundary conditions.

2.7. MSEPY

With the fully discrete weak formulation, the algebraic proxy can be built by calling its method mp, standing for matrix proxy, [41]. Either a linear or nonlinear algebraic system is created. The aforementioned NS equation, it results in a non-linear system.

As the problem is non-linear, Newton's method is used to linearize the system. It makes a current guess of the solution and iteratively improves it. A tolerance of tol $= 10^{-8}$ is implemented to minimize iterative process errors.

MSEPY stands for mimetic spectral elements in Python. "It means that the most computationally intensive part of the simulation, the linear system solving is done with Python", [41]. The advantage of it is that no API is needed for simulation. On the other hand, Python operates at a relatively lower speed so it is not suitable for larger problems.

Implementation starts with the configuration of the manifold and mesh. All predefined MSEPY manifolds are shown in Table 2.2.

Table 2.2: Predefined MSEPY manifolds.

Manifold	Indicator
Rectangular domain	'crazy'
Multiple rectangular domains	'crazy_multi'
Backward step	'backward_step'
Cylinder channel	'cylinder_channel'

The post-processing is done by use of Matplotlib package and external software. The solution can be saved to a VTK file and then loaded into the open-source visualization software Paraview or directly read in Python using VTK package. Alternatively, velocity, vorticity and pressure fields might be visualized with Matplotlib.

3

Conventional CFD

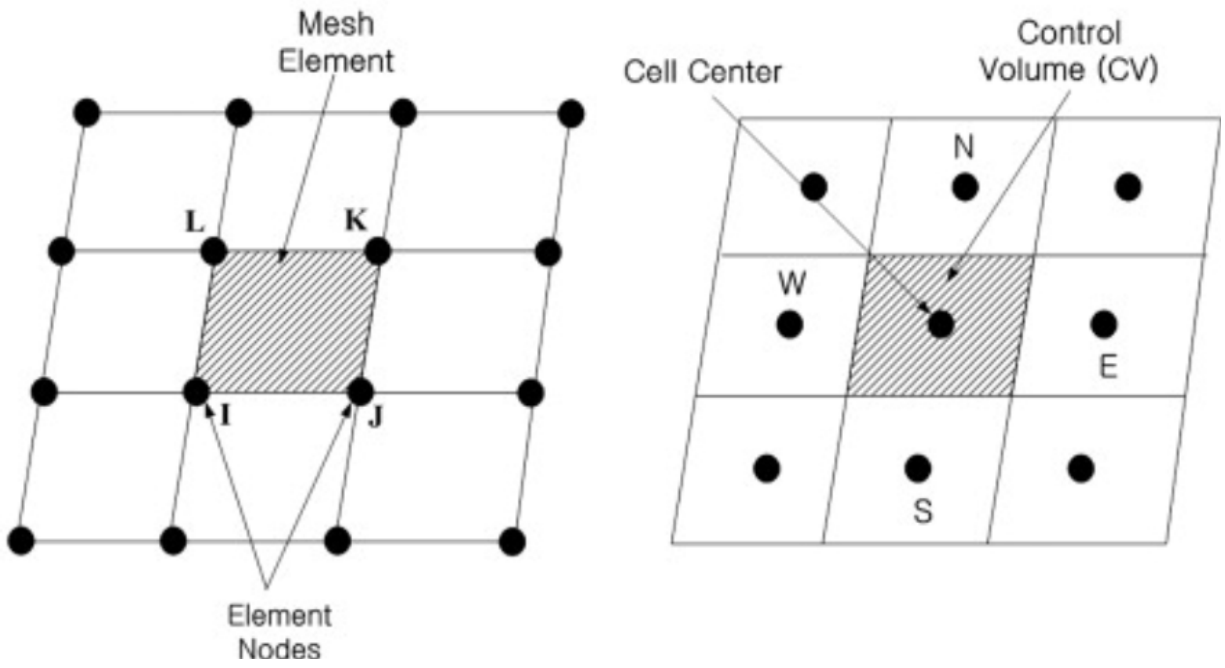
Conventional Computational Fluid Dynamics (CFD) refers to the established numerical techniques and methods used for simulating and analyzing fluid flows through Computer Aided Engineering (CAE). These methods typically involve discretizing the governing equations of fluid motion, such as the Navier-Stokes equations. The description of the model will be discussed in Section 3.1. Mesh types will be presented in Section 3.2. Finally, convergence criteria will be shown in Section 3.3.

3.1. Model

The selected software for the comparison is ANSYS Fluent as CFX is used predominantly for turbomachinery. Also, there is a 2D solver provided in ANSYS FLUENT, which is advantageous considering the research topic specified for this thesis. It provides two-dimensional and three-dimensional imagery and visualization tools, contour and vector images, animation and streamline visualization tools which is quite sufficient to facilitate comparison with MSEM, [45]. The ANSYS Fluent solver utilizes a cell-centred finite volume approach, where the flow variables are stored at the center of each mesh element, [33].

Which turbulence model is appropriate for the ANSYS simulation model of the 2D incompressible Navier-Stokes equation? The Reynolds number Re will not exceed 1600. The transition from laminar to turbulent flow is influenced by the degree of flow disruption caused by surface roughness, pipe vibrations and upstream flow fluctuations, [46]. As the simulations are restricted to the incompressible laminar flow regime, the laminar model was selected for ANSYS Fluent. It operates without incorporating any turbulence modelling, as it assumes smooth, orderly fluid flow and neglects turbulent fluctuations.

Among the popular techniques employed in this research are Finite Volume Method (FVM) and Finite Element Method (FEM), [33]. FVM is the computational approach adopted in ANSYS Fluent and it involves clustering the geometry being analyzed into smaller volumes known as control volumes where the various flow parameters are stored at each volume's center. This technique allows the integration of the governing equations over the particular control volumes. On the other hand, FEM, which is featured in ANSYS CFX. It uses elements containing flow variables at the element nodes. FEM is flexible in the meshing of such geometries and is part of a hierarchical accelerated approach which consists of an algebraic multigrid technique to improve speed. Both discretization method are shown in Figure 3.1, [33].



(a) FEM **(b)** FVM

Figure 3.1: Structured mesh for two discretization methods, [33].

3.1.1. Solver algorithms

The laminar model solves incompressible Navier-Stokes equations which in their vector form follow

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nabla \cdot (\nu \nabla \vec{u}) = -\nabla p \tag{3.1a}$$

$$\nabla \cdot \vec{u} = 0. \tag{3.1b}$$

with momentum equation (Equation 3.1a) and continuity equation (Equation 3.1b). The time derivative of velocity $\vec{u}$ in the momentum equation is set to zero $\frac{\partial \vec{u}}{\partial t} = 0$ as a steady-state is assumed. Even when the transient simulation is solved, for every timestep the needs to reach a steady state (RMS residuals convergence).

SIMPLE algorithms

The SIMPLE algorithm stands for Semi-Implicit Method for Pressure Linked Equations, [45] [47]. It was introduced by Patankar and Spalding in 1972, [48]. It is an iterative method of calculating pressure and velocities. The unknowns are solved sequentially. The procedure is presented in Figure 3.2.

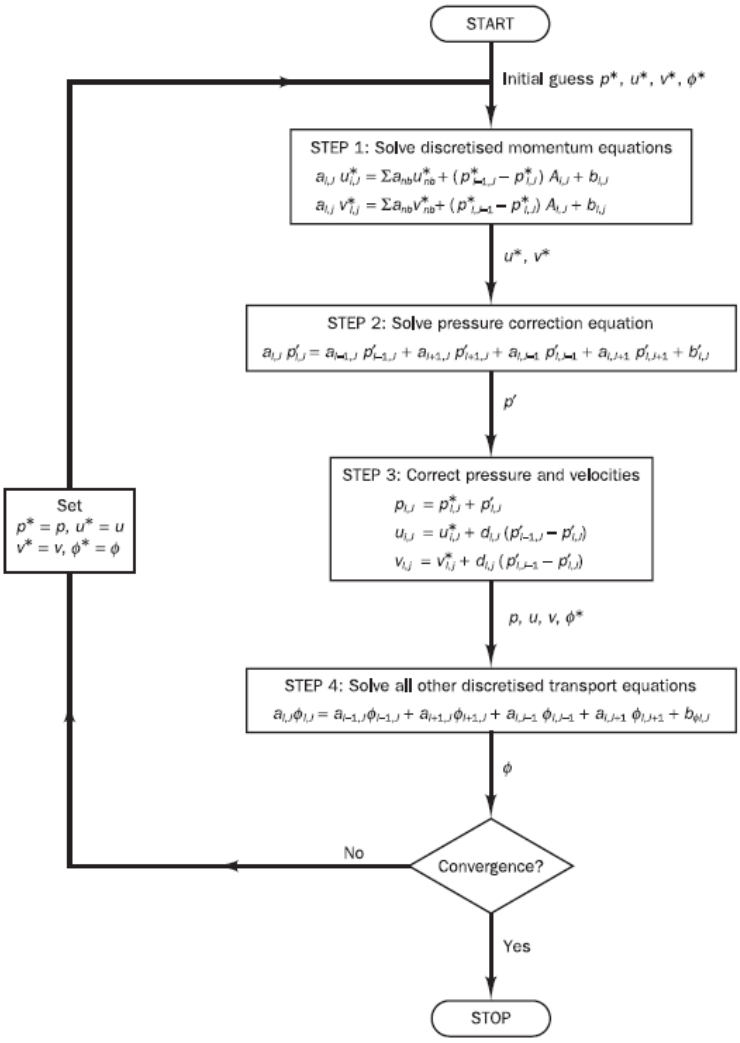


Figure 3.2: SIMPLE algorithm sequence, [48].

Firstly, the pressure p^* is guessed. Then, the momentum equation is solved with this pressure, yielding velocities $\vec{u}$ as

$$\mathcal{L}\vec{u} = -\nabla p \tag{3.2}$$

The calculated velocities do not satisfy the continuity equation yet. In order to do so, pressure is corrected with

$$\nabla \cdot (\mathcal{A}^{-1}\nabla p) = \nabla \cdot (\mathcal{A}^{-1}\mathcal{H}) \tag{3.3}$$

with $\mathcal{H} = \mathcal{A}\vec{u} - \mathcal{L}\vec{u}$ with velocities from previous iteration.

Finally, a new pressure field is used to correct for the velocities as follows

$$\vec{u} = \mathcal{A}^{-1}\mathcal{H} - \mathcal{A}^{-1}\nabla p \tag{3.4}$$

The new velocities do satisfy the momentum equation and the process is repeated until both Equation 3.1a and Equation 3.1b are satisfied.

Coupled algorithms
The coupled algorithm [45] solves the momentum and pressure-based continuity equations together. Unlike the segregated method, which solves these equations separately, the coupled approach uses an implicit discretization for both the momentum and continuity equations.

It is based on the implicit treatment of the pressure gradient in momentum equations and mass source in the pressure correction equation. Contrary to the SIMPLE solver, velocities and pressure are calculated simultaneously.

The pressure gradient in the momentum equations for component U_k is computed as:

$$\sum_f p_f A_k = -\sum_j a_{ij}^{u_k p} p_j \tag{3.5}$$

where $a_{ij}^{u_k p}$ is the coefficient derived from the Gauss divergence theorem and the pressure interpolation scheme.

For any i-th cell, the discretized form of the momentum equation becomes:

$$\sum_j a_{ij} a_{ij}^{u_k u_k} u_{kj} + \sum_j a_{ij}^{u_k p} p_j = b_i^{u_k} \tag{3.6}$$

In the continuity equation, the balance of fluxes is replaced using the flux expression. This results in the discretized form:

$$\sum_k \sum_j a_{ij}^{p u_k} u_{kj} + \sum_j a_{ij}^{pp} p_j = b_i^p \tag{3.7}$$

As a result, the overall system of equations, after being transformed into δ-form, is presented as:

$$\sum_j [A]_{ij} \vec{X}_j = \vec{B}_i \tag{3.8}$$

where the influence matrix A_{ij} between cells i and j has the form in 2D:

$$[A]_{ij} = \begin{pmatrix} a_{ij}^{pp} & a_{ij}^{pu} & a_{ij}^{pv} \\ a_{ij}^{up} & a_{ij}^{uu} & a_{ij}^{uv} \\ a_{ij}^{vp} & a_{ij}^{vu} & a_{ij}^{vv} \end{pmatrix} \tag{3.9}$$

and the unknown and residual vectors are given by:

$$\vec{X}_j = \begin{pmatrix} p_i' \\ u_i' \\ v_i' \end{pmatrix}, \quad \vec{B}_i = \begin{pmatrix} -r_i^p \\ -r_i^u \\ -r_i^v \end{pmatrix} \tag{3.10}$$

The system of equations in (3.8) is solved using the coupled Algebraic MultiGrid (AMG) solver, which accelerates convergence by solving pressure and velocity equations simultaneously.

The linearized equations are solved through the use of an Algebraic MultiGrid (AMG) solver, [45]. It generates coarse-level equations algebraically. This makes AMG particularly suitable for use on unstructured meshes. Unlike geometric multigrid methods, AMG requires no coarse meshes and the discretized equations are evaluated at every level. The benefit of AMG over geometric multigrid is the ability to perform better for non-linear problems, as the non-linearities are retained when transferring information to coarse levels.

Comparison
In Figure 3.3, two algorithms (SIMPLE and Coupled) are tested for the Taylor-Green Vortex (TGV) problem. A more detailed description of the benchmark can be found in Section 4.1. The L^2-error is calculated against the analytical solution to access two methods. As seen from the plot, both solving methods are performing quite similarly. It is with agreement with the literature [49] where SIMPLE and Coupled solvers perform similarly up till around 10^6 gird size. The size of TGV grid is $\approx 10^3$ and in all test cases, it does not go over 10^6 cells. The test by George [49] was performed on 3D lid cavity flow and 3D backward-facing step. For the comparison between MSEM, a coupled solver is chosen.

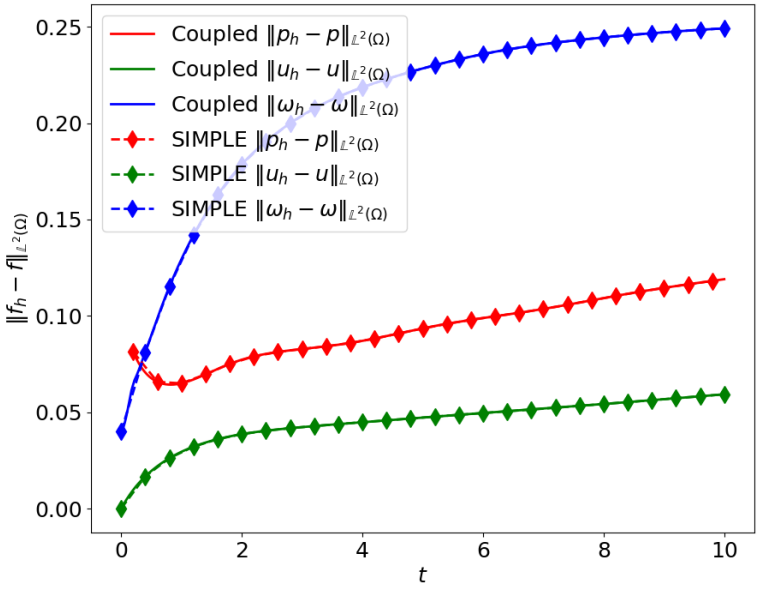


Figure 3.3: L^2-error for SIMPLE and Coupled algorithm over time t.

The L^2-error for continuous functions is defined as:

$$\|f - \hat{f}\|_{L^2(\Omega)} = \left(\int_\Omega \left(f(x,y) - \hat{f}(x,y) \right)^2 dA \right)^{\frac{1}{2}} \tag{3.11}$$

where $f(x,y)$ is the true function, $\hat{f}(x,y)$ is the approximate or predicted function and Ω is the domain of integration.

3.1.2. Spatial and temporal discretiation
For spatial discretization, certain parameters have to be chosen. Firstly, the scheme for gradient calculation which is important for computing secondary diffusion terms and velocity derivatives.

The Least Squares Cell-Based Gradient Evaluation method was selected. The solution is assumed to vary linearly between cell centroids. The change in values between cell c_0 and cell c_i, along the vector $\delta\vec{r_i}$, can be expressed as: [45]

$$(\nabla\phi)_{c_0} \cdot \delta\vec{r_i} = (\phi_{c_i} - \phi_{c_0}) . \tag{3.12}$$

By applying this relationship to all surrounding cells, the system can be written in matrix form:

$$[J] (\nabla\phi)_{c0} = \Delta\phi \tag{3.13}$$

where $[J]$ is a coefficient matrix based on the geometry of the mesh. The cell gradient $\nabla\phi$ is then determined by solving a least-squares minimization problem for the over-determined system, typically using a Gram-Schmidt decomposition to calculate weight factors for each face of the cell, [45].

Secondly, the Pressure Interpolation Scheme had to be selected. ANSYS Fluent uses a co-located scheme, whereby pressure and velocity are both stored at cell centres, [45]. For this, default Second Order scheme is used. It may provide improved accuracy over the other two schemes (linear and standard). It reconstructs the face pressure using a central differencing scheme: [45]

$$P_f = \frac{1}{2} (P_{c_0} + P_{c_1}) + \frac{1}{2} (\nabla P_{c_0} \cdot \vec{r}_{c_0} + \nabla P_{c_1} \cdot \vec{r}_{c_1}) . \tag{3.14}$$

The time discretization of the transient term is done using the Second Order Backward Euler scheme. Lower-order schemes suppress the higher-wave-number components, which are responsible for initiating asymmetry in flow around the cylinder, [50].

The generalized second-order discretization for time derivatives is given by: [45]

$$F(\phi) = \frac{(1 + \tau^{n+\frac{1}{2}})\phi^{n-1} - (1 + \tau^{n+\frac{1}{2}} + \tau^{n-\frac{1}{2}})\phi^n + \tau^{n-\frac{1}{2}}\phi^{n+1}}{2\Delta t^{n+1}} \tag{3.15}$$

where the time step ratio τ is defined as:

$$\tau^{n+\frac{1}{2}} = \frac{\Delta t^{n+1}}{\Delta t^{n+1} + \Delta t^n}, \quad \tau^{n-\frac{1}{2}} = \frac{\Delta t^n}{\Delta t^n + \Delta t^{n-1}}. \tag{3.16}$$

3.1.3. Parallel processing

ANSYS Fluent supports parallel processing by splitting the computational domain into multiple partitions, where each partition is assigned to a separate compute node. This parallel architecture allows simulations to run on multiple CPU logical processors within the same machine or across a network of computers. A diagram can be seen in Figure 3.4, [45]. The number of partitions is typically less than or equal to the number of processors or cores available. As the number of compute nodes increases, the overall simulation time decreases, reflecting the solver's scalability. However, excessive partitioning can lead to increased communication overhead between compute nodes, reducing parallel efficiency. For the numerical simulations, 6 partitions were selected in order to reduce simulation time.

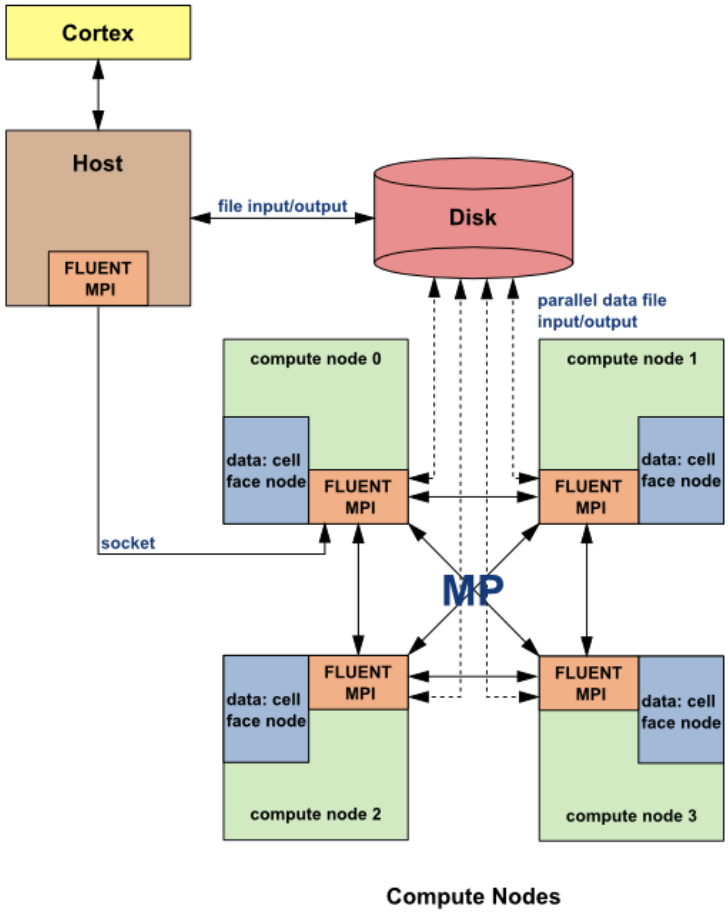


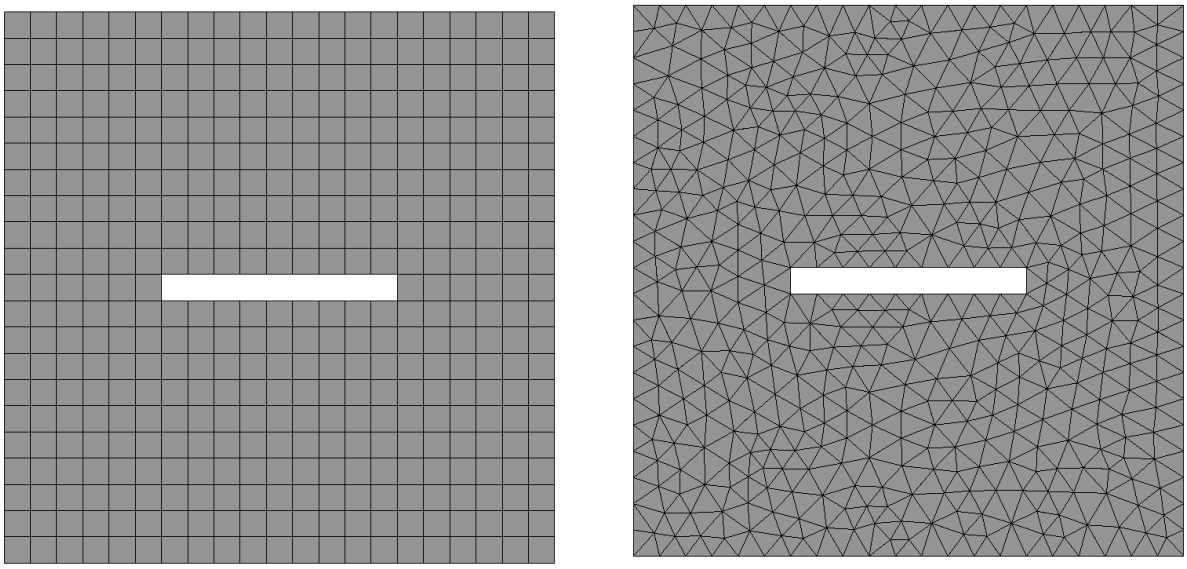
Figure 3.4: ANSYS Fluent architecture, [45].

3.2. Mesh

To discretize space, a mesh is created. The points on the grid are then used to determine the numerical values of the variables, [51]. In CFD simulations, various types of grids are used. The following are some of the primary grid types:

- Structured Grids: cells are topologically similar and all cells have the same number and type of connection
- Unstructured grids: arbitrary topology of the neighbouring cells, no rule for connectivity.
- Hybrid grids: a combination of the two. Usually used on the interface of the geometries.

Both structured and unstructured grids are presented in Figure 3.5. The example shows a mesh around a simple flat plate in a Cartesian coordinate system. As for the MEEVC structured grid is used, the ANSYS Fluent mesh will be structured as well.



(a) Structured grid. **(b)** Unstructured grid.

Figure 3.5: Computational grid for 2D flat plate.

ANSYS FLUENT uses a collocated scheme so both pressure and velocity are stored at cell centers, [45]. A collocated grid requires less computer memory to store variables as they are accessible at the same location, [52]. However, they can cause pressure field oscillations. They are addressed with Rhie and Chow momentum interpolation method proposed in [53]. It resolves pressure checkerboard problem, [52]. The collocated grid can be seen in Figure 3.6

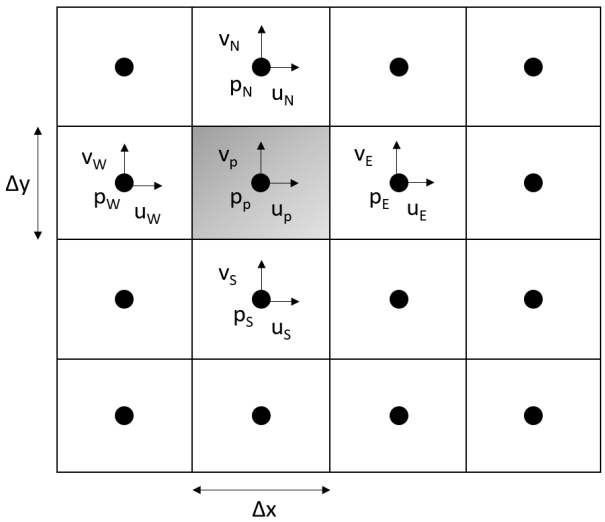


Figure 3.6: Collocated grid.

In a two-dimensional setup, there are 3 unknowns pressure p and velocity u with two components (x and y direction). They are stored at cell centres the degrees of freedom for ANSYS are as follows:

$$\text{DoF} = 3 \cdot K_{xy}. \tag{3.17}$$

with K_{xy} being the total number od cells.

3.3. Convergence criteria

The convergence of the simulation is a subjective matter. However, according to the user guide the target of the RMS residuals is below 10^{-6}. Acceptable values can still include those between 10^{-5} and 10^{-6}, even if they are not always attainable. It is summarized in Table 3.1, [45].

Table 3.1: RMS Residual Levels

RMS value	**Meaning**
$> 10^{-4}$	May be sufficient to obtain a qualitative understanding of the flow field.
$< 10^{-4}$	Relatively loose convergence, but may be sufficient for many engineering applications.
$< 10^{-5}$	Good convergence and usually sufficient for most engineering applications.
$< 10^{-6}$	Very tight convergence and occasionally required for geometrically sensitive problems. It is often not possible to achieve this level of convergence, particularly when using a single precision solver.

4

Numerical simulations

This chapter deals with the numerical simulations carried out for comparison. The numerical methods used in this study include the MSEM using mass, energy, enstrophy and vorticity conserving (MEEVC) solver and traditional Computational Fluid Dynamics (CFD) techniques represented by ANSYS Fluent software. The aim is to show the capabilities and limitations of these methods. Four test cases are presented: Taylor-Grenn Vortex in Section 4.1, lid cavity flow in Section 4.2, backward-facing step in Section 4.3 and flow around a circular cylinder in Section 4.4. The chapter ends with a cross-case comparison study in Section 4.5.

4.1. Test case 1: Taylor-Green Vortex

The Taylor-Green vortex is a classical, well-studied problem in fluid dynamics that serves as a useful test case for numerical methods in computational fluid dynamics. It involves simulating an idealized flow structure composed of vortex pairs, first introduced by G.I. Taylor and A.E. Green in 1937, [54]. The initial velocity field for the Taylor-Green vortex is typically prescribed in a way that generates symmetric, alternating regions of positive and negative vorticity which evolve with time. The main purpose of this test case is to compare the accuracy of both simulation methods.

The domain is $\Omega \in [0, L]^2$ with $L = 2\pi$ and is illustrated in Figure 4.1. The mesh is evenly distributed on the domain with K element in each axis so K^2 elements in total. The coordinate system origin is located in the left bottom corner of the domain Ω. The periodic boundary conditions allow the Taylor-Green vortex to develop symmetrically across the domain without the influence of boundary walls. A square grid was selected because the Taylor-Green vortex problem does not involve particularly complex regions that would require mesh refinement. The flow features are distributed uniformly across the domain, making it appropriate to use a uniformly spaced grid, as the entire domain is of equal interest.

The Taylor–Green vortex analytical solution becomes as follows:

$$u_x(x, y, t) = \sin(x)\cos(y)e^{-2\nu t} \tag{4.1a}$$

$$u_y(x, y, t) = -\cos(x)\sin(y)e^{-2\nu t} \tag{4.1b}$$

$$\omega(x, y, t) = \nabla \times u = 2 \cdot \sin(x)\sin(y)e^{-2\nu t} \tag{4.1c}$$

$$p(x, y, t) = \frac{1}{4}\left(\cos(2x) + \cos(2y)\right)e^{-4\nu t} \tag{4.1d}$$

where x and y are coordinates in 2D domain, t is a time and kinematic viscosity $\nu = 6.25 \cdot 10^{-4}$ so the simulated Reynolds number equals $Re = \frac{L_0 u_0}{\nu} = 1600$. $u_0 = 1$ is a magnitude of the oscillating vortices and $L_0 = \frac{L}{2\pi} = 1$.

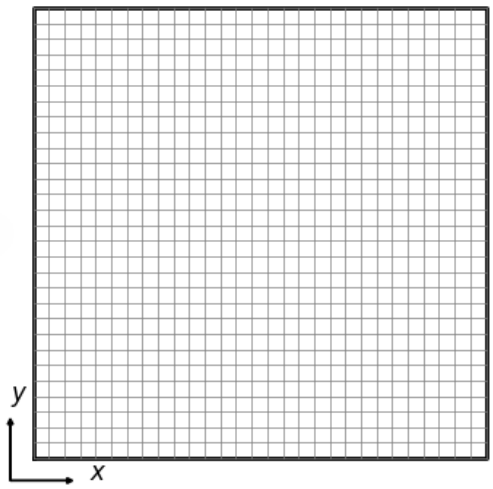


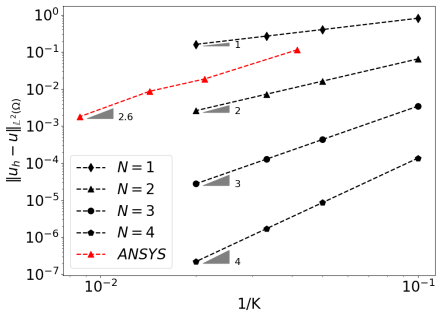
Figure 4.1: Domain Ω and mesh for a Taylor-Green Vortex with $K = 30$.

The initial conditions are prescribed on both velocity and vorticity. The flow is thus initially composed of four vortices, one in each quarter of the domain. For $t = 0$, Equation 4.1a, Equation 4.1b, Equation 4.1c and Equation 4.1d are simplified to

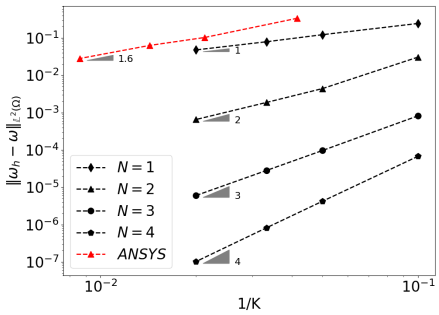
$$u_x(x, y, 0) = \sin(x)\cos(y) \tag{4.2a}$$
$$u_y(x, y, 0) = -\cos(x)\sin(y) \tag{4.2b}$$
$$\omega(x, y, 0) = 2 \cdot \sin(x)\sin(y) \tag{4.2c}$$
$$P(x, y, 0) = \frac{1}{4}(\cos(2x) + \cos(2y)) + \frac{1}{2}u \cdot u \tag{4.2d}$$

with P being the total pressure.

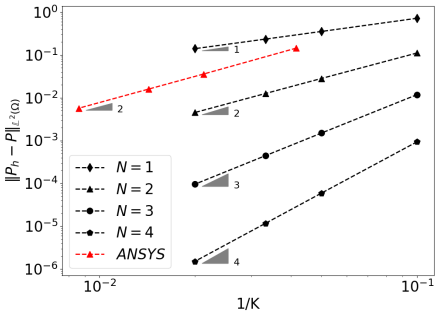
In Figure 4.2 L^2-error for velocity, vorticity and pressure field are presented. The errors are calculated after $t = 1$ with $\Delta t = \frac{1}{100}$. The number of mesh elements is $K \in \{10, 20, 30, 50\}$ with polynomial degrees $N \in \{1, 2, 3\}$ for MEEVC. As the number of grid points is increased so $\frac{1}{K}$ is lowered, the convergence with analytical solution is more visible. Additionally, the convergence rate is higher for higher-degree polynomials. This rate is indicated with a grey triangle next to the curve. It is approximately equal to the polynomial degree N so the error decreases with the power of N for finer gird. For ANSYS, the number of mesh elements is $K \in \{24, 47, 70, 116\}$. The numbers were chosen to match the Degrees of Freedom (DoF) for MEEVC with $N = 2$. The rate of convergence varies and is not as uniform across unknowns as in the MEEVC scheme. It equals around 2.6 for velocity u so it converges faster than MEEVC with $N = 2$, 1.6 for vorticity ω and 2 for pressure P. The lowest convergence for vorticity can be explained with the solver approach. It does not solve for the ω directly but is later calculated using the velocity field gradients. So the error does not only come from velocity field calculation but also from gradient approximation.



(a) Velocity u



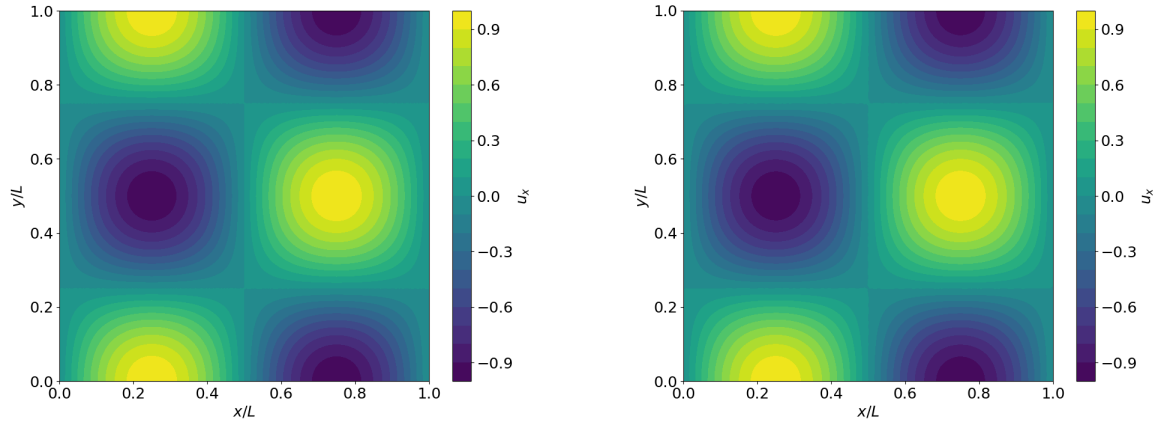
(b) Vorticity ω



(c) Pressure P

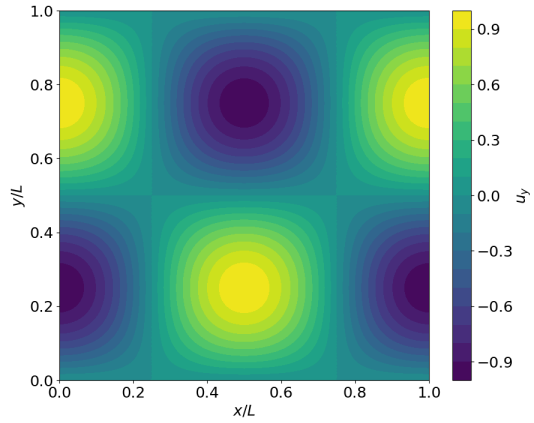
Figure 4.2: L^2-error for velocity, vorticity and pressure.

The velocity solution of the Taylor-Green vortex simulation, obtained using the MEEVC with $K = 30$ and ANSYS with $K = 105$, are presented in Figure 4.3. The polynomial degree for MEEVC is set to $N = 3$. The ANSYS grid resolution was selected to match the DoF$= 33000$ of the MEEVC scheme. The snapshots for both methods are taken at $t = 10$ with timestep $\Delta t = \frac{1}{100}$ being used. The velocities u_x and u_y fields display alternating regions of positive and negative values, consistent with the periodic and symmetric nature of the Taylor-Green vortex (see Equation 4.1a, Equation 4.1b). There are no visual discrepancies between the two methods.

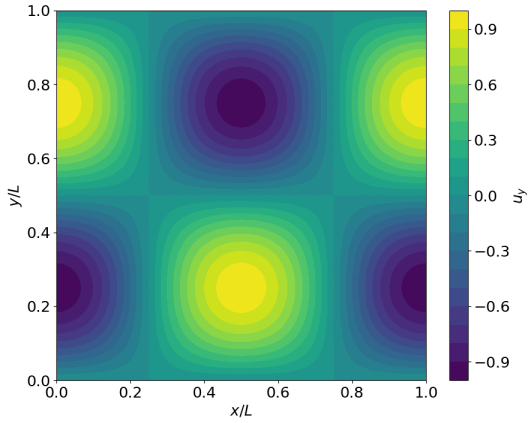


(a) MEEVC velocity u_x

(b) ANSYS velocity u_x



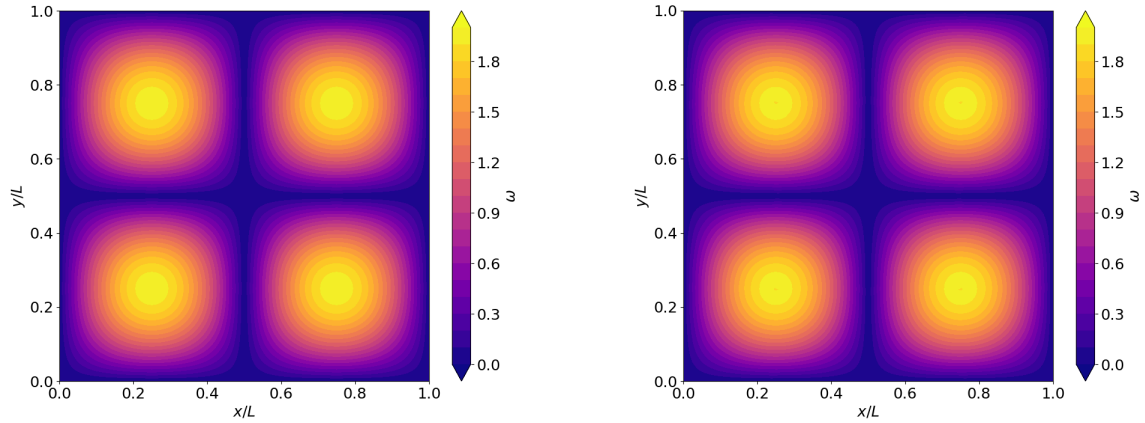
(c) MEEVC velocity u_y



(d) ANSYS velocity u_y

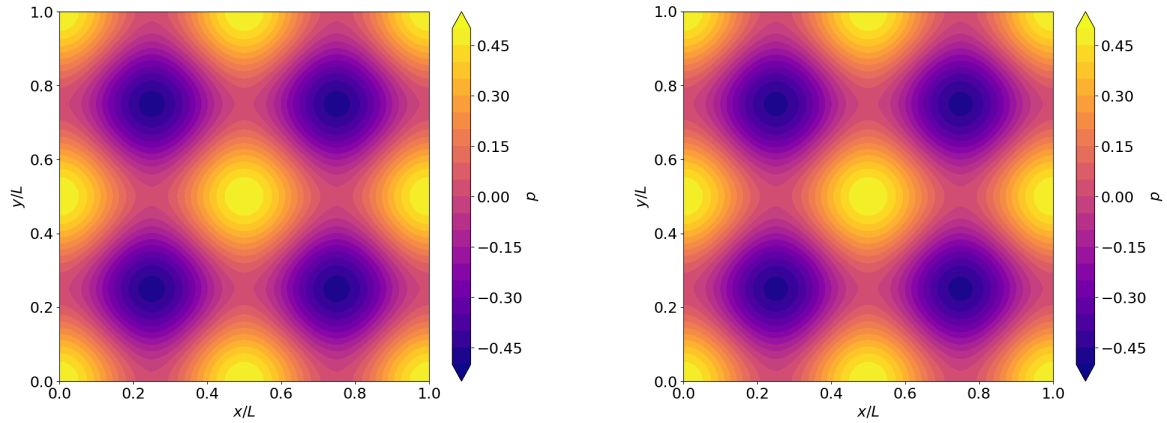
Figure 4.3: Velocity contour plots for MEEVC with $N = 3$ and $K = 30$ and ANSYS with $K = 105$ at $t = 10$.

The magnitude of vorticity ω contour plot for MEEVC and ANSYS can be seen in Figure 4.4. The ω highlights four distinct vortical structures within the domain. There is a discrepancy in the magnitude of vorticity in the middle of the aforementioned vertical structures. It will be later discussed when section plots are presented. The static pressure field p (see Figure 4.5), where the distribution of high and low-pressure regions corresponds directly to the vortex structures observed in the velocity and vorticity fields. Similarly to velocity, no direct differences are visible between the two methods.



(a) MEEVC **(b)** ANSYS

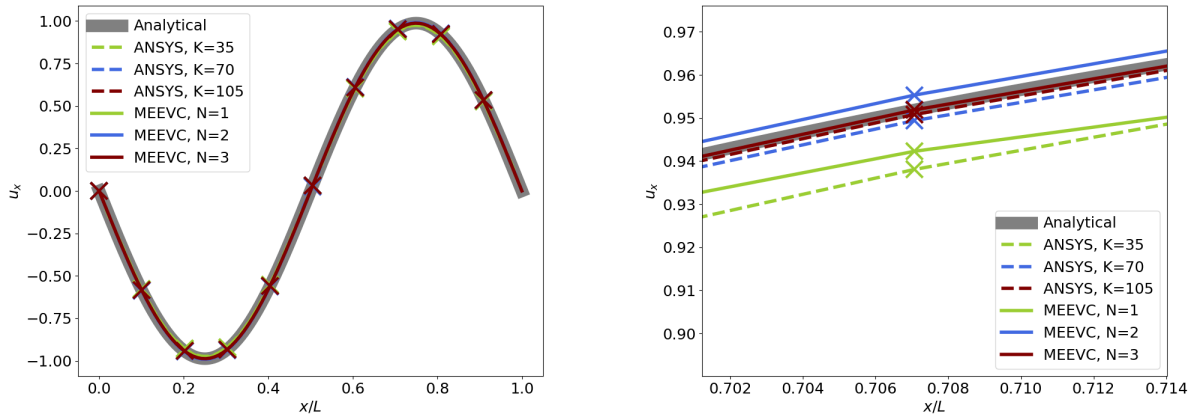
Figure 4.4: Vorticity ω contour plots for MEEVC with $N = 3$ and $K = 30$ and ANSYS with $K = 105$ at $t = 10$.



(a) MEEVC **(b)** ANSYS

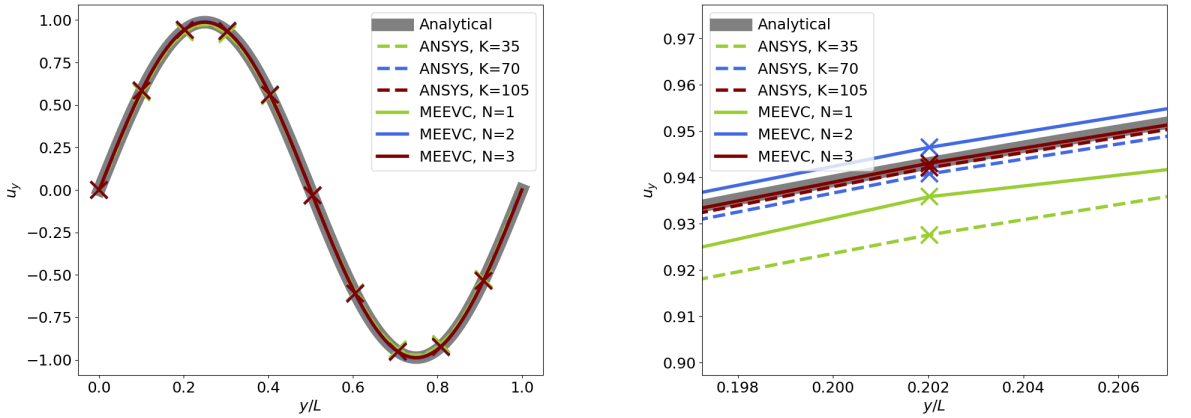
Figure 4.5: Static pressure p contour plots for MEEVC with $N = 3$ and $K = 30$ and ANSYS with $K = 105$ at $t = 10$.

The Figure 4.6 presents a comparison between the numerical solutions of MEEVC and ANSYS for the horizontal velocity component u_x along the centerline of the domain at $t = 10$ with the corresponding analytical solution. Similarly, the vertical velocity component is shown in Figure 4.7. For velocity u_x the location is $y = \pi$ and for velocity u_y it is $x = \pi$. The analytical solution is depicted as a solid grey line. The degree of the polynomial is in range $N \in \{1, 2, 3\}$ for MEEVC with $K = 30$. The mesh for ANSYS includes $K \in \{35, 70, 105\}$ in order to match MEEVC DoF for the respective polynomial degree N. The same DoF are indicated with one color on the plots. The figure demonstrates that as the degree is increased, the numerical solution converges towards the analytical solution. It is seen more clearly in Figure 4.6b and Figure 4.7b as the section of the plots are enlarged. MEEVC performs better for the same DoF compared to ANSYS, except for $N = 2$ where the values are close.



(a) Velocity u_x **(b)** Zoom on the section

Figure 4.6: Velocity u_x profiles at $y = \pi$ at $t = 10$.



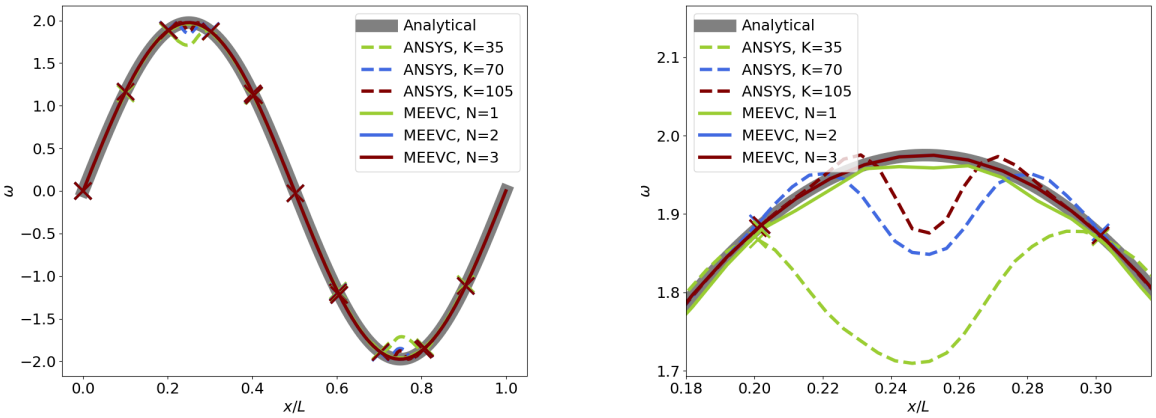
(a) Velocity u_y **(b)** Zoom on the section

Figure 4.7: Velocity u_y profiles at $x = \pi$ at $t = 10$.

The Figure 4.8 shows a comparison between the numerical solutions of MEEVC and ANSYS for the vorticity ω along $y = \frac{\pi}{2}$ at $t = 10$ with the corresponding analytical solution. Similarly to velocity, as the mesh resolution/polynomial degree is increased, the numerical solution converges towards the analytical solution. However, when zooming in on the peak at $x/L = 0.25$, there are some discrepancies in ANSYS data. The vorticity is not an unknown so it is not solved for in ANSYS. This might explain it. The error is from calculating velocity gradients as we can recall in 2D

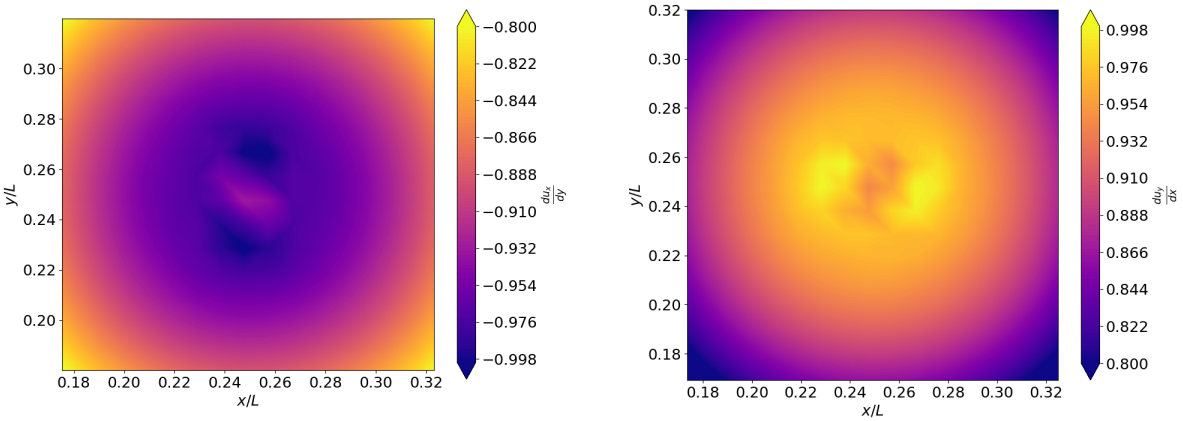
$$\omega = \nabla \times u = \frac{du_y}{dx} - \frac{du_x}{dy}. \tag{4.3}$$

Then, looking at the velocity gradient distribution in the region of interest, $x = \frac{\pi}{2}$ and $y = \frac{\pi}{2}$. The $\frac{du_y}{dx}$ and $\frac{du_y}{dx}$ are presented in Figure 4.9. Even though the most accurate discretization for gradients is used - least-squares method, ANSYS fails to capture accurately gradients in the middle of four vortices.



(a) Vorticity ω **(b)** Zoom on the section

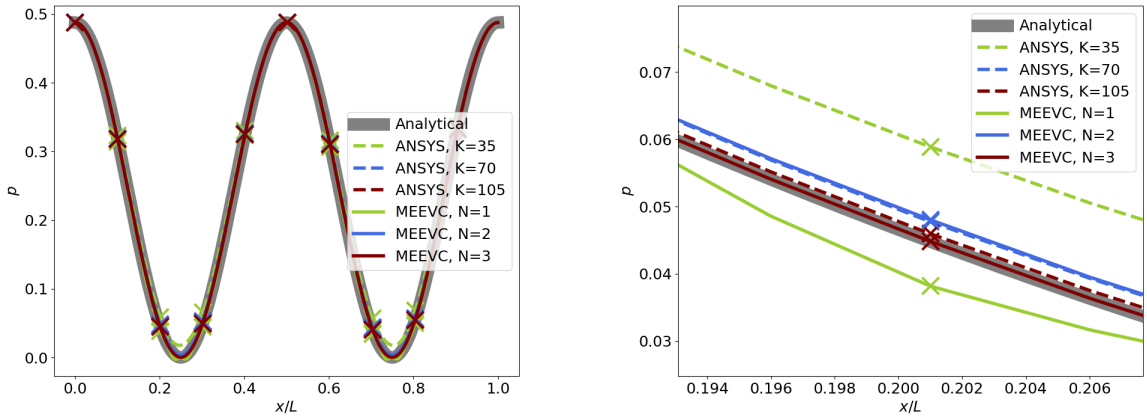
Figure 4.8: Vorticity ω profiles at $y = \frac{\pi}{2}$ at $t = 10$.



(a) $\frac{du_x}{dy}$ **(b)** $\frac{du_y}{dx}$

Figure 4.9: Velocity gradients at $x = \frac{\pi}{2}$, $y = \frac{\pi}{2}$ at $t = 10$.

Lastly, in Figure 4.10 a comparison between the numerical solutions of MEEVC and ANSYS for the static pressure p along the centerline of the domain at $t = 10$ with the corresponding analytical solution is shown. Similarly to velocity, the solution converges with increased resolution/polynomial degree. As already mentioned, MEEVC shows better accuracy for the same amount of DoF for the pressure as well.



(a) Static pressure p **(b)** Zoom on the section

Figure 4.10: Static pressure p profiles at $y = \pi$ at $t = 10$.

Another way of comparing both results was to calculate the maximum flow properties (u, ω, p) at a specific timestep and compare it with an analytical solution at $t = 10$. Table 4.1 presents the maximum velocity values ε_{u_x}, ε_{u_y}, vorticity ε_ω and static pressure ε_p for both ANSYS and MEEVC numerical methods compared with the analytical solution. The analytical solution yields a reference maximum velocity of 0.9875778, vorticity of 1.951555 and static pressure of 0.493789. The MEEVC method exhibits systematically increased accuracy with increasing values of N. Likewise, the ANSYS simulations are also seen getting more accurate as the meshes are refined. Both methods are compared with each other for the same amount of DoF. As already deduced from the section plots MEEVC performs better for a given DoF. What is presented in this section is that the two approaches are indeed converging to the analytical solution. It turns out that the most accurate result by far is produced by the MEEVC with $N = 3$.

Table 4.1: Maximum flow properties error for ANSYS and MEEVC at $t = 10$.

	DoF	ε_{u_x} [%]	ε_{u_y} [%]	ε_ω [%]	ε_p [%]
MEEVC, $N = 1$	3700	1.3379	1.3352	0.8462	0.0142
ANSYS, $K = 35$	3700	1.4750	1.4927	13.3876	0.1629
MEEVC, $N = 2$	14700	0.3641	0.3641	0.0002	0.0076
ANSYS, $K = 70$	14700	0.2453	0.2364	6.4262	0.0010
MEEVC, $N = 3$	33000	0.0007	0.0002	0.0009	0.0002
ANSYS, $K = 105$	33000	0.0637	0.0748	5.1036	0.0309

In order to see if conserving properties of MEEVC play a role as the simulation progresses, for DoF$= 33000$ simulation was carried for a bit longer to $t = 20$. Table 4.2 shows the findings of this approach. As can be seen, the absolute errors ε for MEEVC have not changed significantly compared to the state at $t = 10$. On the other hand, ANSYS errors have increased slightly. It confirms the hypothesis that MEEVC accuracy becomes more prominent as simulation time progresses due to its conserving properties.

Table 4.2: Maximum flow properties error for ANSYS and MEEVC at $t = 20$.

	DoF	ε_{u_x} [%]	ε_{u_y} [%]	ε_{ω} [%]	ε_{p} [%]
MEEVC, $N = 3$	33000	0.0007	0.0002	0.0009	0.0003
ANSYS, $K = 105$	33000	0.0853	0.0816	5.3314	0.0310

4.2. Test case 2: Lid cavity flow

The next test case chosen for comparison is lid-driven cavity flow. It involves the motion of a fluid within a square or rectangular cavity driven by the movement of the top boundary. Despite its seemingly straightforward setup, the flow characteristics within the cavity are highly sensitive to the Reynolds number. Therefore flow at $Re = 400$ and $Re = 1000$ were selected. The Reynolds number is defined as

$$Re = \frac{u_\infty \cdot L}{\nu} \tag{4.4}$$

where $u_\infty = 1$ is a tangential velocity on the top boundary, $L = 1$ is the length of the domain Ω and ν is kinematic viscosity calculated based on the Reynolds number. At low Reynolds numbers, the flow is characterized by a steady, symmetric primary vortex, while at higher Reynolds numbers, secondary and even tertiary vortices may emerge. To ensure that a steady-state solution was achieved for the MEEVC simulation, the proper stopping criterion needed to be introduced. The velocity residual $\frac{\|u_h^k - u_h^{k-1}\|_{L^2(\Omega)}}{\Delta t} \leq 1 \cdot 10^{-6}$ is used. The expression was divided by timestep to make the convergence criterion independent of it. Otherwise, it could have been met by decreasing timestep. The timestep of $\Delta t = \frac{1}{100}$ was set. For ANSYS simulation tight convergence was selected so RMS Residual Levels of 10^{-6}.

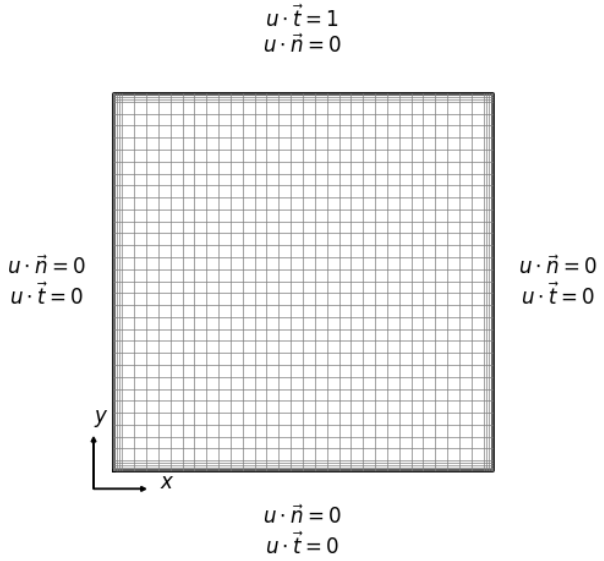


Figure 4.11: Domain Ω and mesh for a lid cavity flow with $K = 40$.

The flow domain for a lid-driven cavity problem is shown in Figure 4.11. The domain is a square cavity discretized with a structured grid, $\Omega \in [0, 1]^2$. The top boundary, or lid, is subject to a boundary condition where the non-dimensional tangential velocity component is set to 1 (in a positive x direction). The normal velocity component is zero, ensuring no flow across the boundary. The side and bottom boundaries are stationary, with no-slip conditions applied, meaning both the normal and tangential velocity components are zero. The coordinate system is located in the bottom left corner of the domain Ω similar to TGV test case. The mesh used for this case

is square mesh with refinement towards the domain edges ($K = 40$). It is used for MEEVC simulation but ANSYS mesh is refined to ensure the same number of DoF are compared.

A schematic diagram for lid cavity vortex formation can be seen in Figure 4.12, [55]. The primary vortex is a large recirculation region that dominates the flow pattern located near the center of the domain Ω. The secondary vortices appear in the bottom corners of the cavity, where the flow re-circulates due to the no-slip conditions at the walls. These secondary vortices are labelled using two letters and a number. The first letter denotes the position as either bottom (B) or top (T), while the second letter specifies right (R) or left (L). The number then indicates whether the vortex is secondary (1) or tertiary (2). For example, BR1 stands for the Secondary Bottom Right vortex.

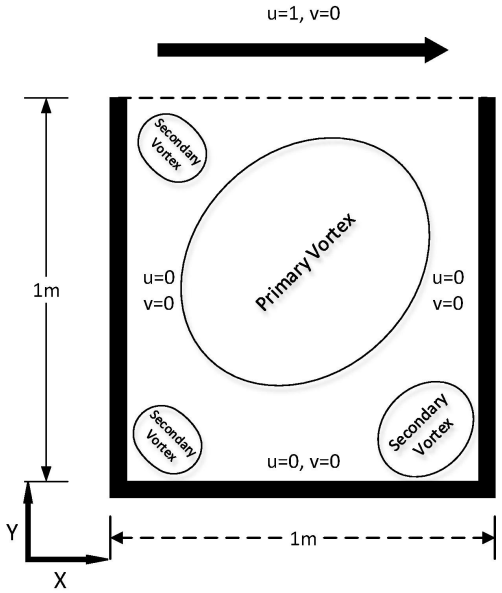
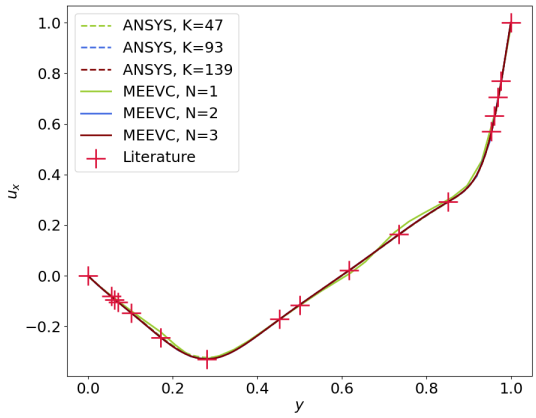
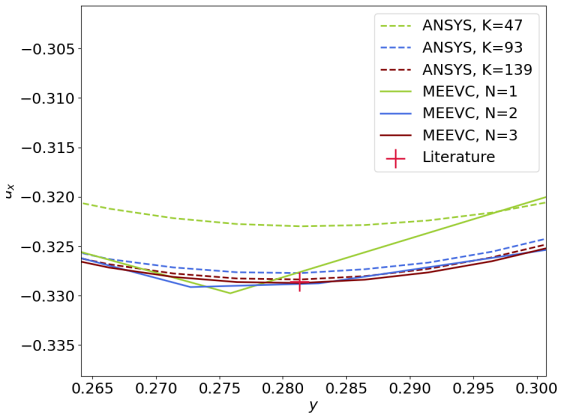


Figure 4.12: Schematic diagram for lid cavity vortex formation, [55].

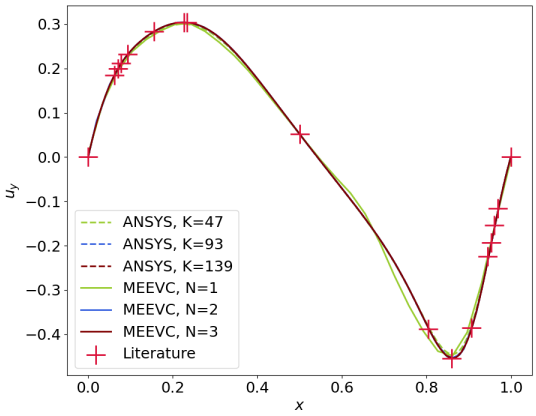
The set of section plots comparing the velocity components u_x and u_y and the vorticity magnitude ω for Reynolds number $Re = 400$ can be seen in Figure 4.13. The plots are shown for different polynomial degrees for MEEVC and mesh resolutions for ANSYS simulation, alongside reference literature values, [56]. The same DoF are indicated with the same color i.e. green for $DoF = 6600$, blue for $DoF = 25900$ and maroon for $DoF = 58000$. ANSYS results are shown with a dotted line and MEEVC with a solid line. Figure 4.13a shows the variation of the horizontal velocity u_x along the vertical centerline $x = 0.5$, comparing the results from MEEVC and ANSYS, respectively. Both methods demonstrate mesh convergence to the reference values. Figure 4.13c depict the vertical velocity u_y along the horizontal centerline $y = 0.5$. Both methods converge with increasing DoF, although MEEVC seems to be performing better for the same number of DoF. Finally, Figure 4.13e displays the magnitude of vorticity ω along the top boundary of the cavity $y = 1$. The logarithmic scale is used to capture the vorticity distribution, highlighting the convergence of the numerical solutions towards the literature values as the mesh is refined or the polynomial degree is increased. All the section plots are enlarged to magnify the differences between the methods. The little spikes can be seen near the singularities ($x = 0$ and $x = 1$) for ANSYS. Its impact is decreased as the number of mesh cells increases. The vorticity is calculated from the velocity field and its gradients. A least-squares Gradient Scheme is used which assumes a smooth gradient across neighbouring cells. This is not the case at singular points in lid-driven cavity flow which may result in inaccurate gradient estimation.



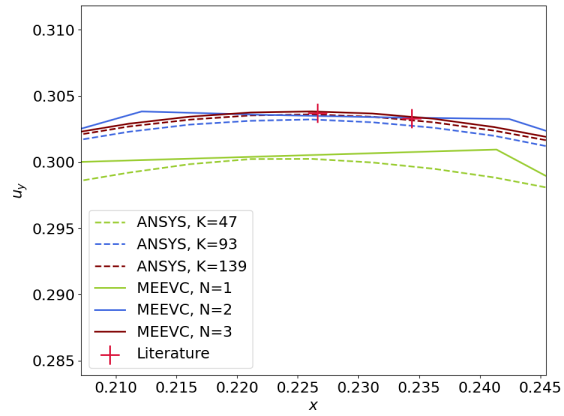
(a) u_x at $x = 0.5$.



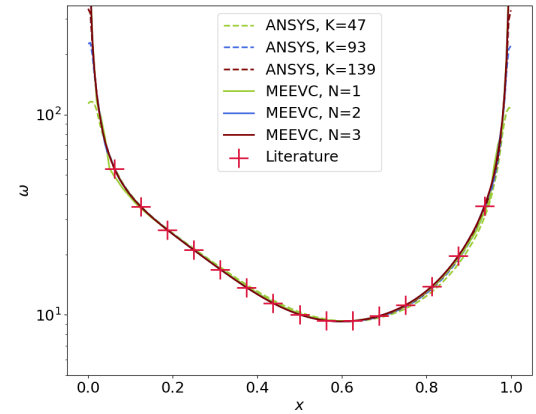
(b) Enlarged view of u_x at $x = 0.5$.



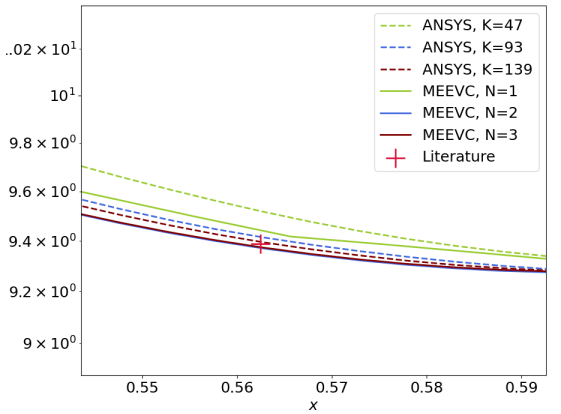
(c) u_y at $y = 0.5$.



(d) Enlarged view of u_y at $y = 0.5$.

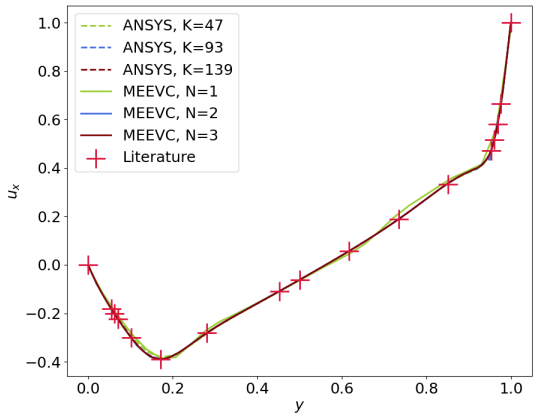


(e) ω magnitude at $y = 1$.

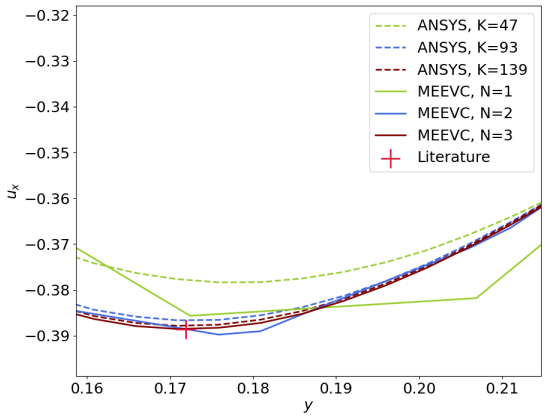


(f) Enlarged view of ω magnitude at $y = 1$.

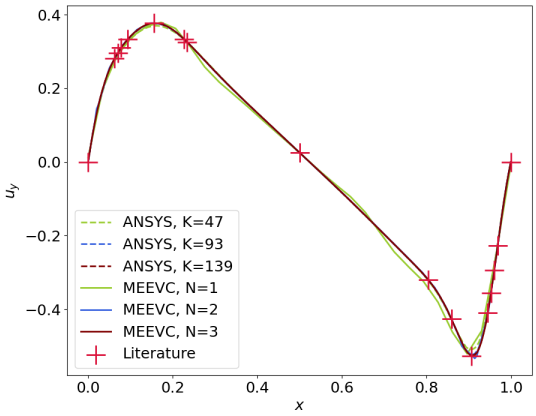
Figure 4.13: Section plots for u_x, u_y and ω at $Re = 400$, [56].



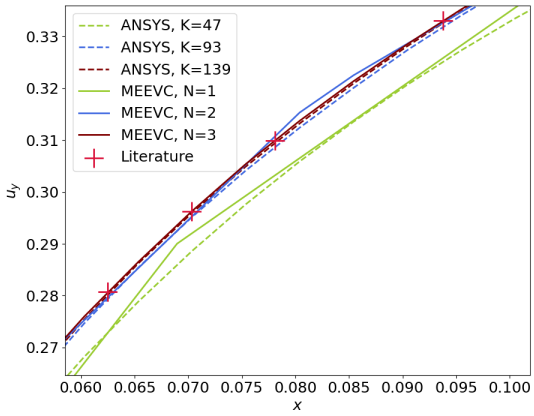
(a) u_x at $x = 0.5$.



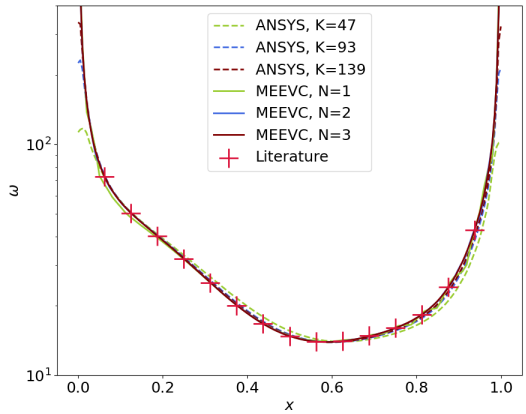
(b) Enlarged view of u_x at $x = 0.5$.



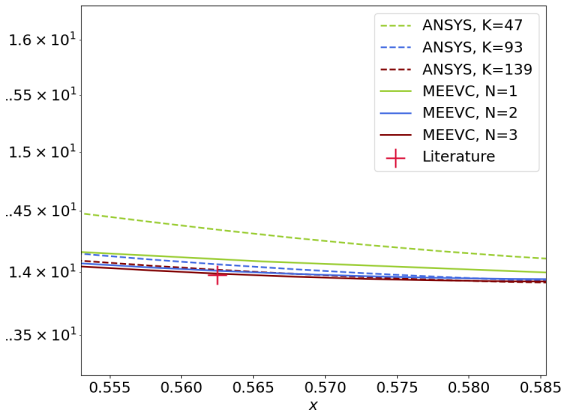
(c) u_y at $y = 0.5$.



(d) Enlarged view of u_y at $y = 0.5$.



(e) ω magnitude at $y = 1$.



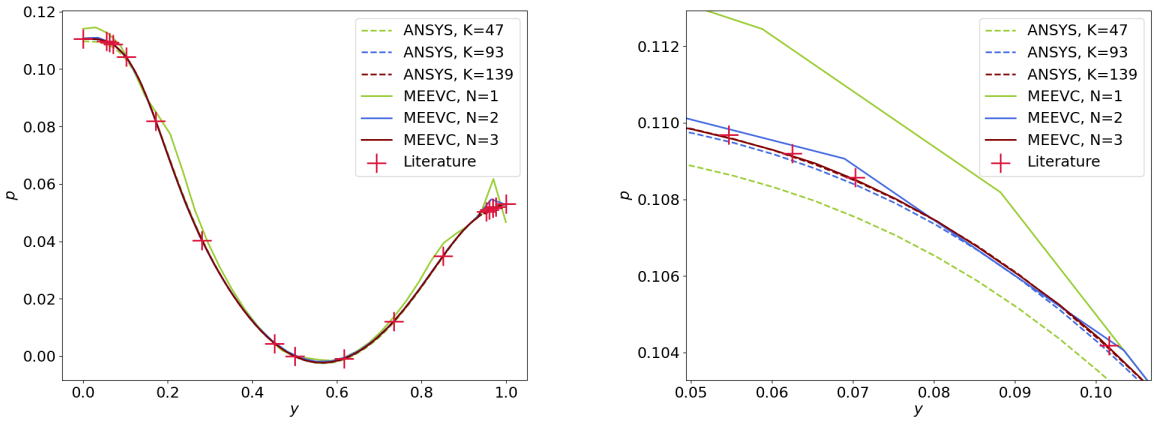
(f) Enlarged view of ω magnitude at $y = 1$.

Figure 4.14: Section plots for u_x, u_y and ω at $Re = 1000$, [57].

The Figure 4.14 presents a set of section plots comparing the horizontal velocity component u_x, vertical velocity component u_y and vorticity ω for Reynolds number $Re = 1000$. The same number of DoF is used as for the case of $Re = 400$. However, data from Botella [57] is used

as reference values as it provides also section values for static pressure p. The plots indicate that both numerical methods closely match the literature values. For ANSYS simulation, the little spike can be seen near $x = 1$ at vorticity plot Figure 4.14e. They are more prominent than in the case of $Re = 400$. Similarly to the previous case, values converge to literature values as DoF is increased.

Additionally, for $Re = 1000$ case pressure data was available. The section plot of static pressure p at $x = 0.5$ is illustrated in Figure 4.15. As solutions had different reference values, static pressure at centre of domain $x = 0.5$, $y = 0.5$, was set at 0. Both methods converge as the DoF is increased and the errors to reference are very close between the two.



(a) p at $x = 0.5$.

(b) Enlarged view of p at $x = 0.5$.

Figure 4.15: Section plots for static pressure p at $Re = 1000$, [57].

The Root Mean Squared Error (RMSE) provides a measure of the differences between values predicted by a model and the actual observed values. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.5}$$

where n is a number of data points, y_i represents the actual value and $\hat{y}_i$ represents the reference value.

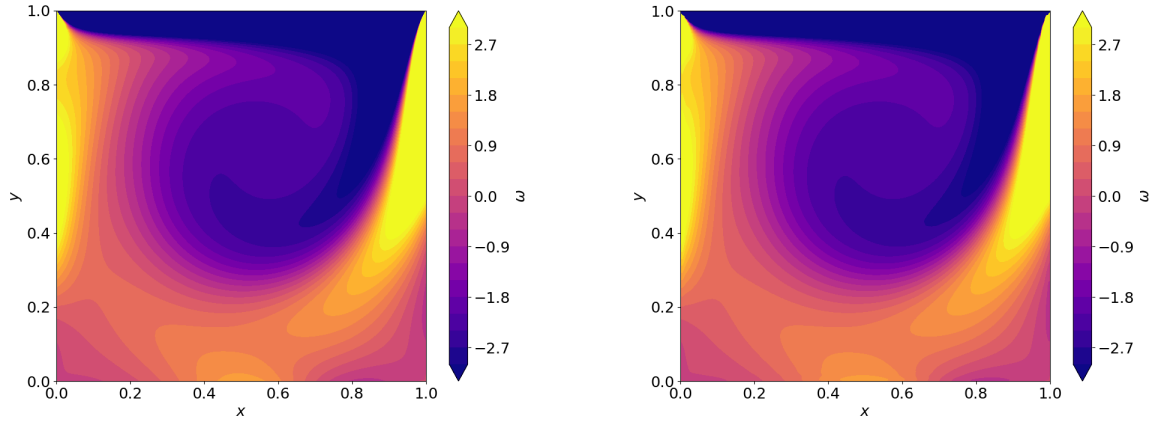
Table 4.3 presents the Root Mean Squared Error (RMSE) for both the MEEVC and ANSYS simulations, compared against reference values from [56] for $Re = 400$ and [57] for $Re = 1000$ presented in the section plots before. The RMSE values for vorticity (ω) are notably higher due to the generally larger magnitude of vorticity values which results in a correspondingly greater absolute error. Consequently, the RMSE for vorticity exceeds that of the velocity components u_x and u_y. The error for static pressure RMSE_p was done only for $Re = 1000$ as the data was unavailable for $Re = 400$. In the case of the MEEVC scheme, further increasing N to 3 yielded only marginal improvements in RMSE. ANSYS shows lower RMSE values for lower DoF. However, at $N = 2$ the MEEVC and ANSYS give similar errors for velocity. The vorticity error RMSE_ω is consistently lower for MEEVC by an order of magnitude at higher DoF. On the other hand, ANSYS produces smaller errors in static pressures for a given DoF. A coupled solver ensures that both velocity and pressure are resolved as one system which proved to be superior to the segregated solution schemes, [45].

Table 4.3: RMSE for MEEVC and ANSYS against reference values. [56, 57]

	DoF	RMSE$_{u_x}$	RMSE$_{u_y}$	RMSE$_{\omega}$	RMSE$_{p}$
		$Re = 400$			
MEEVC, $N = 1$	6600	0.03214	0.01301	1.0571	–
ANSYS, $K = 46$	6600	0.00743	0.00473	1.29247	–
MEEVC, $N = 2$	25900	0.00608	0.00455	0.06942	–
ANSYS, $K = 93$	25900	0.00657	0.00414	0.64373	–
MEEVC, $N = 3$	58000	0.00641	0.00438	0.04072	–
ANSYS, $K = 139$	58000	0.00646	0.00424	0.42300	–
		$Re = 1000$			
MEEVC, $N = 1$	6600	0.03183	0.03313	1.56999	0.00567
ANSYS, $K = 46$	6600	0.00528	0.00863	2.59503	0.00052
MEEVC, $N = 2$	25900	0.00509	0.00206	0.11548	0.00114
ANSYS, $K = 93$	25900	0.00133	0.00203	1.33846	0.00011
MEEVC, $N = 3$	58000	0.00089	0.00045	0.03636	0.00024
ANSYS, $K = 139$	58000	0.00092	0.00085	0.86331	0.00005

The Figure 4.16 and Figure 4.17 compare the vorticity field ω for a lid-driven cavity flow at Reynolds number $Re = 400$ and $Re = 1000$ respectively. Both plots display the vorticity distribution within the cavity, with values ranging from approximately -3 to 3 with 21 increments. The results are visually similar between the two methods. However, it is worth noting that both vorticity plots for ANSYS have a flattening near the top right corner (1,1). It is also visible on the section plots for vorticity ω at $y = 1$ where MEEVC approaches ∞ at (0,1) and (1,1) whereas ANSYS flattens at a certain point.

The Table 4.4 presents the main properties of primary and secondary vortices for a lid-driven cavity flow at a Reynolds number $Re = 400$. It compares the results obtained from three sources: Ghia's Finite Difference method with 257^2 gird [56], Abdelmigid's Finite Volume method with 1301^2 staggered gird [55] and Vanka's Finite Difference method with 321^2 gird [58]. The properties listed include the stream function Ψ, vorticity ω and the coordinates (x,y) of the vortex center for the primary vortex, as well as for the secondary/tertiary vortices labelled BR1, BR2 and BL1. The stream function Ψ was calculated using a given velocity field u by Runge-Kutta integration (see Appendix A). The Primary Vortex properties are in good agreement with the references, with slight variations in the values of Ψ and ω and a close agreement in the vortex center coordinates. For BR1 and BL1, the secondary vortices located at the bottom right and bottom left, the stream function Ψ shows minor differences across the sources, with vorticity ω and coordinates (x, y) also showing variations but maintaining similar trends. The BR2 vortex, a tertiary vortex, exhibits greater variability in results as it is much smaller. The values are near zero and some references don't even examine this area of the lid cavity. Similarly, the main properties of primary and secondary vortices for a lid-driven cavity flow at a Reynolds number $Re = 1000$ are shown in Table 4.5. The reference data consists of Abdelmigid's Finite Volume method with 1301^2 staggered gird [55], Botella's Chebyshev collocation method with a polynomial of degree $N = 160$ [57] and Erturk's Finite Difference method with 601^2 grid [59].

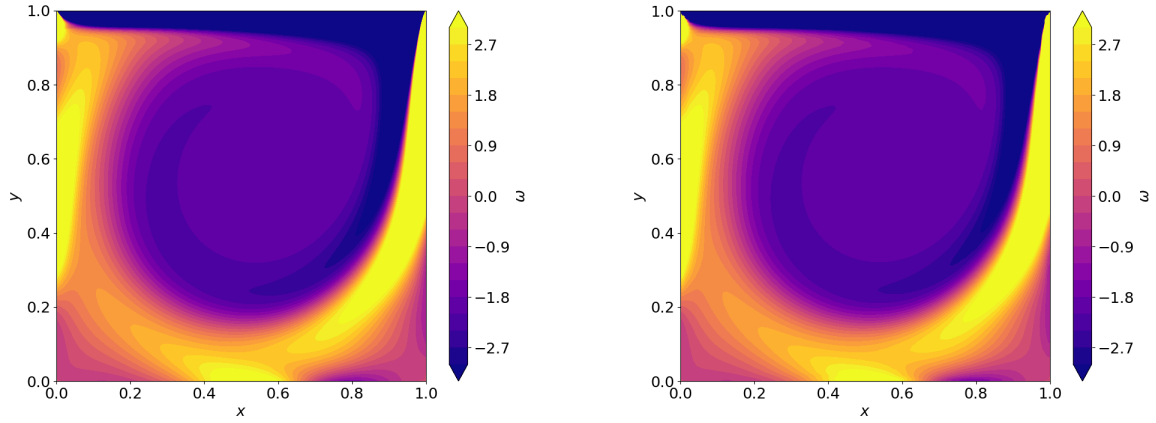


(a) MEEVC **(b)** ANSYS

Figure 4.16: MEEVC and ANSYS vorticity ω field for $Re = 400$.

Table 4.4: Primary and secondary vortices main properties at $Re = 400$ for DoF$= 58000$, [55, 56, 58].

Reference	MEEVC	ANSYS	Abdelmigid [55]	Ghia [56]	Vanka [58]
			Primary vortex		
Ψ	0.11404682	0.11398245	0.113964	0.113909	0.1136
ω	2.29522	2.294428	2.295985	2.281	–
(x, y)	$(0.5543, 0.6052)$	$(0.5543, 0.6052)$	$(0.5541, 0.6057)$	$(0.5547, 0.6055)$	$(0.5563, 0.6)$
			BR1		
Ψ	0.00064483	0.00064674	0.000645	0.00064235	0.000645
ω	0.440505	0.451267	0.44839	0.433519	-
(x, y)	$(0.8854, 0.1221)$	$(0.8865, 0.1236)$	$(0.8852, 0.1215)$	$(0.8906, 0.1250)$	$(0.8875, 0.1188)$
			BR2		
Ψ	1.7499×10^{-8}	1.4814×10^{-8}	1.94847×10^{-8}	1.86595×10^{-8}	–
ω	0.0034314	0.00277	0.003591	0.00438726	–
(x, y)	$(0.9929, 0.0072)$	$(0.9932, 0.0065)$	$(0.9917, 0.0067)$	$(0.9922, 0.0078)$	–
			BL1		
Ψ	1.4342×10^{-5}	1.4246×10^{-5}	1.443588×10^{-5}	1.41951×10^{-5}	1.46×10^{-5}
ω	0.0576284	0.0581056	0.05471926	0.0569697	–
(x, y)	$(0.0512, 0.0473)$	$(0.0509, 0.0471)$	$(0.0516, 0.0466)$	$(0.0508, 0.0469)$	$(0.05, 0.05)$



(a) MEEVC **(b)** ANSYS

Figure 4.17: MEEVC and ANSYS vorticity ω field for $Re = 1000$.

Table 4.5: Primary and secondary vortices main properties at $Re = 1000$ for DoF$= 58000$, [55, 57, 59].

Reference	MEEVC	ANSYS	Abdelmigid [55]	Botella [57]	Erturk [59]
			Primary vortex		
Ψ	0.11896562	0.11892122	0.118866	0.1189366	0.118781
ω	2.0662	2.0678	2.066581	2.067753	2.06553
(x, y)	$(0.5308, 0.5652)$	$(0.5306, 0.5658)$	$(0.5308, 0.5657)$	$(0.5308, 0.5625)$	$(0.53, 0.565)$
			BR1		
Ψ	0.00173185	0.00173127	0.001732	0.00175102	0.0017281
ω	1.1084	1.1055	1.113969	1.15465	1.115505
(x, y)	$(0.8641, 0.1118)$	$(0.8641, 0.1121)$	$(0.8636, 0.1115)$	$(0.8594, 0.1094)$	$(0.8633, 0.1117)$
			BR2		
Ψ	5.12721×10^{-8}	4.37631×10^{-8}	5.46775×10^{-8}	5.03944×10^{-8}	5.4962×10^{-8}
ω	0.0082027	0.0073583	0.006969814	—	0.0077076
(x, y)	$(0.9924, 0.0076)$	$(0.9928, 0.0074)$	$(0.9917, 0.0067)$	$(0.9917, 0.0067)$	$(0.9923, 0.0077)$
			BL1		
Ψ	0.00023359	0.000231149	0.000233412	0.000233453	0.00023261
ω	0.3489871	0.350122	0.3409262	0.3522861	0.353473
(x, y)	$(0.0831, 0.0778)$	$(0.0831, 0.0779)$	$(0.0832, 0.0782)$	$(0.0833, 0.0781)$	$(0.0833, 0.0783)$

The tables present a significant amount of data. To summarize the findings, it is crucial to calculate absolute error w.r.t. literature. For the case of $Re = 400$, the error for the primary vortex is similar for both methods, around 0.15%. ANSYS accurately predicts the vortices' location (slightly better than MEEVC) but the values for stream function Ψ and vorticity ω are off by a

bigger margin for BR1, BR2 and BL1. Therefore it is safe to say that MEEVC demonstrates a higher level of accuracy in capturing the secondary and tertiary vortices compared to ANSYS for the same number of DoF. In the case of $Re = 1000$, MEEVC is slightly more accurate in the secondary and tertiary vortices regions. However, the difference is not as significant as for $Re = 400$. Overall, the average ANSYS error for the location of vortex coordinates is lower $\varepsilon_{(x,y)_{ANSYS}} = 0.15\%$ than for MEEVC $\varepsilon_{(x,y)_{MEEVC}} = 0.4\%$. The average stream function error for MEEVC equals $\varepsilon_{\Psi_{ANSYS}} = 0.2\%$ and for ANSYS $\varepsilon_{\Psi_{ANSYS}} = 0.46\%$. Finally, the largest differences are seen in vorticity predictions - $\varepsilon_{\omega_{ANSYS}} = 0.84\%$ and $\varepsilon_{\omega_{ANSYS}} = 1.47\%$.

4.3. Test case 3: Backward-facing step

The next test case chosen for comparison is a backward-facing step. It involves fluid flow in a channel that suddenly expands in a cross-sectional area (step). As the fluid passes over the step, it typically experiences flow separation, leading to the formation of a recirculation zone downstream of the step. Then it reattaches further downstream.

The Reynolds number for the backward-facing step is defined as

$$Re = \frac{H \cdot \bar{u}_{inlet}}{\nu} \tag{4.6}$$

with $H = 2$ being the height of the whole domain (not just a step), $\bar{u}_{inlet} = 1$ is an average velocity at the inlet and ν is the dynamic viscosity of the fluid. No fluid density ρ is present in the equation. That's because the MEEVC uses a dimensionless form of the NS. Unfortunately, it is not possible to make a dimensionless simulation in ANSYS so the fluid density is set to $\rho_{ansys} = 1$ kg/m^3.

The domain for the backward-facing step simulation can be seen in Figure 4.18. The height of the domain is $H = 2$ and the step height is set to half of that so $h = 1$. It makes an expansion ratio of 2 which is comparable with the literature study of Erturk, [60]. The length of the inlet channel is equal to $L_{inlet} = 10$ and the whole domain has a span of $L_{outlet} = 70$. Boundary conditions are indicated as follows. At the inlet, the non-dimensional normal component of velocity is specified as u_{inlet} with zero tangential velocity, the top and bottom walls of the channel, the no-slip condition is applied. At the outlet, the static pressure is set to zero. The inlet velocity was defined as parabolic with an average value of $\bar{u}_{inlet} = 1$. The viscosity was calculated based on the aforementioned values. It is equal to $\nu = \frac{1}{50}$ for $Re = 100$.

The boundary conditions are as follows. The essential BC is imposed along the whole boundary. The normal velocity $u \cdot \vec{n}$ is prescribed at both inlet and outlet while tangential velocity is set to 0 along the boundary. The outlet is a pressure outlet where static pressure is equal to zero. As the flow is incompressible, we can assume that $p = P - \frac{1}{2}u^2$. The following inlet profile is prescribed:

$$u_{inlet}(y) = -6 \cdot (y-1) \cdot (y-2) \quad \Gamma\perp \tag{4.7}$$

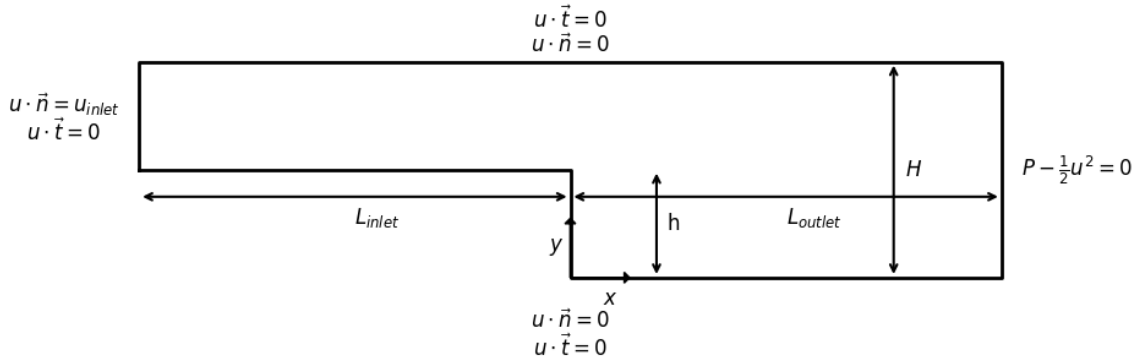
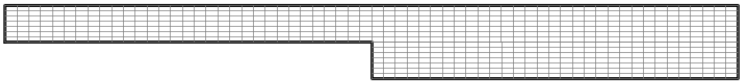


Figure 4.18: Domain Ω for a Backward-Facing Step.

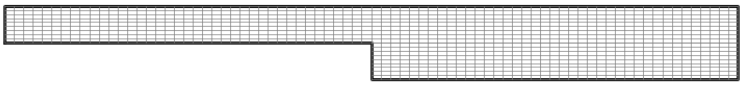
The final convergence criterion was set at $\frac{\|u_h^k - u_h^{k-1}\|_{L^2(\Omega)}}{\Delta t} \leq 10^{-6}$. The timestep used for the simulation was $\Delta t = \frac{1}{10}$. For ANSYS simulation tight convergence was selected so RMS Residual Levels of 10^{-6}.

The different meshes for the MEEVC are shown in Figure 4.19. Firstly, mesh A is presented in Figure 4.19a. The mesh size in the vertical direction is $h_y = 0.125$ and in the horizontal direction $h_x = 0.625$. Similarly to mesh A, mesh B consists of a rectangular grid shown in Figure 4.19b. Here $h_y = 0.1$ and $h_x = 0.5$. Mesh C which can be seen in Figure 4.19c, is refined near the walls and after the step until around $x = 24$. Figure 4.19d shows mesh D which is basically mesh C with a larger number of cells in the horizontal direction. The polynomial degree chosen for the mesh convergence test was $N = 2$.

A similar mesh convergence test was performed on the ANSYS. However, in order to maintain the same amount of Degrees of Freedom number of nodes was increased for each selective mesh. The shape and refinement regions were kept the same.

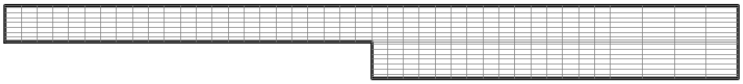


(a) Mesh A.



(b) Mesh B.

(c) Mesh C.



(d) Mesh D.

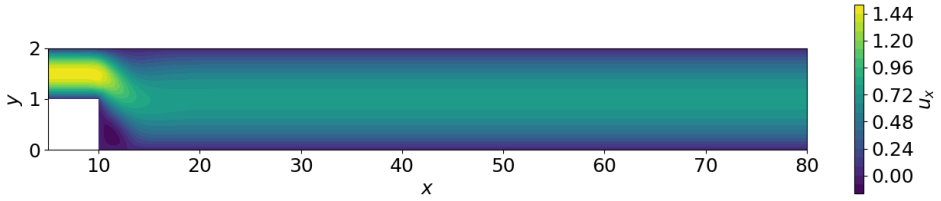
Figure 4.19: Different meshes used for mesh convergence test.

Table 4.6 shows relative error for the reattachment length for each mesh used. The values are compared to [60] with a reattachment length of $L_r = 2.922$. Mesh C was chosen for both methods as refining the grid to Mesh D does not reduce the error as significantly.

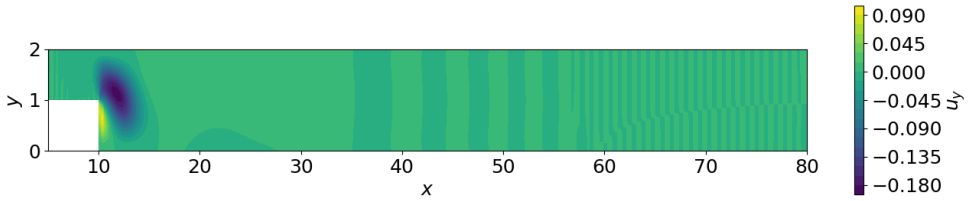
Table 4.6: Error in reattachment length estimation for each mesh.

	Mesh A	Mesh B	Mesh C	Mesh D
MEEVC	2.16%	1.57%	0.22%	0.07%
ANSYS	2.81%	2.57%	0.82%	0.72%

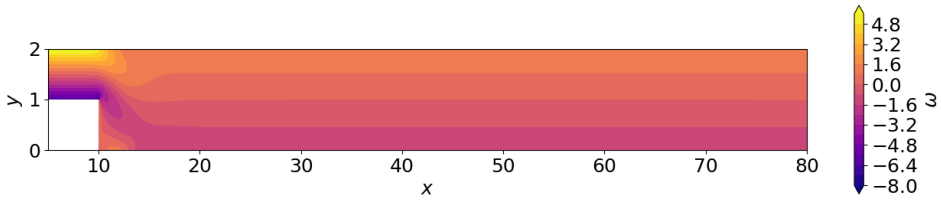
For $Re = 100$ velocity, vorticity and stream function contour plots are presented in Figure 4.20 for MEEVC and in Figure 4.21 for ANSYS. The same procedure was implemented for the $Re = 800$ case. The results of this simulation are shown in Figure 4.22 and Figure 4.23. In both cases, all plots represent similar contours and range of values. The DoF for the simulation was equal to around 32500.



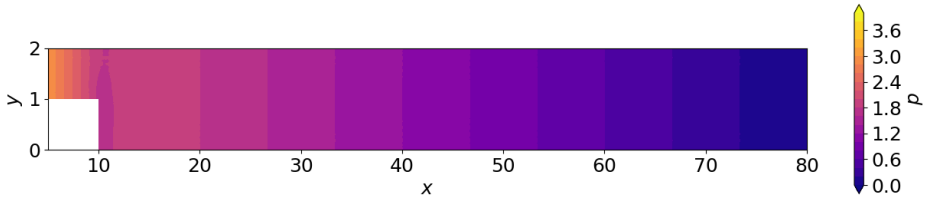
(a) Horizontal velocity u_x.



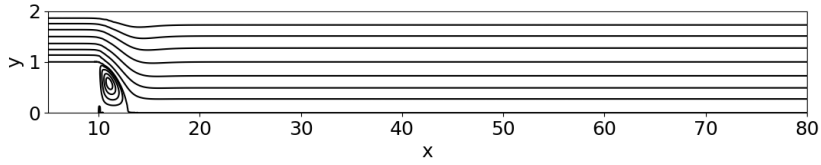
(b) Vertical velocity u_y.



(c) Vorticity ω.

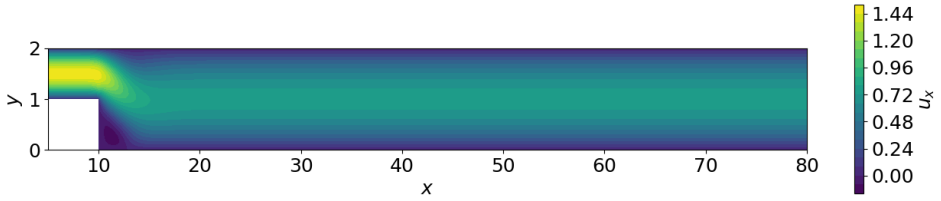


(d) Static pressure p.

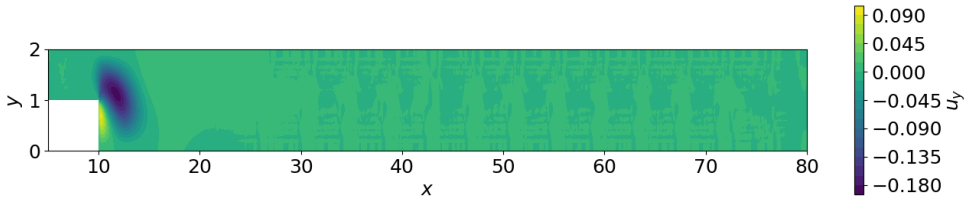


(e) Stream function Ψ.

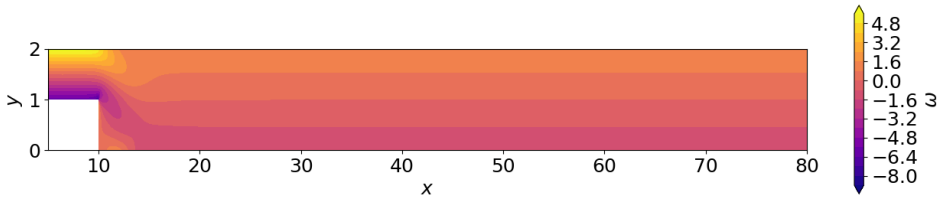
Figure 4.20: MEEVC contour plots for $Re = 100$ with mesh C.



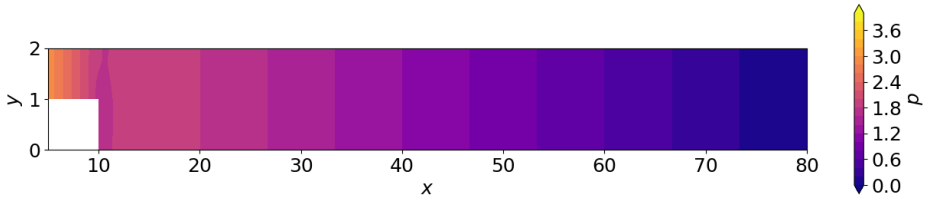
(a) Horizontal velocity u_x.



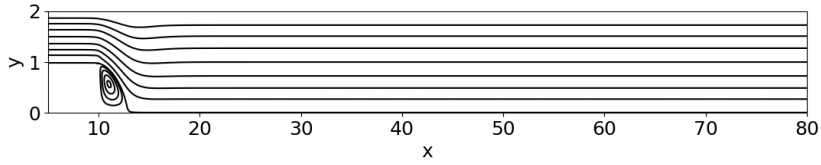
(b) Vertical velocity u_y.



(c) Vorticity ω.

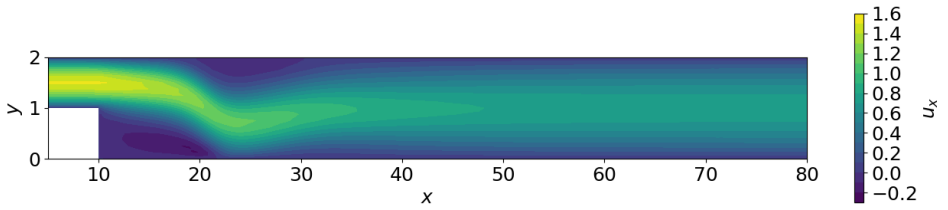


(d) Static pressure p.

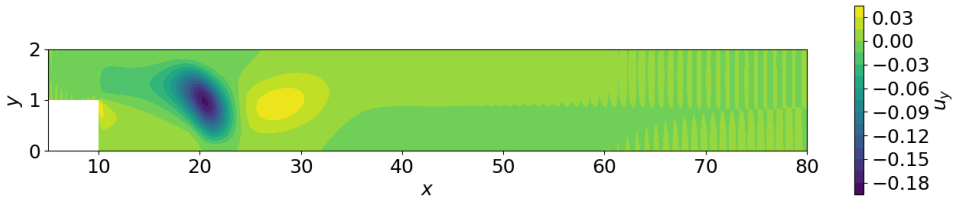


(e) Stream function Ψ.

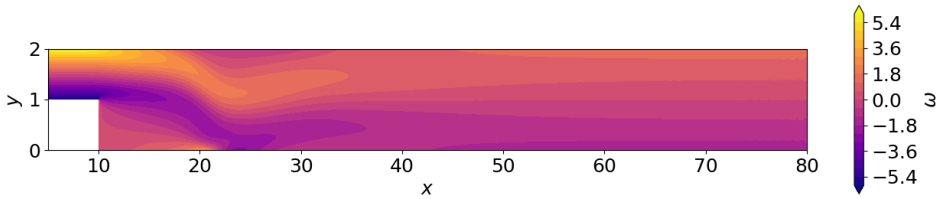
Figure 4.21: ANSYS contour plots for $Re = 100$ with mesh C.



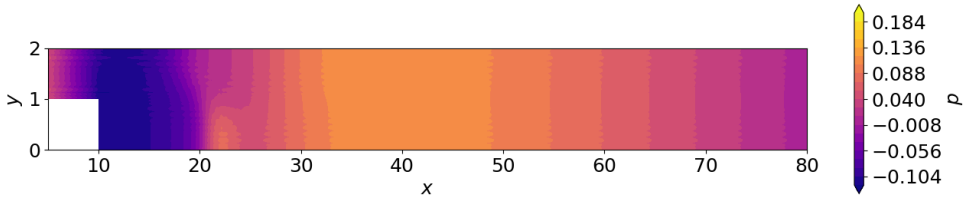
(a) Horizontal velocity u_x.



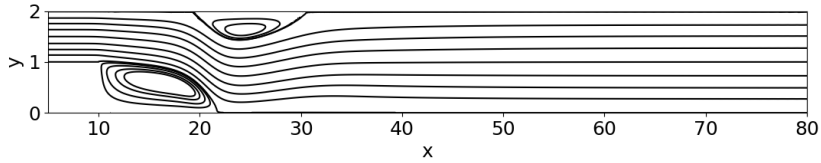
(b) Vertical velocity u_y.



(c) Vorticity ω.

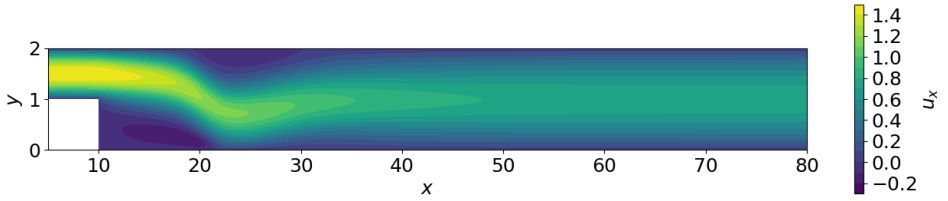


(d) Static pressure p.

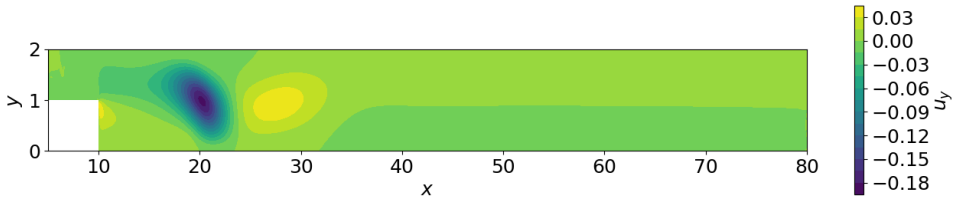


(e) Stream function Ψ.

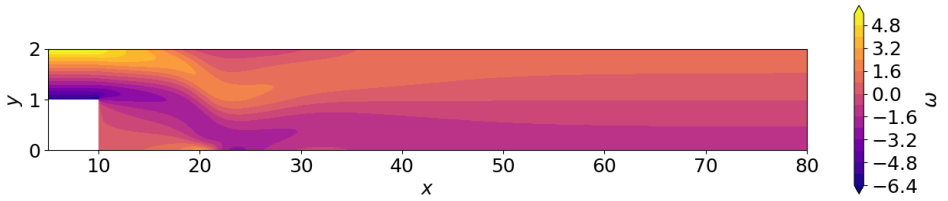
Figure 4.22: MEEVC contour plots for $Re = 800$ with mesh C.



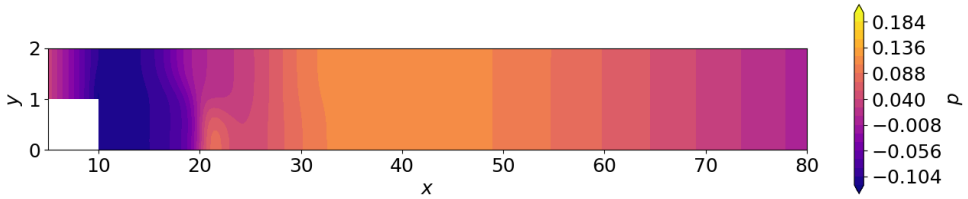
(a) Horizontal velocity u_x.



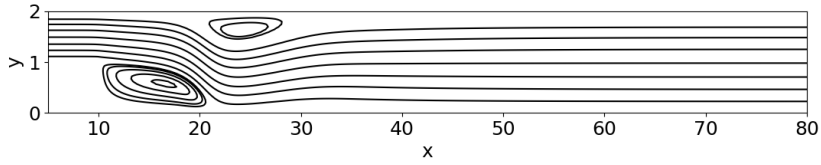
(b) Vertical velocity u_y.



(c) Vorticity ω.



(d) Static pressure p.

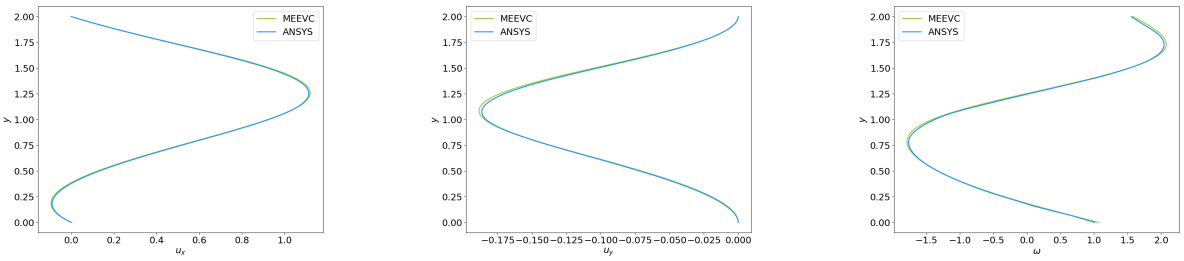


(e) Stream function Ψ.

Figure 4.23: ANSYS contour plots for $Re = 800$ with mesh C.

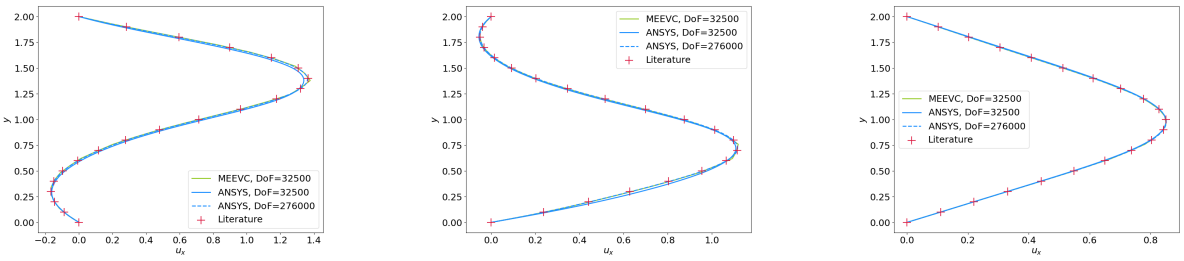
In order to validate both simulations reference literature data has to be introduced. Although numerous studies exist on flow over a backward-facing step, these studies provide very few tabulated results. Highly accurate numerical solutions of the 2-D steady incompressible backward-facing step flow are given in the study of Erturk, [60]. The data for velocity and vorticity profiles is only available for $Re = 800$. Therefore different segments were compared for both $Re = 100$ and $Re = 800$ as the reattachment points are different. $\frac{x}{h} = 12$ for $Re = 100$ and $\frac{x}{h} = 6$, $\frac{x}{h} = 14$ and $\frac{x}{h} = 30$ for $Re = 800$.

The results for $Re = 800$ are shown in Figure 4.25 for horizontal velocity u_x, in Figure 4.26 for vertical velocity u_y and in Figure 4.27 for vorticity ω. The values match the literature data. However, for $Re = 100$ (presented in Figure 4.24), the profiles are collected only at $\frac{x}{h} = 2$ as it is a region of flow recirculation. Both methods perform quite similarly. Unfortunately, no literature data was found for the case of $Re = 100$. All plots are made with mesh C and DoF $= 32500$, indicated with a solid line. The MEEVC shows better agreement with data from Erturk [60] for the same DoF. ANSYS converges to literature data and becomes comparable to MEEVC with mesh refinements. However, the required DoF are around 8.5 times higher - 276000. The most prominent disagreement with literature values is for $Re = 800$ at $\frac{x}{h} = 14$ for vertical velocity v profile. Various flow reattachment points might explain the difference. The flow is in different stages of development after reattachment, thus, fluctuations around that region.



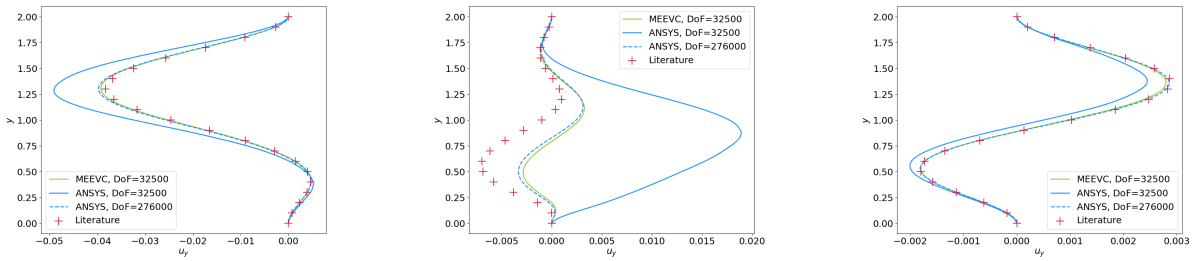
(a) Horizontal velocity u_x **(b)** Vertical velocity u_y **(c)** Vorticity ω.

Figure 4.24: Profiles for $Re = 100$ at $\frac{x}{h} = 2$.



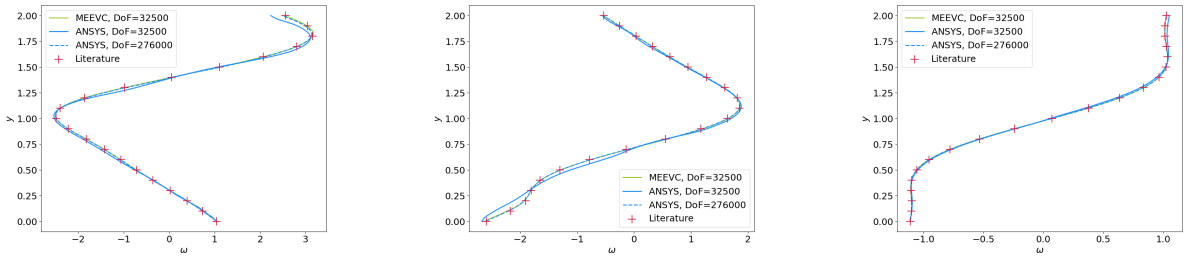
(a) $\frac{x}{h} = 6$. **(b)** $\frac{x}{h} = 14$. **(c)** $\frac{x}{h} = 30$.

Figure 4.25: Profiles for horizontal velocity u_x for $Re = 800$, [60].



(a) $\frac{x}{h} = 6.$ **(b)** $\frac{x}{h} = 14.$ **(c)** $\frac{x}{h} = 30.$

Figure 4.26: Profiles for vertical velocity u_y for $Re = 800$, [60].



(a) $\frac{x}{h} = 6.$ **(b)** $\frac{x}{h} = 14.$ **(c)** $\frac{x}{h} = 30.$

Figure 4.27: Profiles for vorticity ω for $Re = 800$, [60].

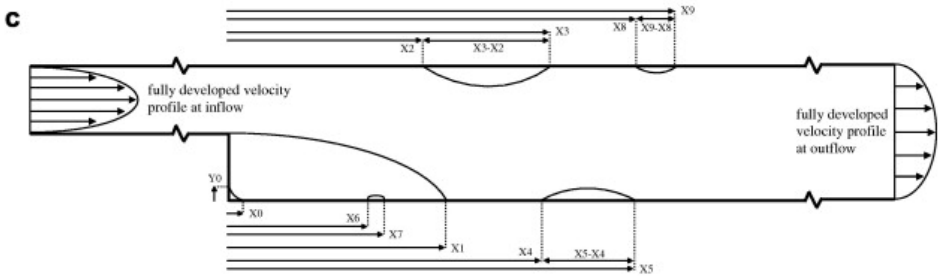


Figure 4.28: The reattachment lengths of the Backward-Facing step, [60].

As different the reattachment lengths explained fluctuations in section plots, they had to be compared. The Figure 4.28 shows all significant lengths in the backward-facing step. For smaller Reynolds numbers only $X1$ to $X3$ values are necessary as the other regions are simply not formed. The bottom reattachment length is denoted as $X1$ and the upper recirculating region's left and right positions are X2 and X3, respectively. The compared values can be found in Table 4.7. For $Re = 100$ Erturk data is shown [60]. For the case of $Re = 800$, three sources are presented: finite difference method by Erturk [60], Galerkin-based finite element method by Gartling [61] and pseudocompressibility method by Rogers [62]. The differences between the results at $Re = 100$ are very small, indicating close agreement between the methods. At $Re = 800$, there is slightly more variation. MEEVC reattachment length error w.r.t. literature data equals $\varepsilon_{X_1} \approx 0.83\%$. For ANSYS with the same DoF$= 32500$, this error increases to $\varepsilon_{X_1} \approx 6.57\%$. Firstly, it might mean that the solution didn't converge so RMS criteria were changed to residuals at 10^{-8}. It did not alter the solution for ANSYS with DoF$= 32500$. Therefore this significant

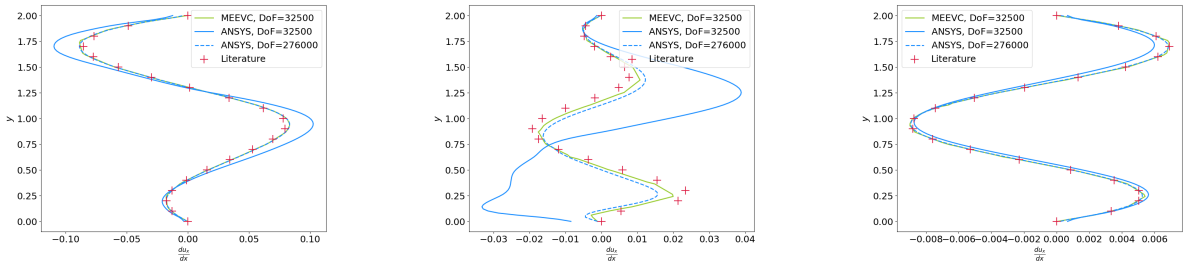
difference can only be compensated with mesh refinements. Only for DoF$= 276000$, so 8.5 times more nodes, ANSYS produces similar results to MEEVC.

Table 4.7: Reattachment lengths, [60, 61, 62].

	X_1	X_2	X_3	$X_3 - X_2$
	$Re = 100$			
MEEVC, DoF$= 32500$	2.915	–	–	–
ANSYS, DoF$= 32500$	2.898	–	–	–
Erturk [60]	2.922	–	–	–
	$Re = 800$			
MEEVC, DoF$= 32500$	11.74	9.38	20.54	11.16
ANSYS, DoF$= 32500$	11.06	8.82	20.27	11.45
ANSYS, DoF$= 276000$	11.72	9.32	20.55	11.23
Erturk [60]	11.834	9.476	20.553	11.077
Gartling [61]	12.20	9.70	20.96	11.26
Rogers and Kwak [62]	11.48	9.15	20.26	11.11

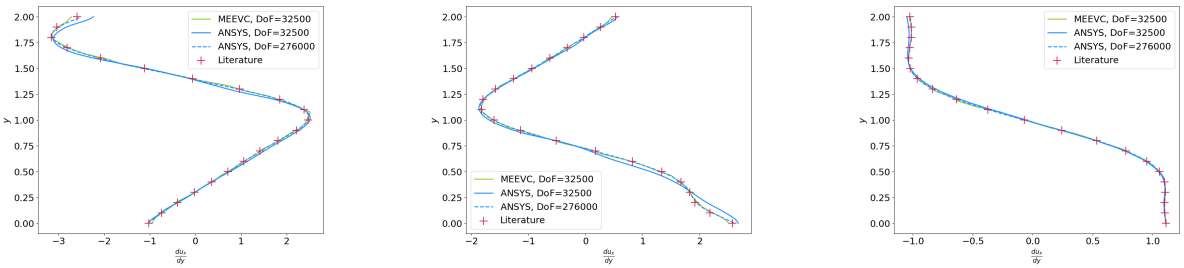
For the test $N = 3$ was used in the MEEVC scheme to see if the accuracy can be further increased. However, numerous simulations provided unexpected outcomes. Despite a few adjustments, the solution did not converge to a steady-state. The results can be seen in Appendix B.

The velocity gradients are shown in Figure 4.29, Figure 4.30, Figure 4.31 and Figure 4.32, [60]. MEEVC shows good agreement but with some small deviations for $\frac{x}{h} = 14$. This deviation might be explained by a slightly underestimated reattachment length. As flow attaches further downstream, the velocities gradients will differ as the flow is in a different stage of development at a certain cross-section. ANSYS errors are higher per the same amount of DoF. The ANSYS with DoF$= 32500$ is way off for $\frac{x}{h} = 14$. As already established, the reattachment length error for ANSYS was around 6.57% which explains the deviation. When DoF are increased the solution converges to literature results by Erturk.



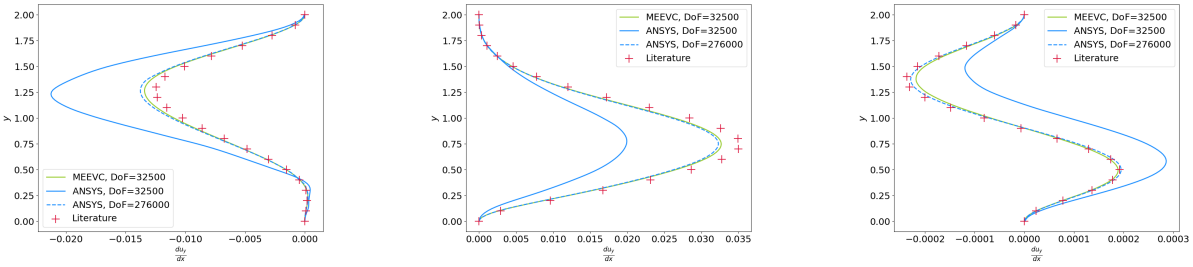
(a) $\frac{x}{h} = 6$. **(b)** $\frac{x}{h} = 14$. **(c)** $\frac{x}{h} = 30$.

Figure 4.29: Profiles for $\frac{du_x}{dx}$ for $Re = 800$, [60].



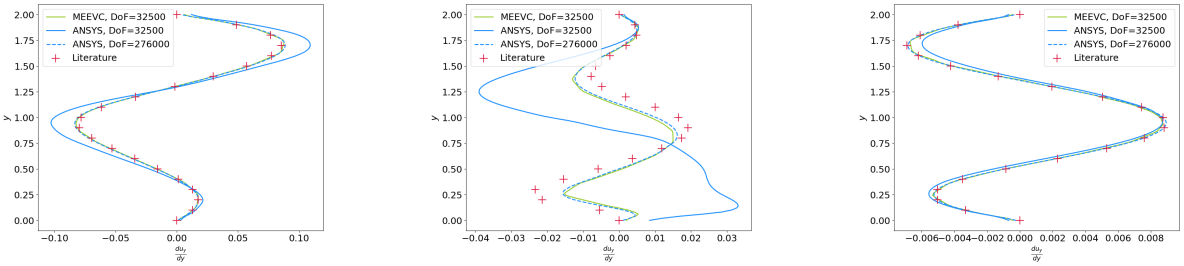
(a) $\frac{x}{h} = 6.$ **(b)** $\frac{x}{h} = 14.$ **(c)** $\frac{x}{h} = 30.$

Figure 4.30: Profiles for $\frac{du_x}{dy}$ for $Re = 800$, [60].



(a) $\frac{x}{h} = 6.$ **(b)** $\frac{x}{h} = 14.$ **(c)** $\frac{x}{h} = 30.$

Figure 4.31: Profiles for $\frac{du_y}{dx}$ for $Re = 800$, [60].



(a) $\frac{x}{h} = 6.$ **(b)** $\frac{x}{h} = 14.$ **(c)** $\frac{x}{h} = 30.$

Figure 4.32: Profiles for $\frac{du_y}{dy}$ for $Re = 800$, [60].

Lastly, the conservation laws are checked in the following manner. Both mass conservation $\nabla \cdot u_h = 0$ and vorticity $\mathcal{W} = \langle \omega \ , \ 1 \rangle = 0$. The results are shown in Figure 4.33. MEEVC sustains both errors at around 10^{-15} close to the floating point precision. For ANSYS the situation is a little different. The mass conservation error starts at a higher initial level but drops to 10^{-16} when the simulation progresses. Important to note that in order to reach such low levels of mass conservation error, the convergence criteria for ANSYS had to be adjusted from RMS of 10^{-6} to 10^{-14}. This change did not have any noticeable impact on the solution. ANSYS vorticity conservation reaches $\approx 3 \cdot 10^{-5}$ compared to MEEVC 10^{-15}. It might be caused by MEEVC formulation of the strong form which implies [17]

$$\mathcal{W} := \int_{\Omega} \omega \ d\Omega = \oint_{d\Omega} u \times n \ d\Gamma \tag{4.8}$$

with the help of a Stokes theorem. Therefore vorticity is conserved over time. In ANSYS general incompressible Navier-Stokes are computed not in rotational form which would impose Equation 4.8.

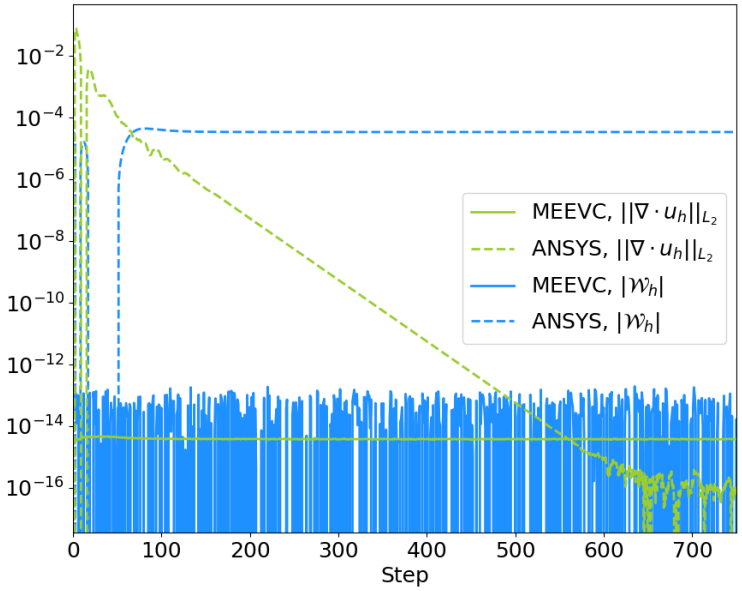


Figure 4.33: Discrete mass and vorticity conservation for backward-facing step at $Re = 800$.

4.4. Test case 4: Flow around the cylinder

The last test case for the comparison is a flow around a cylinder. The setup involves fluid flow past a cylindrical object, creating flow separation, vortex shedding and characteristic wakes. Shedding vortices start to occur at a critical Reynolds number Re_c. This value was found to be around $Re_c \approx 46$, [38, 63]. The study by Sengupta et al. [64] proposed that the critical Reynolds number Re_c may be influenced by the choice of numerical scheme and the experimental setup.

The Reynolds number for the flow around the cylinder is defined as

$$Re = \frac{D \cdot u_\infty}{\nu} \tag{4.9}$$

with $D = 1$ being the diameter of the cylinder, $u_\infty = 1$ is an average velocity at the inlet and ν is the dynamic viscosity of the fluid.

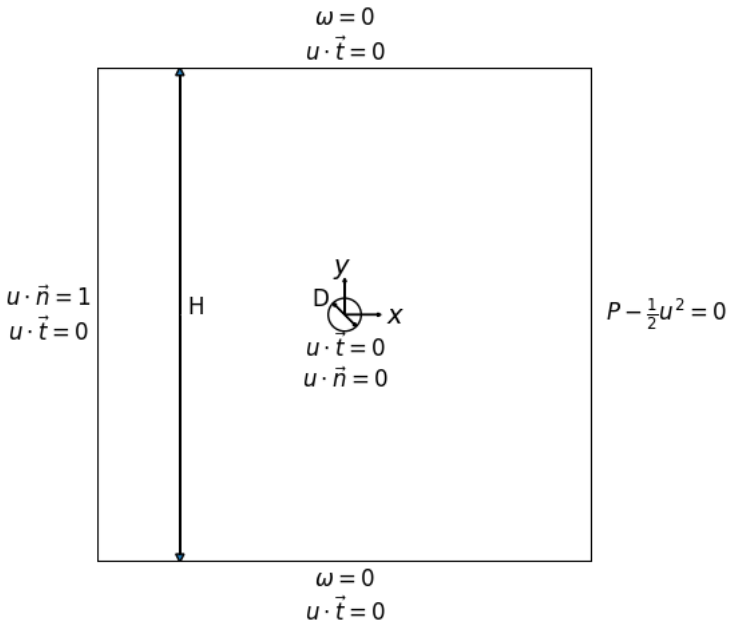


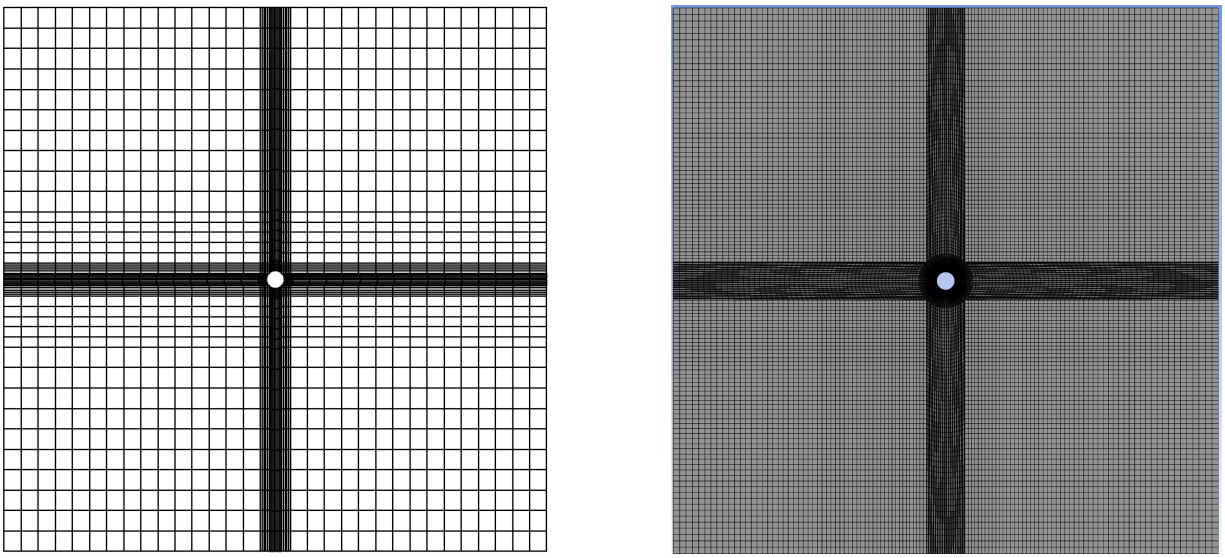
Figure 4.34: Domain Ω for a flow around the cylinder.

The Figure 4.34 shows a computational domain for flow around a circular cylinder of diameter $D = 1$, positioned at the center of a square domain with height $H = 30$. The relatively large domain is chosen to minimize the impact of the wall boundary on the cylinder, [65]. A study by Behr et al. [66] indicated that lateral (top and bottom) boundaries should be distant at least $8D$ from a circular cylinder. Otherwise, the Strouhal number St may be artificially overestimated. At the inlet, a uniform inflow is prescribed where the normal component of velocity is $u \cdot \vec{n} = 1$. At the outlet, a pressure boundary condition is applied $p = P - \frac{1}{2}u^2 = 0$. The walls are free-slip so no normal velocity ($u \cdot \vec{n} = 0$) and no vorticity ($\omega = 0$). On the cylinder surface, a no-slip condition is enforced, where both the normal and tangential velocity components are zero ($u \cdot \vec{n} = 0$ and $u \cdot \vec{t} = 0$). The coordinate system is positioned at the center of the cylinder, with the x-axis pointing to the right and the y-axis pointing upward. It is done that way so the values at cylinder surface are easily calculated ($r = \sqrt{x^2 + y^2} = 1$).

The convergence criteria for steady flow around the cylinder are as follows. Timestep of $\Delta t = \frac{1}{10}$ was chosen. The convergence criterion was set to be $\frac{\|u_h^k - u_h^{k-1}\|_{L^2(\Omega)}}{\Delta t} \leq 10^{-5}$ as for this

particular case no significant improvement was observed with previously used $\frac{\|u_h^k - u_h^{k-1}\|_{L^2(\Omega)}}{\Delta t} \leq 10^{-6}$. The order of the element is chosen to be $N = 3$. For ANSYS transient simulation for $Re = 200$, RMS residuals are kept below 10^{-6} for every timestep iteration. So for each timestep, the solution converges to a steady state before it progresses to the next timestep. Similarly to MEEVC, $\Delta t = \frac{1}{10}$ was selected for ANSYS transient simulation. The DoF$= 90000$ for both setups.

The mesh used for both ANSYS and MEEVC can be seen in Figure 4.35. The mesh is refined around the cylinder to accurately resolve the boundary layer. Furthermore, additional points are distributed along the surface of the cylinder, visible as a cross in the figure. This refinement is done in order to accurately estimate both the lift coefficient (c_l) and the drag coefficient (c_d), as precise pressure and vorticity data are required from the cylinder's wall.



(a) MEEVC. **(b)** ANSYS.

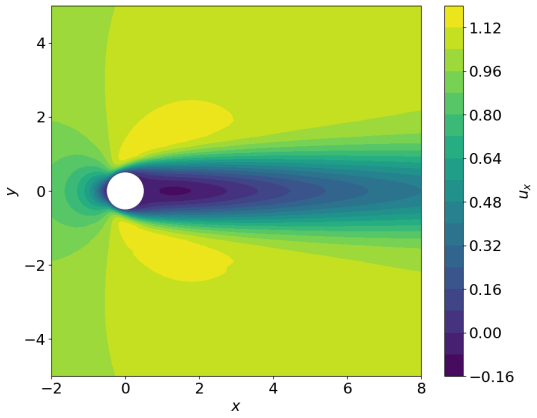
Figure 4.35: Mesh for MEEVC and ANSYS.

Numerical results for MEEVC and ANSYS at steady $Re = 40$ are shown in Figure 4.36 and Figure 4.37. Contour plots for horizontal velocity u_x, vertical velocity u_y, vorticity ω, stream function Ψ and static pressure p are illustrated. The full computational domain is not displayed, as it extends significantly. Instead, the region of interest surrounding the cylinder is presented to focus on the relevant flow characteristics.

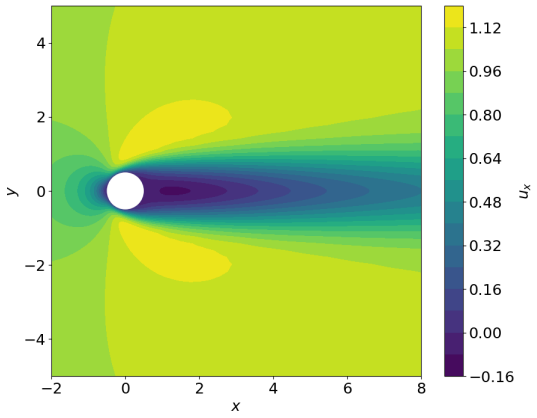
For comparison with literature data, lift and drag coefficients need to be computed. The drag force F_d and lift force F_l acting on a cylinder are computed using the surface integrals as follows

$$F_d = \frac{1}{2}\int_0^{2\pi} \left(-p\cos\theta - \nu\omega\sin\theta\right) D d\theta \tag{4.10}$$

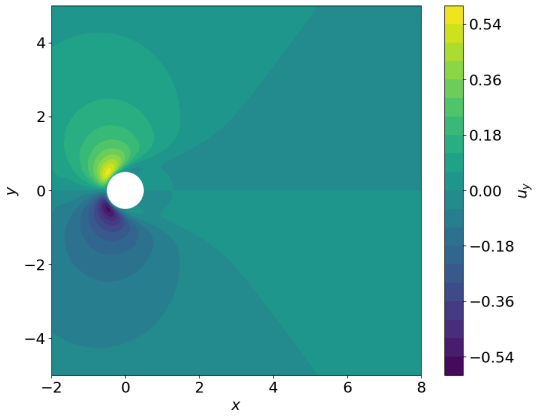
$$F_l = \frac{1}{2}\int_0^{2\pi} \left(-p\sin\theta + \nu\omega\cos\theta\right) D d\theta \tag{4.11}$$



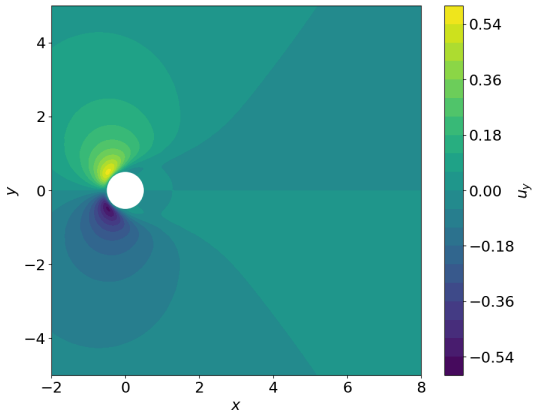
(a) MEEVC horizontal velocity u_x.



(b) ANSYS horizontal velocity u_x.



(c) MEEVC vertical velocity u_y.



(d) ANSYS vertical velocity u_y.

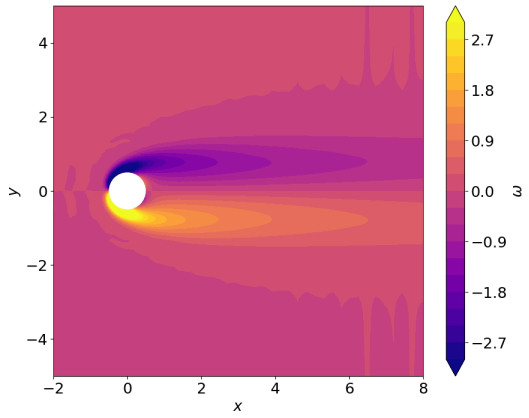
Figure 4.36: Velocity contour plots for $Re = 40$.

These integrals account for the contributions of pressure p and vorticity ω along the surface of the cylinder, parameterized by the angular position θ. The angular coordinate θ is measured starting from the most downstream point of the cylinder, located at (0.5, 0) and progresses in a counterclockwise direction. It can be seen in Figure 4.38. The factor ν, representing the kinematic viscosity and the cylinder diameter D, scales the viscous terms. Then the lift and drag force are normalized to obtain the coefficients

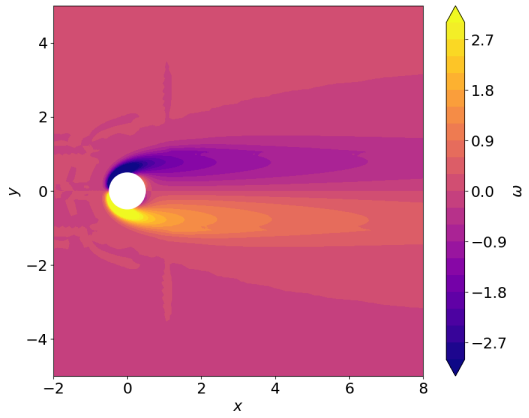
$$c_d = \frac{F_d}{\frac{1}{2}\rho u_\infty^2 D} \tag{4.12}$$

$$c_l = \frac{F_l}{\frac{1}{2}\rho u_\infty^2 D} \tag{4.13}$$

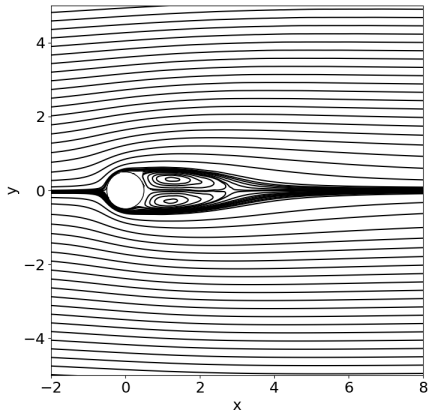
with fluid density $\rho = 1$, inlet velocity $u_\infty = 1$ and cylinder diameter $D = 1$.



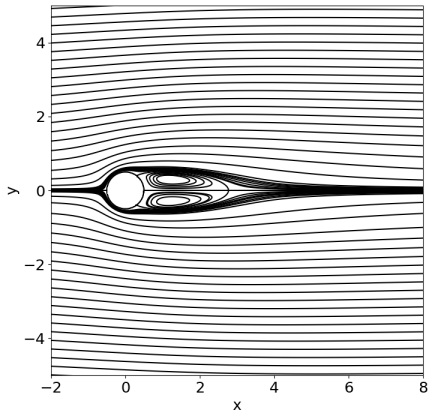
(a) MEEVC vorticity ω.



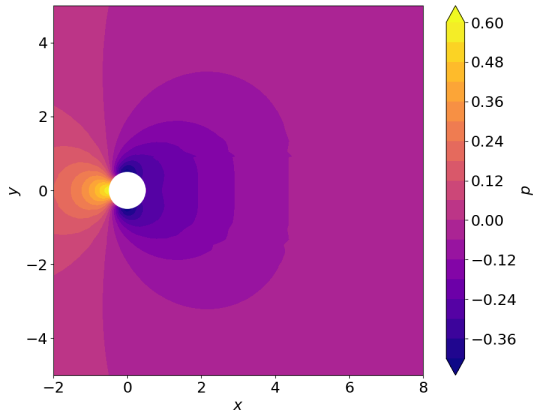
(b) ANSYS vorticity ω.



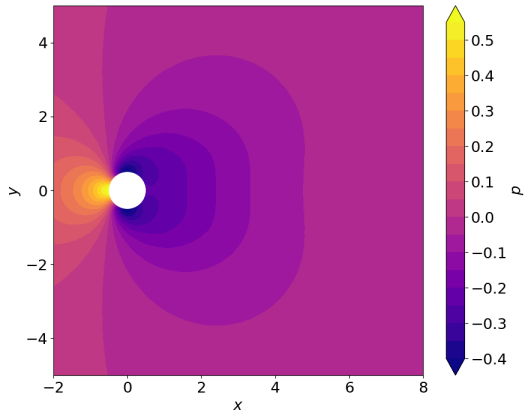
(c) MEEVC stream function Ψ.



(d) ANSYS stream function Ψ.



(e) MEEVC static pressure p.



(f) ANSYS static pressure p.

Figure 4.37: Vorticity ω, stream function Ψ and static pressure p contour plots for $Re = 40$.

In Figure 4.38 the characteristic dimensions of the wake structure behind a cylinder are illustrated. These dimensions are essential for comparison with literature data. The parameter a represents the distance of the cylinder to the center of the wake vortex. The width of the wake vor-

tices is denoted as b. It is a critical measure for determining how far the flow disturbance spreads behind the cylinder. l is the wake length which is the distance from the rear of the cylinder to the point where the flow fully reattaches or becomes more streamlined. Finally, θ denotes the separation angle, marking the point on the cylinder surface where the boundary layer separates and the wake begins to form.

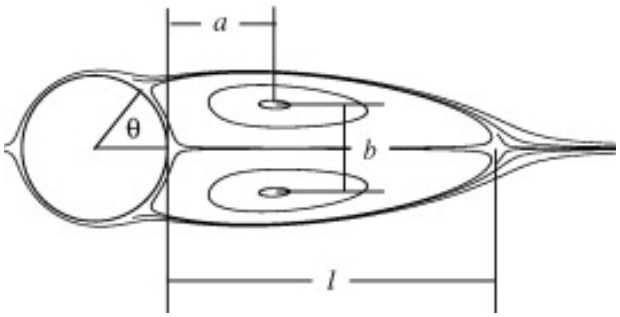


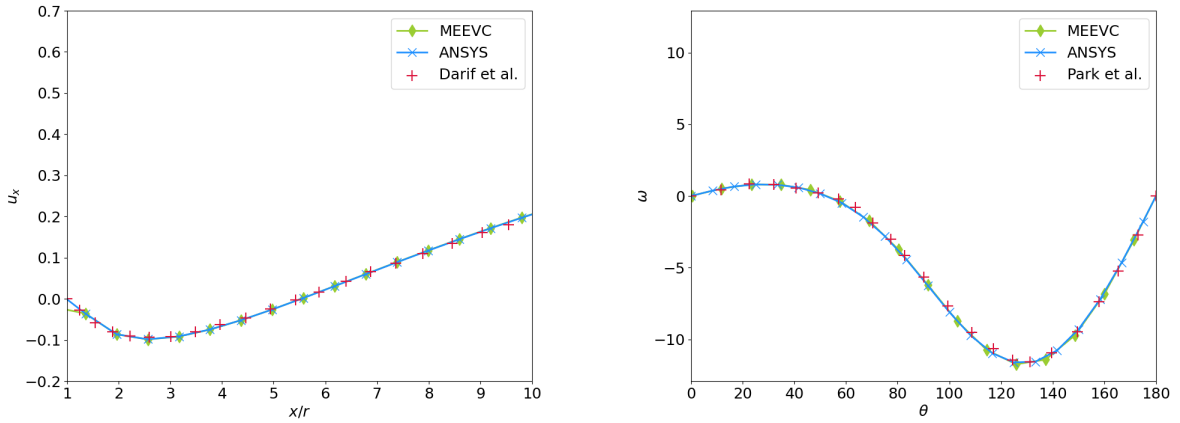
Figure 4.38: Definition of the characteristic dimensions of the wake structure, [65].

The flow characteristic lengths are shown in Table 4.8. A few source data were used for comparison: Taira's Finite Volume method with immersed boundary and 300^2 grid [65], well-renowned experimental data by Coutanceau and Bouard [67], least-square spectral element method with $N = 7$ by Groot [68], Park's numerical data with 641×241 grid [69], Finite Difference method on 641×321 grid by Linnick and Fasel [70] and very recent new solution method for infinite domains on 160^2 grid by Darif [71]. The values are normalized by a cylinder diameter D. Overall, both methods agree with the literature data. MEEVC accurately predicts all values except for the separation angle θ_{sep}. For MEEVC separation angle tends to be slightly underestimated.

Table 4.8: Comparison of flow characteristics for $Re = 40$. [65, 67, 68, 72, 70, 71]

	l/D	a/D	b/D	θ_{sep}	c_d
Taira [65]	2.30	0.73	0.60	53.7°	1.54
Coutanceau and Bouard [67]	2.13	0.76	0.59	53.8°	–
Groot [68]	2.17	0.68	0.58	53.77°	1.61
Park et al. [69]	2.25	–	–	52.2°	1.51
Linnick and Fasel [70]	2.28	0.72	0.60	53.6°	1.54
Darif [71]	2.24	–	–	53.6°	1.50
MEEVC	2.264	0.735	0.593	52.22°	1.552
ANSYS	2.262	0.715	0.594	52.57°	1.563

In Figure 4.39 two profiles are presented, horizontal velocity u_x and vorticity ω. u_x is accessed from the centerline ($y = 0$) and compared to study by Darif in Figure 4.39a, [71]. The horizontal velocity starts at 0 as it is the surface of the cylinder, followed by the wake region with negative values. The Figure 4.39b presents vorticity ω vs angular coordinate θ at the upper side of the cylinder with the numerical study by Park et al., [69]. Both methods follow the reference values very similarly.



(a) Horizontal velocity u_x at $y = 0$, [71].

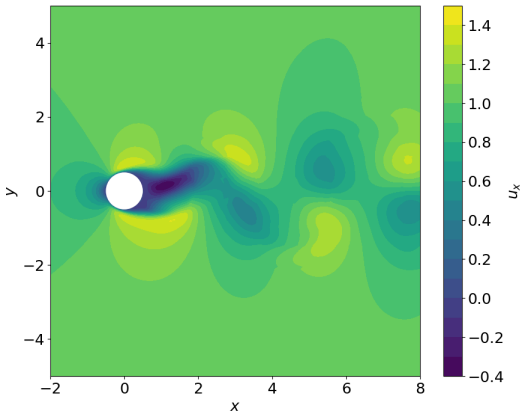
(b) Vorticity at the cylinder, [69].

Figure 4.39: Velocity and vorticity profile for $Re = 40$, [71, 69].

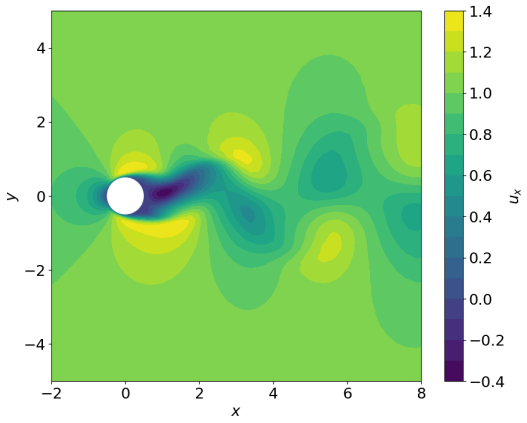
There are several ways to trigger the onset of flow asymmetry: impulsive start and perturbation of initial condition [73], perturbation by cylinder (rotation, oscillation or surface roughness) [74] and perturbation of inlet boundary condition [75]. In this study, the method of perturbed initial conditions proposed by Mouna Laroussi [76] is used. The method is designed to ensure that no additional energy is introduced into the flow. It follows the initial conditions for the flow

$$u_{initial}(y) = u_\infty \cdot \left(1 + \alpha \cdot sin\left(\frac{2\pi y}{H}\right)\right) \tag{4.14}$$

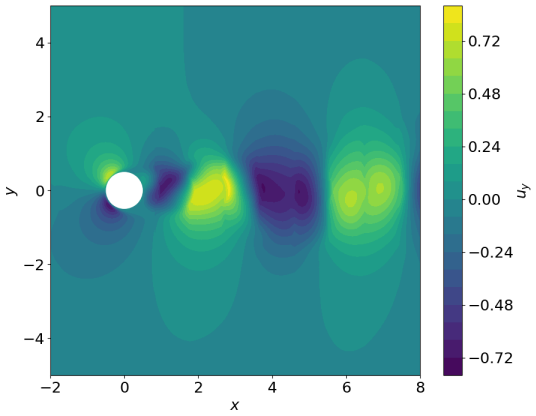
where $u_\infty = 1$ is an inlet velocity, α is a coefficient of the perturbed velocity and $H = 30$ is domain height. The optimal coefficient for $Re = 200$ follows an expression $\alpha_{opt} = -\frac{6.25}{Re} + 0.34$, [76] which gives $\alpha \approx 0.3$. The simulation is carried out to $t_{max} = 80$. The vorticity at the outlet is set to zero as otherwise, reflections are created. The same timestep as for steady case is used so $\Delta t = \frac{1}{10}$. The simulation diverges and thus, no results are generated. In order to prevent reflections of waves at the outflow boundary zero vorticity is set in the case of flow past a bluff body, [77].



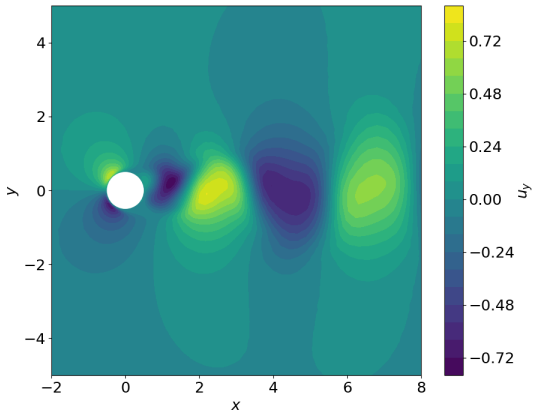
(a) MEEVC horizontal velocity u_x.



(b) ANSYS horizontal velocity u_x.



(c) MEEVC vertical velocity u_y.



(d) ANSYS vertical velocity u_y.

Figure 4.40: Velocity contour plots for $Re = 200$ at $t = 80$.

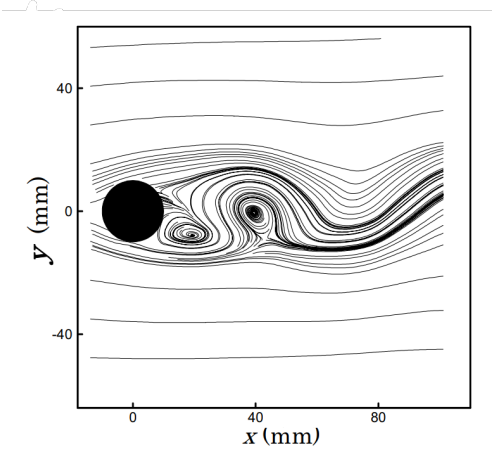
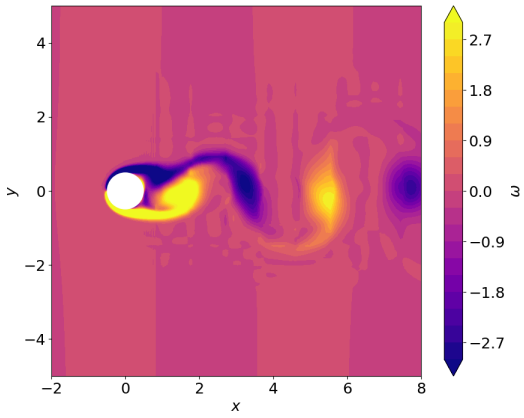
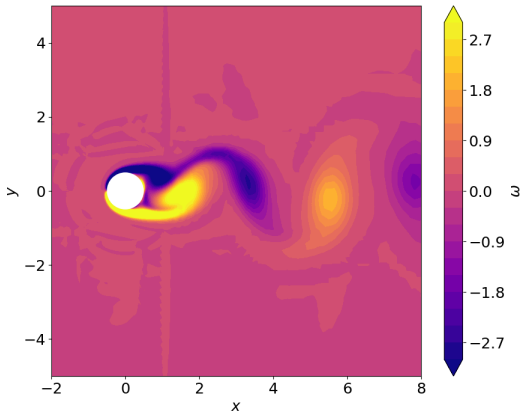


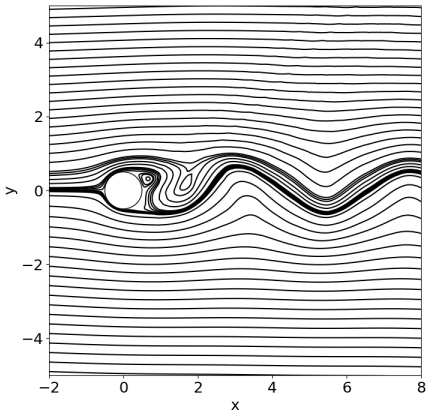
Figure 4.41: Instantaneous in-plane flow field for stream function Ψ at $Re = 200$, [78].



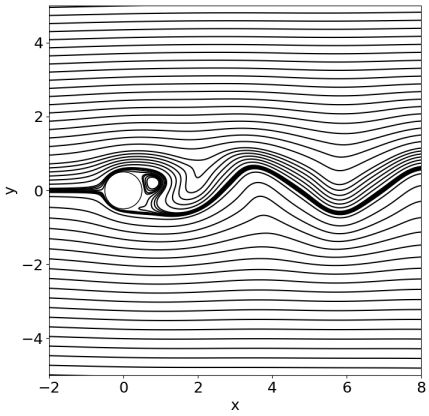
(a) MEEVC vorticity ω.



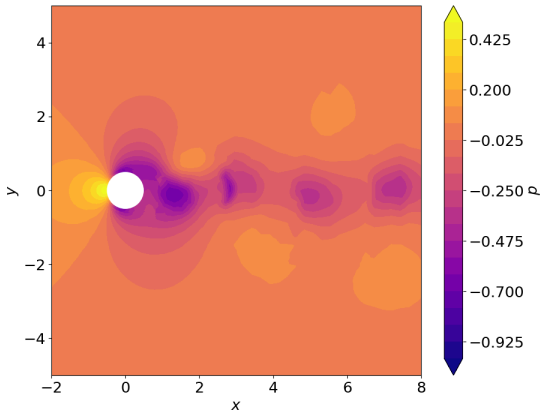
(b) ANSYS vorticity ω.



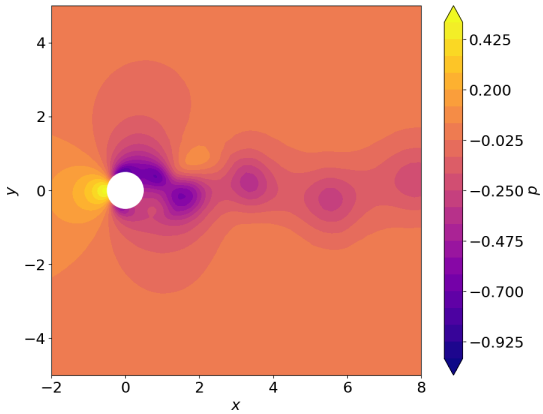
(c) MEEVC stream function Ψ.



(d) ANSYS stream function Ψ.



(e) MEEVC static pressure p.



(f) ANSYS static pressure p.

Figure 4.42: Vorticity ω, stream function Ψ and static pressure p contour plots for $Re = 200$ at $t = 80$.

The results for velocity MEEVC and ANSYS at $Re = 200$ are shown in Figure 4.40. Contour plots for horizontal velocity u_x and vertical velocity u_y are illustrated. Figure 4.42 shows coutour

plots for vorticity ω and stream function Ψ. Similarly to $Re = 40$ case, region near the cylinder is shown. For both methods, strong alternating vorticity patterns downstream indicate the formation of vortices. High vorticity is concentrated near the cylinder's surface and in the shed vortices. For validation, instantaneous flow field of stream function Ψ for $Re = 200$ can be seen at Figure 4.41, [78]. The experimental results are derived from Digital Particle Imaging Velocimetry (DPIV). Both ANSYS and MEEVC recreate streamlines similarly to Wang et al., [78].

For the unsteady simulation of flow around the cylinder at $Re = 200$, different convergence conditions had to be introduced. As flow does not reach a steady-state, velocity residuals $\frac{\|u_h^k - u_h^{k-1}\|_{L^2(\Omega)}}{\Delta t}$ cannot be used as a convergence criterion. The Kármán vortex street is formed from the periodic shedding of vortices. Both c_l and c_d fluctuate over time. The lift coefficient c_l should fluctuate around 0. Therefore two convergence conditions are introduced

$$\frac{1}{2}\left(c_{l,max} - |c_{l,min}|\right) \leq 10^{-3} \tag{4.15}$$

$$\int_{\tau_1}^{\tau_2} c_l(\tau) \cdot d\tau \leq 10^{-3} \tag{4.16}$$

with $c_{l,max}$ and $c_{l,min}$ being maximal and minimal value at a given shedding period, $(\tau_2 - \tau_1)$ represents the shedding period. These criteria ensure that the periodic vortex shedding becomes consistent. The period is calculated from c_l data for each timestep.

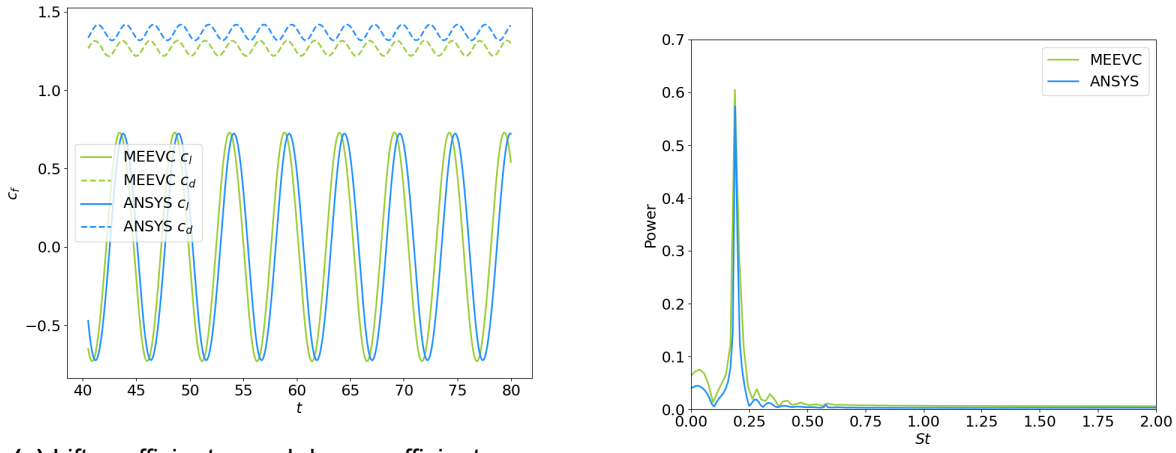
The findings of the aforementioned analysis can be found in Table 4.9. Condition 1 refers to Equation 4.15 whereas condition 2 refers to Equation 4.16. Those represent the first timestep t_c where both conditions are met. It is 38.8 for MEEVC and 40.5 for ANSYS so both methods converge at similar timesteps.

Table 4.9: Stationary shedding conditions for $Re = 200$.

	Convergence time t_c	Condition 1 ($\times 10^{-3}$)	Condition 2 ($\times 10^{-3}$)
MEEVC	38.8	0.506	0.682
ANSYS	40.5	0.021	0.842

The Figure 4.43 presents force coefficient c_l, c_d and Strouhal number St for a flow around the cylinder at Reynolds number $Re = 200$. The variation of the lift coefficient over time is depicted for both MEEVC and ANSYS. The comparison shows a close agreement between the two methods, with periodic oscillations in the lift coefficient over time, indicative of vortex shedding. The amplitude and frequency of the lift coefficient oscillations are well-captured by both solvers, suggesting accurate modelling of the unsteady flow behaviour. The power spectrum of the lift coefficient is done in order to determine the dominant frequency peak corresponding to the shedding frequency f. This analysis is achieved through the application of a Fourier transform to the time-dependent lift coefficient. Both solvers exhibit a similar power spectrum, reinforcing the consistency of the results across the two methods. From this frequency, the Strouhal number St is calculated using

$$St = f \cdot \frac{D}{u_\infty}. \tag{4.17}$$



(a) Lift coefficient c_l and drag coefficient c_d over time t.

(b) Power spectrum of lift coefficient c_l.

Figure 4.43: Lift, drag coefficients over time and Strouhal number estimation for $Re = 200$.

In Figure 4.44, two time discretization schemes are shown: 1^{st} and 2^{nd} order Backward Euler scheme. As already discussed in Chapter 3, lower-order schemes tend to suppress the higher-wave-number components that are responsible for initiating asymmetry in flow around the cylinder, [50]. The 1^{st} order scheme damps both drag and lift coefficients and produces higher errors w.r.t. literature values.

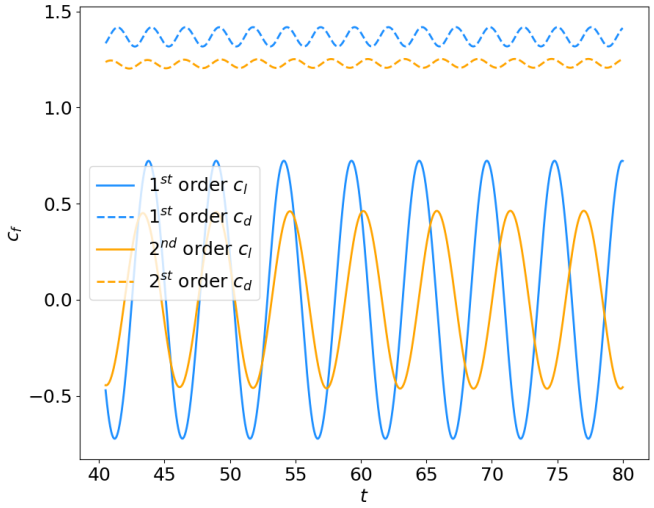


Figure 4.44: Time discretization schemes comparison for ANSYS.

A comparison of flow properties at $Re = 200$, including the Strouhal number St, drag coefficient c_d and lift coefficient c_l is presented in Table 4.10. Results from various studies are shown alongside the computational results from MEEVC and ANSYS. A few data sources were used for comparison: Taira's Finite Volume method with immersed boundary and 300^2 grid [65], Finite Difference method on 641×321 grid by Linnick and Fasel [70], hybrid method (FEM/FVM) by Laroussi with 55000 elements [76], Belov's implicit algorithm with multigrid [79], pseudo-

compressibility method by Liu et al. on 256^2 grid [80] and Smoothed Particle Hydrodynamics (SPH) by Marrone and Colagrossi on stagerred grid [81]. The Strouhal number which represents the frequency of vortex shedding, is generally consistent across the studies, with small variations. Both methods, ANSYS and MEEVC, share the same Strouhal number $St = 0.189$ which is on the lower end of reference values. The lift coefficient is also similar between the two at around $c_l = 0.72$ which tends to be a bit overestimated. The biggest difference is in mean drag coefficient which is off by around $\varepsilon_{c_d} = 5.3\%$ for MEEVC but only $\varepsilon_{c_d} = 2.1\%$ for ANSYS across the literature values. On the other hand, the fluctuations in the drag coefficient are better represented with MEEVC (only 0.4% error) whereas it equals 6% for ANSYS.

Table 4.10: Comparison of computed flow properties $Re = 200$. [65, 70, 76, 79, 80, 81]

	St	c_d	c_l
Taira [65]	0.196	1.35 ± 0.048	± 0.68
Linnick and Fasel [70]	0.197	1.34 ± 0.044	± 0.69
Laroussi [76]	0.199	1.47 ± 0.05	± 0.77
Belov et al. [79]	0.193	1.19 ± 0.042	± 0.64
Liu et al. [80]	0.192	1.31 ± 0.049	± 0.69
Marrone and Colagrossi [81]	0.2	1.38 ± 0.05	± 0.68
MEEVC	0.189	1.268 ± 0.047	± 0.729
ANSYS	0.189	1.368 ± 0.05	± 0.722

The reason for overestimated drag might be the lateral boundaries used in the studies. Linnick and Fasel [70] and Laroussi [76] use $5D$ of lateral distance between walls and cylinder, for Marrone and Colagrossi [81] it is equal to even lower $2.5D$. As already mentioned this might lead to artificially higher shedding frequency and as a result higher St. All of these methods estimate Strouhal number close to 0.2 compared to 0.189 of ANSYS and MEEVC. When compared to the rest of the studies, the error reduces from 3.7% to 2.4%.

4.5. Cross-case study

This section presents a cross-case study comparing the performance of the MEEVC and ANSYS Fluent of the test cases presented in this chapter. The tests examined are the Taylor-Green vortex, lid-driven cavity flow, backward-facing step and flow around a cylinder. The comparison focuses on accuracy, agreement with literature/experimental data, flow type influence and computational efficiency.

4.5.1. Accuracy

Accuracy is the difference between a computed value and the actual value. In order to check the accuracy, a Taylor-Green Vortex test case was carried out. An analytical solution to the Navier-Stokes equations represents the true solution to the problem. Literature data may contain errors or uncertainties which may alter the accuracy measurement. It is still important to compare with literature as it reflects real-world complexities and imperfections. It showcases where certain methods excel and where it lacks credibility.

During the Taylor-Green Vortex (TGV) test case, two types of errors were assessed to evaluate the accuracy of both solvers.

Firstly, the L^2-error for velocity, vorticity and static pressure was computed at $t = 1$ in order to analyze the $p - h$ convergence behaviour. The results showed that, as the mesh refinement increased, the L^2-error for velocity, vorticity and pressure consistently decreased, indicating convergence. MEEVC L^2-error was consistent with ANSYS at the same degrees of freedom (DoF) for velocity and pressure. However, the highest difference was found in vorticity ω where MEEVC performed much better for the same DoF (about an order of magnitude lower L^2-error). The MEEVC showed improved convergence, particularly with higher polynomial degrees ($N = 3$, $N = 4$). ANSYS convergence was comparable with that of MEEVC with $N = 2$. Slightly higher for velocity u and lower for vorticity ω.

Subsequently, the relative error in the maximum velocity was measured at $t = 10$, comparing the numerical results to the analytical solution. MEEVC's relative error decreased significantly as the polynomial degree N increased. With $N = 1$, the relative error for velocity was approximately 1.3379% but as the polynomial degree was increased to $N = 3$, the error was reduced to as low as 0.0007%. For ANSYS, it performed better than MEEVC at low DoF but produced larger errors at moderate and high DoF. For the highest DoF$= 33000$, velocity and pressure errors were 2 orders of magnitude smaller for MEEVC. The vorticity error was 5% for ANSYS and 0.0009% for MEEVC which is a significant difference.

Additionally, similar test at $t = 20$ was done to check the conserving properties of both methods. The errors for MEEVC did not change at all. On the other hand, ANSYS errors increased for all flow properties. It shows that MEEVC accuracy becomes more prominent as simulation time progresses due to its conserving properties. For backward-facing step at $Re = 800$, mass and vorticity conservation was evaluated. While ANSYS mass conservation reached levels of MEEVC after a few hundred iterations, it required setting RMS levels to ridiculously low levels of 10^{-12}. At RMS of 10^{-6} which was used throughout the test cases, the conservation of mass lowered to $\approx 10^{-8}$. The vorticity conservation fluctuated around 10^{-15} for MEEVC and 10^{-5} for ANSYS. So MEEVC satisfies conservation laws with a higher degree of accuracy.

4.5.2. Comparison with literature/experimental data

Table 4.11 presents a summary of errors for both MEEVC and ANSYS methods across different test cases where literature data was presented: lid cavity, backward-facing step and flow around the cylinder. The errors have been averaged across different Reynolds numbers. For c_d and c_l of the cylinder, only mean value was taken for comparison.

For the lid cavity, the error in the stream function (Ψ) is low for both methods, with MEEVC showing 0.1957% and ANSYS slightly higher at 0.4591%. The vorticity (ω) error is higher for both methods, but MEEVC performs better, showing 0.8482% compared to ANSYS's 1.4724%. The error in the coordinates (x, y) is comparable for both methods, with MEEVC having higher 0.4000% and ANSYS 0.1528%.

In the backward-facing step case, MEEVC demonstrates significantly better accuracy in predicting the first recirculation length (X_1), with an error of 0.5337% compared to ANSYS's 3.6967%. MEEVC also outperforms ANSYS in predicting the separation of second recirculation length (X_2), where the errors are 0.6566% and 6.5876%, respectively. The error in the end position second recirculation zone (X_3) is much lower for MEEVC (0.2477%) compared to ANSYS (1.5589%). Finally, for the length of the second recirculation zone between $X_3 - X_2$, MEEVC is accurate at 0.0987% while ANSYS is off by 2.6998%. In order to match the performance of MEEVC, ANSYS required about 8.5 more DoF.

Table 4.11: Summary of error w.r.t. literature data along test cases.

	ε_{MEEVC} [%]	ε_{ANSYS} [%]
Lid cavity		
Ψ	0.1957	0.4591
ω	0.8482	1.4724
(x, y)	0.4000	0.1528
Backward-facing step		
X_1	0.5337	3.6967
X_2	0.6566	6.5876
X_3	0.2477	1.5589
$X_3 - X_2$	0.0987	2.6998
Flow around the cylidner		
l/D	1.6006	1.5108
θ_{sep}	2.2921	1.6372
c_d	3.0762	1.7916
c_l	5.3977	4.3855
St	2.4016	2.3976

For the flow around the cylinder case, ANSYS performs better across all parameters. The length-to-diameter ratio (l/D) and the Strouhal number (St) are quite close between the two methods, with errors of 1.5% and 2.4%, respectively. Strouhal number error excluded a few sources as the walls right interfered with the results. The biggest difference is in drag coefficient (c_d) which is predicted better with ANSYS with 1.7916% error compared to 3.0762% for MEEVC. Separation angle θ_{sep} and lift coefficient are predicted similarly but with a slight advantage for ANSYS.

The values for BR2 so tertiary vortex for lid cavity are excluded from the average. As values are very sensitive to relatively small fluctuations in measured quantity (also measurement error),

they can produce large relative errors. For BR2, the relative error in stream function equals 6% for MEEVC and 20% for ANSYS. Nevertheless, these values suggest that the MEEVC tends to be more accurate in capturing tertiary vortices compared to ANSYS.

4.5.3. Flow type influence

Table 4.12 summarizes the percentage error with respect to literature data for various Reynolds numbers in each test case.

Table 4.12: Summary of error w.r.t. literature data along test cases for various Reynolds numbers.

	ε_{MEEVC} [%]		ε_{ANSYS} [%]	
Lid cavity				
	$Re = 400$	$Re = 1000$	$Re = 400$	$Re = 1000$
Ψ	0.2008	0.1906	0.5033	0.4148
ω	1.1005	0.5960	2.1424	0.8023
(x, y)	0.4696	0.3296	0.1572	0.1482
Backward-facing step				
	$Re = 100$	$Re = 800$	$Re = 100$	$Re = 800$
X_1	0.2396	0.8278	0.8214	6.5721
X_2	–	0.6566	–	6.5876
X_3	–	0.2477	–	1.5589
$X_3 - X_2$	–	0.0987	–	2.6998
Flow around the cylidner				
	$Re = 40$	$Re = 200$	$Re = 40$	$Re = 200$
l/D	1.6006	–	1.5108	–
θ_{sep}	2.2921	–	1.6372	–
c_d	0.7792	5.3731	1.4935	2.0896
c_l	–	5.3977	–	4.3855
St	–	2.4016	–	2.3976

The performance of MEEVC and ANSYS varies across different flow types (internal, external, steady and unsteady). For internal flows like the lid-driven cavity, MEEVC demonstrates good accuracy for the same DoF, especially at higher Reynolds numbers (Re). At $Re = 1000$, MEEVC shows significantly reduced errors for the vorticity (0.5960%), better than ANSYS (0.8023%). For more complex internal flows, such as the backward-facing step, MEEVC performs notably better, especially at higher Reynolds numbers. At $Re = 800$, MEEVC exhibits lower errors in the reattachment lengths for both $Re = 100$ and $Re = 800$ test cases. This indicates that MEEVC is particularly effective in capturing flow separation and reattachment phenomena in the internal flows.

For external steady flows, such as flow around a cylinder at $Re = 40$, MEEVC accurately captures wake characteristics. Wake length l/D and drag coefficient error is below 1% while separation angle is underestimated with 2% error. On the other hand, ANSYS captures the separation angle more precisely but overestimates the drag coefficient with 1.5%. Therefore, for external steady, both methods perform similarly but excel in capturing different flow properties.

For external unsteady flows, such as flow around a cylinder at $Re = 200$, MEEVC struggles to accurately capture vortex shedding. The lift coefficient ($\varepsilon_{c_l} = 5.4\%$) and drag coefficient ($\varepsilon_{c_d} = 5.4\%$) errors are quite off from the literature data. In contrast, ANSYS produces lower errors for drag coefficients ($\approx 2\%$) but similar for lift coefficients (4.4%). Also, Strouhal number St is approximated very closely to MEEVC. Therefore, even though ANSYS can predict c_d better, other parameters are estimated with similar errors.

4.5.4. Computational Efficiency

When focused on computational time, MEEVC was significantly less efficient in this aspect than ANSYS. Table 4.13 show computational time for both methods. Generally, ANSYS requires less time to perform a simulation, especially for the lid cavity where a steady-state solver was used. For transient simulations where timestep was prescribed, several iterations were required for each timestep in order to achieve the residual value which increased the computational time but not as much. It was still lower than that for MEEVC. MEEVC on average requires from 1 to 30 seconds per iteration depending on the flow regime and the polynomial degree N, thus it is not very practical for unsteady and complex flow cases. In contrast, ANSYS completed the timesteps in a fraction of a second. This stark difference in computational performance highlights ANSYS's advantage in terms of time-to-solution, especially for large-scale simulations or flows requiring a large number of timesteps. The main drawback of this implementation is that Python's relatively slower performance makes it unsuitable for handling large-scale problems, [82]. The ANSYS supports parallel processing as discussed in Section 3.1. It also reduces its computational time. ANSYS also scales better with increasing DoF and Re, making it more suitable for scenarios that demand finer computational grids.

Table 4.13: Computational Data for Various Simulations

Method	Re	DoF	Solver	Iterations	Time [min]	Time per iteration [s]
				TGV		
ANSYS	1600	3700	Coupled	7847	8	0.060
ANSYS	1600	3700	SIMPLE	8073	8	0.059
ANSYS	1600	33000	Coupled	15530	15	0.058
MEEVC	1600	3700	–	2000	56	1.7
MEEVC	1600	33000	–	2000	408	12.2
				Lid cavity		
ANSYS	400	58000	Coupled	127	1	0.5
ANSYS	1000	58000	Coupled	292	2	0.2
MEEVC	400	58000	–	6200	846	8.18
MEEVC	1000	58000	–	16000	2118	7.94
				Flow around the cylinder		
ANSYS	200	90000	Coupled	17500	22	0.08
MEEVC	200	90000	–	2000	857	25.71

However, hardware and software constraints have been reducing for the last few decades, [3]. In cloud computing, CPU performance is levelling but GPU provides a decrease in power consumption and hardware costs. Modern GPUs have millions of cores instead of 1000s. A single GPU can offer the same performance as more than 400 CPUs, [83]. So, the time-to-solution might not be as big of a problem in the future as it currently is.

5

Conclusion & Recommendation

Conclusion

In this study, the Mimetic Spectral Element Method using mass, energy, enstrophy and vorticity conserving (MEEVC) solver and conventional Computational Fluid Dynamics (CFD) approaches, portrayed by ANSYS Fluent, were compared across a variety of benchmark flow problems. A detailed description of the MEEVC scheme is provided in Chapter 2 where the open-source module for Python called phyem is used to solve 2D incompressible Navier-Stokes equations using a mimetic scheme. The conventional CFD methodology of ANSYS Fluent is outlined in Chapter 3. The results of the test cases as well as a cross-case study are presented in Chapter 4.

Four test cases were chosen for the comparison. The tests included the Taylor-Green vortex, lid-driven cavity flow, backward-facing step and flow around a cylinder. They were chosen in order to answer three research questions:

- *How does the accuracy of the Mimetic Spectral Element Method (MSEM) compare to conventional Computational Fluid Dynamics (CFD) techniques?*

  The results of the Taylor-Green Vortex test case demonstrate that MEEVC consistently outperforms ANSYS Fluent in terms of accuracy, particularly as mesh resolution and polynomial degrees increase. The MEEVC has a lower L_2-error for velocity, vorticity and static pressure compared to ANSYS, achieving it with the same number of degrees of freedom (DoF). As the polynomial degree increased, the relative error for MEEVC significantly decreased, with higher polynomial degrees, closely approximating the analytical solution. The differences were mostly seen for DoF$= 33000$ where velocity and pressure L^2-errors were 2 orders of magnitude higher for ANSYS. For the vorticity, it was 3 orders of magnitude. Additionally, for backward-facing step mass and vorticity conservation laws were evaluated. MEEVC satisfies conservation laws with a higher degree of accuracy compared to ANSYS.
- *How are both methods compared to literature/experimental data?*

  In comparison with the literature data, both methods exhibited some strengths across different test cases. MEEVC demonstrated better accuracy in the backward-facing step problem as it predicted recirculation lengths closer to literature values. Moreover, the section plots for velocity and vorticity were in closer agreement with the literature for MEEVC. However, ANSYS performed better in the flow around the cylinder case, especially in predicting drag, lift and separation angles. Both methods show similar performance in the lid cavity test case, with only minor differences in error percentages between them. The MEEVC tended to be more accurate in capturing tertiary vortices than ANSYS. Generally, both methods show good agreement with the literature/experimental data.
- *How do the Mimetic Spectral Element Method and conventional CFD methods perform across different flow types (e.g. internal, external, steady, unsteady)?*

MEEVC has proven to be highly accurate in capturing the details of internal flow cases such as backward-facing step, especially at higher Reynolds numbers. It produced lower errors in predicting vortex recirculation lengths than ANSYS, underscoring its superior performance in flows involving separation and reattachment phenomena (internal flows). In external steady flow cases, such as flow around a cylinder at $Re = 40$, MEEVC and ANSYS provided a satisfying approximation of wake characteristics, with errors in drag coefficient and wake length below 2%, though MEEVC slightly underestimates separation angles. However, in the unsteady flow cases, like the vortex shedding in flow around the cylinder at $Re = 200$, ANSYS outperformed MEEVC in capturing time-dependent phenomena. It showed lower errors in the drag coefficient. While lift coefficient and Stouhal number were approximated similarly for both methods.

The computational efficiency of MEEVC has a notable disadvantage. It requires longer computation times than ANSYS. In contrast, ANSYS uses parallel processing, making it far more efficient with increasing DoF and Re, making it more suitable for scenarios that demand finer computational grids. On the other hand, the GPUs are becoming more popular in computing environments. They can offer the same performance as more than 400s CPUs which might reduce computational constraints. MEEVC turned out to perform better in terms of accuracy and conservation properties. But still lacks in terms of computational efficiency compared to ANSYS. And for the current moment, the accuracy does not outweigh the computational time. ANSYS is preferable in the early stages of the design process. However, MEEVC might find its usage for simulations involving complex separation phenomena where accuracy to capture it accurately is crucial.

Recommendation

Based on the findings of this study, the results suggest that both methods have their own advantages and drawbacks. The following recommendations are proposed for future work and practical applications.

1. Investigation of the outflow instabilities. While the backward-facing step was accurately simulated by MEEVC, the case for $N = 3$ did not reach steady-state as discussed in Section 4.3. Several adjustments were tested but with no success. Similarly, for flow around the cylinder at $Re = 200$, the solution errors increased for the unsteady case. These two instances should be thoroughly analyzed to assess the limitations of the scheme. The original MEEVC scheme had some difficulties with handling no-slip conditions without sacrificing the property of vorticity conservation.
2. Further optimization of MEEVC. It consistently demonstrated great accuracy across the test cases. Unfortunately, it is computationally more demanding, particularly at higher polynomial degrees. Future efforts should focus on optimizing the MEEVC solver to reduce computation times. All the non-linear system solving is done within Python using the Netwon iterative process. It does not linearize the discrete systems that were used in the original MEEVC scheme. Potentially integrating parallel processing techniques might reduce solving time as the domain is divided into segments computed individually by one logical processor.
3. Exploration of other test cases. The current study demonstrated MEEVC's effectiveness in moderate Reynolds number flows in 2D. Future studies can explore higher Reynolds number regimes as well as a 3D flow. The phyem package shows capabilities in handling three-dimensional cases. Additionally, different boundary conditions might be tested e.g. oscillating BC for lid cavity flow.

References

[1] P. R. Spalart et al. “On the role and challenges of CFD in the aerospace industry”. en. In: *The Aeronautical Journal* 120.1223 (Jan. 2016). Publisher: Cambridge University Press, pp. 209–232. DOI: 10.1017/aer.2015.10. URL: https://www.cambridge.org/core/journals/aeronautical-journal/article/on-the-role-and-challenges-of-cfd-in-the-aerospace-industry/AB70FEF00301B20648F5B0627893B787 (visited on 06/28/2023).

[2] Krotil Stefan et al. “CFD-Simulations in the Early Product Development”. In: *Procedia CIRP*. 13th Global Conference on Sustainable Manufacturing – Decoupling Growth from Resource Use 40 (Jan. 2016), pp. 443–448. DOI: 10.1016/j.procir.2016.01.090. URL: https://www.sciencedirect.com/science/article/pii/S2212827116001050 (visited on 11/02/2024).

[3] David F. Fletcher. “The future of computational fluid dynamics (CFD) simulation in the chemical process industries”. In: *Chemical Engineering Research and Design* 187 (Nov. 2022), pp. 299–305. DOI: 10.1016/j.cherd.2022.09.021. URL: https://www.sciencedirect.com/science/article/pii/S0263876222005202 (visited on 11/02/2024).

[4] *Computational fluid dynamics—retrospective and prospective: International Journal of Computational Fluid Dynamics: Vol 19, No 8*. URL: https://www-tandfonline-com.tudelft.idm.oclc.org/doi/abs/10.1080/10618560600585315 (visited on 08/07/2023).

[5] Qian-Min Huang et al. “High-order compact finite volume schemes for solving the Reynolds averaged Navier-Stokes equations on the unstructured mixed grids with a large aspect ratio”. en. In: *Journal of Computational Physics* 467 (Oct. 2022), p. 111458. DOI: 10.1016/j.jcp.2022.111458. URL: https://www.sciencedirect.com/science/article/pii/S0021999122005204 (visited on 06/28/2023).

[6] Z.j. Wang et al. “High-order CFD methods: current status and perspective”. en. In: *International Journal for Numerical Methods in Fluids* 72.8 (2013), pp. 811–845. DOI: 10.1002/fld.3767. URL: http://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3767 (visited on 08/07/2023).

[7] Anthony T Patera. “A spectral element method for fluid dynamics: Laminar flow in a channel expansion”. en. In: *Journal of Computational Physics* 54.3 (June 1984), pp. 468–488. DOI: 10.1016/0021-9991(84)90128-1. URL: https://www.sciencedirect.com/science/article/pii/0021999184901281 (visited on 06/28/2023).

[8] J. M. Hyman et al. “Natural discretizations for the divergence, gradient, and curl on logically rectangular grids”. In: *Computers & Mathematics with Applications* 33.4 (Feb. 1997), pp. 81–104. DOI: 10.1016/S0898-1221(97)00009-6. URL: https://www.sciencedirect.com/science/article/pii/S0898122197000096 (visited on 10/10/2024).

[9] James M. Hyman et al. “Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids”. In: *Applied Numerical Mathematics* 25.4 (Dec. 1997), pp. 413–442. DOI: 10.1016/S0168-9274(97)00097-4. URL: https://www.sciencedirect.com/science/article/pii/S0168927497000974 (visited on 10/10/2024).

[10] Pavel B. Bochev et al. “Principles of Mimetic Discretizations of Differential Operators”. en. In: *Compatible Spatial Discretizations*. Ed. by Douglas N. Arnold et al. The IMA Volumes in Mathematics and its Applications. New York, NY: Springer, 2006, pp. 89–119. DOI: 10.1007/0-387-38034-5_5.

[11] Marc Gerritsma. “Edge Functions for Spectral Element Methods”. en. In: *Spectral and High Order Methods for Partial Differential Equations*. Ed. by Jan S. Hesthaven et al. Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer, 2011, pp. 199–207. DOI: 10.1007/978-3-642-15337-2_17.

[12] Enzo Tonti. “The Formal Structure of Physical Theories”. In: (Feb. 2004).

[13] Jasper Kreeft et al. *Mimetic framework on curvilinear quadrilaterals of arbitrary order*. arXiv:1111.4304 [math]. Nov. 2011. DOI: 10.48550/arXiv.1111.4304. URL: http://arxiv.org/abs/1111.4304 (visited on 08/24/2024).

[14] Marc Gerritsma. “An Introduction to a Compatible Spectral Discretization Method”. en. In: *Mechanics of Advanced Materials and Structures* 19.1-3 (Jan. 2012), pp. 48–67. DOI: 10.1080/15376494.2011.572237. URL: http://www.tandfonline.com/doi/abs/10.1080/15376494.2011.572237 (visited on 08/08/2023).

[15] Jasper Kreeft et al. “Mixed mimetic spectral element method for Stokes flow: A pointwise divergence-free solution”. In: *Journal of Computational Physics* 240 (May 2013), pp. 284–309. DOI: 10.1016/j.jcp.2012.10.043. URL: https://www.sciencedirect.com/science/article/pii/S002199911200647X (visited on 09/29/2023).

[16] A. Palha et al. “A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2D incompressible Navier–Stokes equations”. en. In: *Journal of Computational Physics* 328 (Jan. 2017), pp. 200–220. DOI: 10.1016/j.jcp.2016.10.009. URL: https://www.sciencedirect.com/science/article/pii/S0021999116305071 (visited on 06/28/2023).

[17] Yi Zhang et al. “A MEEVC discretization for two-dimensional incompressible Navier-Stokes equations with general boundary conditions”. In: *Journal of Computational Physics* 510 (Aug. 2024), p. 113080. DOI: 10.1016/j.jcp.2024.113080. URL: https://www.sciencedirect.com/science/article/pii/S0021999124003292 (visited on 09/14/2024).

[18] Marc Gerritsma et al. “A spectral mimetic least-squares method for the Stokes equations with no-slip boundary condition”. en. In: *Computers & Mathematics with Applications*. Proceedings of the conference on Advances in Scientific Computing and Applied Mathematics. A special issue in honor of Max Gunzburger’s 70th birthday 71.11 (June 2016), pp. 2285–2300. DOI: 10.1016/j.camwa.2016.01.033. URL: https://www.sciencedirect.com/science/article/pii/S0898122116300293 (visited on 06/28/2023).

[19] Pavel Bochev et al. “A spectral mimetic least-squares method”. en. In: *Computers & Mathematics with Applications*. Minimum Residual and Least Squares Finite Element Methods 68.11 (Dec. 2014), pp. 1480–1502. DOI: 10.1016/j.camwa.2014.09.014. URL: https://www.sciencedirect.com/science/article/pii/S0898122114004623 (visited on 06/28/2023).

[20] Marc Gerritsma et al. “Mimetic Spectral Element Method for Anisotropic Diffusion”. en. In: *Numerical Methods for PDEs: State of the Art Techniques*. Ed. by Daniele Antonio Di Pietro et al. SEMA SIMAI Springer Series. Cham: Springer International Publishing, 2018, pp. 31–74. DOI: 10.1007/978-3-319-94676-4_3. URL: https://doi.org/10.1007/978-3-319-94676-4_3 (visited on 08/08/2023).

[21] Artur Palha et al. “Mimetic Spectral Element Advection”. In: *Lecture Notes in Computational Science and Engineering* 95 (Apr. 2013). DOI: 10.1007/978-3-319-01601-6_26.

[22] Oksana Guba et al. “Optimization-based limiters for the spectral element method”. In: *Journal of Computational Physics* 267 (June 2014), pp. 176–195. DOI: 10.1016/j.jcp.2014.02.029. URL: https://www.sciencedirect.com/science/article/pii/S0021999114001491 (visited on 11/03/2024).

[23] Yi Zhang et al. “A hybrid mimetic spectral element method for three-dimensional linear elasticity problems”. en. In: *Journal of Computational Physics* 433 (May 2021), p. 110179. DOI: 10.1016/j.jcp.2021.110179. URL: https://www.sciencedirect.com/science/article/pii/S0021999121000747 (visited on 06/28/2023).

[24] David Lee. “Petrov–Galerkin flux upwinding for mixed mimetic spectral elements, and its application to geophysical flow problems”. In: *Computers & Mathematics with Applications* 89 (May 2021), pp. 68–77. DOI: 10.1016/j.camwa.2021.02.017. URL: https://www.sciencedirect.com/science/article/pii/S0898122121000651 (visited on 11/03/2024).

[25] Claudio Mattiussi. “The finite volume, finite element, and finite difference methods as numerical methods for physical field problems”. en. In: *Advances in Imaging and Electron Physics*. Ed. by Peter W. Hawkes. Vol. 113. Elsevier, Jan. 2000, pp. 1–146. DOI: 10.1016/S1076-5670(00)80012-9. URL: https://www.sciencedirect.com/science/article/pii/S1076567000800129 (visited on 08/03/2023).

[26] Claudio Mattiussi. “A reference discretization strategy for the numerical solution of physical field problems”. en. In: *Advances in Imaging and Electron Physics*. Ed. by Peter W. Hawkes. Vol. 121. Electron Microscopy and Holography. Elsevier, Jan. 2002, pp. 143–279. DOI: 10.1016/S1076-5670(02)80027-1. URL: https://www.sciencedirect.com/science/article/pii/S1076567002800271 (visited on 08/07/2023).

[27] D. Lee et al. “Discrete conservation properties for shallow water flows using mixed mimetic spectral elements”. In: *Journal of Computational Physics* 357 (Mar. 2018), pp. 282–304. DOI: 10.1016/j.jcp.2017.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0021999117309166 (visited on 11/03/2024).

[28] D. Lee et al. “A mixed mimetic spectral element model of the rotating shallow water equations on the cubed sphere”. In: *Journal of Computational Physics* 375 (Dec. 2018), pp. 240–262. DOI: 10.1016/j.jcp.2018.08.042. URL: https://www.sciencedirect.com/science/article/pii/S0021999118305734 (visited on 11/03/2024).

[29] D. Fenton et al. “Viscoelastic flows of a lid-driven cavity using spectral element methods”. In: *Journal of Non-Newtonian Fluid Mechanics* (May 2024), p. 105263. DOI: 10.1016/j.jnnfm.2024.105263. URL: https://www.sciencedirect.com/science/article/pii/S0377025724000790X (visited on 06/07/2024).

[30] Xinlei Liu et al. “A comparative numerical investigation of reactivity controlled compression ignition combustion using Large Eddy Simulation and Reynolds-Averaged Navier-Stokes approaches”. en. In: *Fuel* 257 (Dec. 2019), p. 116023. DOI: 10.1016/j.fuel.2019.116023. URL: https://www.sciencedirect.com/science/article/pii/S0016236119313778 (visited on 06/28/2023).

[31] Nicoleta Herzog et al. “A comparative study of different CFD-codes for numerical simulation of gas–solid fluidized bed hydrodynamics”. In: *Computers & Chemical Engineering* 39 (Apr. 2012), pp. 41–46. DOI: 10.1016/j.compchemeng.2011.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0098135411003395 (visited on 08/24/2024).

[32] R. M. Stringer et al. “Unsteady RANS computations of flow around a circular cylinder for a wide range of Reynolds numbers”. In: *Ocean Engineering* 87 (Sept. 2014), pp. 1–9. DOI: 10.1016/j.oceaneng.2014.04.017. URL: https://www.sciencedirect.com/science/article/pii/S0029801814001565 (visited on 09/06/2024).

[33] Woowon Jeong et al. “Comparison of effects on technical variances of computational fluid dynamics (CFD) software based on finite element and finite volume methods”. In: *International Journal of Mechanical Sciences* 78 (Jan. 2014), pp. 19–26. DOI: 10.1016/j.ijmecsci.2013.10.017. URL: https://www.sciencedirect.com/science/article/pii/S0020074031300297X (visited on 09/13/2024).

[34] Alejandro López et al. “CFD study of Jet Impingement Test erosion using Ansys Fluent® and OpenFOAM®”. In: *Computer Physics Communications* 197 (Dec. 2015), pp. 88–95. DOI: 10.1016/j.cpc.2015.07.016. URL: https://www.sciencedirect.com/science/article/pii/S0010465515002945 (visited on 09/13/2024).

[35] H. A. Mohammed et al. “Convective heat transfer and fluid flow study over a step using nanofluids: A review”. In: *Renewable and Sustainable Energy Reviews* 15.6 (Aug. 2011), pp. 2921–2939. DOI: 10.1016/j.rser.2011.02.019. URL: https://www.sciencedirect.com/science/article/pii/S136403211100061X (visited on 09/05/2024).

[36] R. Ruisi et al. “Active flow control over a backward-facing step using plasma actuation”. In: *Acta Astronautica*. Space Flight Safety 126 (Sept. 2016), pp. 354–363. DOI: 10.1016/j.actaastro.2016.05.016. URL: https://www.sciencedirect.com/science/article/pii/S0094576516303010 (visited on 09/05/2024).

[37] Mohammadreza Baigmohammadi et al. “An experimental study of methane–oxygen–carbon dioxide premixed flame dynamics in non-adiabatic cylinderical meso-scale reactors with the backward facing step”. In: *Chemical Engineering and Processing: Process Intensification* 95 (Sept. 2015), pp. 105–123. DOI: 10.1016/j.cep.2015.05.013. URL: https://www.sciencedirect.com/science/article/pii/S025527011530026X (visited on 09/05/2024).

[38] D. Barkley. “Linear analysis of the cylinder wake mean flow”. en. In: *Europhysics Letters* 75.5 (July 2006). Publisher: IOP Publishing, p. 750. DOI: 10.1209/epl/i2006-10168-7. URL: https://iopscience.iop.org/article/10.1209/epl/i2006-10168-7/meta (visited on 09/05/2024).

[39] Yi Zhang. *About — phyem, a finite element library powering LEGO-like simulations for (multi-)physics.* URL: https://phyem.org/tutorial/about (visited on 02/03/2024).

[40] Seymour V. Parter. “On the Legendre–Gauss–Lobatto Points and Weights”. en. In: *Journal of Scientific Computing* 14.4 (Dec. 1999), pp. 347–355. DOI: 10.1023/A:1023204631825. URL: https://doi.org/10.1023/A:1023204631825 (visited on 09/12/2024).

[41] Yi Zhang. *Documentations — phyem, a finite element library powering LEGO-like simulations for (multi-)physics.* URL: https://phyem.org/tutorial/documentations (visited on 02/03/2024).

[42] Enzo Tonti. “General Structure of the Diagrams”. en. In: *The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram*. Ed. by Enzo Tonti. Modeling and Simulation in Science, Engineering and Technology. New York, NY: Springer, 2013, pp. 415–428. DOI: 10.1007/978-1-4614-7422-7_14. URL: https://doi.org/10.1007/978-1-4614-7422-7_14 (visited on 08/08/2023).

[43] Gennaro Coppola et al. “Discrete Energy-Conservation Properties in the Numerical Simulation of the Navier–Stokes Equations”. In: *Applied Mechanics Reviews* 71.010803 (Mar. 2019). DOI: 10.1115/1.4042820. URL: https://doi.org/10.1115/1.4042820 (visited on 08/08/2023).

[44] Ernst Hairer et al. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. en. Springer Science & Business Media, Mar. 2013.

[45] ANSYS. *FLUENT User’s Guide*.

[46] Yunus A. Çengel et al. *Fluid mechanics: fundamentals and applications*. en. 3. ed. New York, NY: McGraw-Hill, 2014.

[47] J. P. Van Doormaal et al. “Enhancements of the Simple Method for Predicting Incompressible Fluid Flows”. In: *Numerical Heat Transfer* 7.2 (Apr. 1984). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01495728408961817, pp. 147–163. DOI: 10.1080/01495728408961817. URL: https://doi.org/10.1080/01495728408961817 (visited on 09/13/2024).

[48] H. Versteeg. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. English. 2nd edition. Harlow, England ; New York: Pearson, Feb. 2007.

[49] Mark A. George et al. “A coupled block implicit solver for the incompressible Navier–Stokes equations on collocated grids”. In: *Computers & Fluids* 284 (Nov. 2024), p. 106426. DOI: 10.1016/j.compfluid.2024.106426. URL: https://www.sciencedirect.com/science/article/pii/S0045793024002573 (visited on 10/22/2024).

[50] Manoj T. Nair et al. “Onset of Asymmetry: Flow Past Circular and Elliptic Cylinders”. en. In: *International Journal for Numerical Methods in Fluids* 23.12 (1996), pp. 1327–1345. DOI: 10.1002/(SICI)1097-0363(19961230)23:12<1327::AID-FLD476>3.0.CO;2-Q. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0363%281996 1230%2923%3A12%3C1327%3A%3AAID-FLD476%3E3.0.CO%3B2-Q (visited on 09/13/2024).

[51] Charles Hirsch. “Basic Discretization Techniques”. en. In: *Numerical Computation of Internal and External Flows (Second Edition)*. Ed. by Charles Hirsch. Oxford: Butterworth-Heinemann, Jan. 2007, pp. 141–144. DOI: 10.1016/B978-075066594-0/50044-8. URL: https://www.sciencedirect.com/science/article/pii/B9780750665940500448 (visited on 08/08/2023).

[52] R. Abbasi et al. “A comparative study of finite volume pressure-correction projection methods on co-located grid arrangements”. In: *Computers & Fluids* 81 (July 2013), pp. 68–84. DOI: 10.1016/j.compfluid.2013.03.014. URL: https://www.sciencedirect.com/science/article/pii/S0045793013001126 (visited on 11/01/2024).

[53] C. M. Rhie et al. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA Journal* 21.11 (Nov. 1983). Publisher: American Institute of Aeronautics and Astronautics, pp. 1525–1532. DOI: 10.2514/3.8284. URL: https://arc.aiaa.org/doi/10.2514/3.8284 (visited on 11/01/2024).

[54] Geoffrey Ingram Taylor et al. “Mechanism of the production of small eddies from large ones”. In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 158.895 (Jan. 1937). Publisher: Royal Society, pp. 499–521. DOI: 10.1098/rspa.1937.0036. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1937.0036 (visited on 08/26/2024).

[55] Tamer A. AbdelMigid et al. “Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using GPU accelerated code”. In: *Alexandria Engineering Journal* 56.1 (Mar. 2017), pp. 123–135. DOI: 10.1016/j.aej.2016.09.013. URL: https://www.sciencedirect.com/science/article/pii/S1110016816302800 (visited on 06/07/2024).

[56] U Ghia et al. “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method”. In: *Journal of Computational Physics* 48.3 (Dec. 1982), pp. 387–411. DOI: 10.1016/0021-9991(82)90058-4. URL: https://www.sciencedirect.com/science/article/pii/0021999182900584 (visited on 08/29/2024).

[57] O. Botella et al. “Benchmark spectral results on the lid-driven cavity flow”. In: *Computers & Fluids* 27.4 (May 1998), pp. 421–433. DOI: 10.1016/S0045-7930(98)00002-4. URL: https://www.sciencedirect.com/science/article/pii/S0045793098000024 (visited on 08/29/2024).

[58] S. P Vanka. “Block-implicit multigrid solution of Navier-Stokes equations in primitive variables”. In: *Journal of Computational Physics* 65.1 (July 1986), pp. 138–158. DOI: 10.1016/0021-9991(86)90008-2. URL: https://www.sciencedirect.com/science/article/pii/0021999186900082 (visited on 08/29/2024).

[59] E. Erturk et al. “Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers”. en. In: *International Journal for Numerical Methods in Fluids* 48.7 (2005). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.953, pp. 747–774. DOI: 10.1002/fld.953. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.953 (visited on 08/29/2024).

[60] Ercan Erturk. “Numerical solutions of 2-D steady incompressible flow over a backward-facing step, Part I: High Reynolds number solutions”. In: *Computers & Fluids* 37.6 (July 2008), pp. 633–655. DOI: 10.1016/j.compfluid.2007.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0045793007001545 (visited on 11/15/2023).

[61] David K. Gartling. “A test problem for outflow boundary conditions—flow over a backward-facing step”. en. In: *International Journal for Numerical Methods in Fluids* 11.7 (1990). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650110704, pp. 953–967. DOI: 10.1002/fld.1650110704. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650110704 (visited on 10/29/2024).

[62] Stuart E. Rogers et al. “An upwind differencing scheme for the incompressible navier–strokes equations”. In: *Applied Numerical Mathematics* 8.1 (Aug. 1991), pp. 43–64. DOI: 10.1016/0168-9274(91)90097-J. URL: https://www.sciencedirect.com/science/article/pii/016892749190097J (visited on 03/11/2024).

[63] Jiten C. Kalita et al. “Triggering asymmetry for flow past circular cylinder at low Reynolds numbers”. In: *Computers & Fluids* 59 (Apr. 2012), pp. 44–60. DOI: 10.1016/j.compfluid.2012.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0045793012000515 (visited on 09/09/2024).

[64] Tapan K. Sengupta et al. “Solving Navier–Stokes equation for flow past cylinders using single-block structured and overset grids”. In: *Journal of Computational Physics* 229.1 (Jan. 2010), pp. 178–199. DOI: 10.1016/j.jcp.2009.09.026. URL: https://www.sciencedirect.com/science/article/pii/S002199910900518X (visited on 09/09/2024).

[65] Kunihiko Taira et al. “The immersed boundary method: A projection approach”. In: *Journal of Computational Physics* 225.2 (Aug. 2007), pp. 2118–2137. DOI: 10.1016/j.jcp.2007

.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0021999107001234 (visited on 08/12/2024).

[66] M. Behr et al. “Incompressible flow past a circular cylinder: dependence of the computed flow field on the location of the lateral boundaries”. In: *Computer Methods in Applied Mechanics and Engineering* 123.1 (June 1995), pp. 309–316. DOI: 10.1016/0045-7825(94)00736-7. URL: https://www.sciencedirect.com/science/article/pii/0045782594007367 (visited on 09/02/2024).

[67] Madeleine Coutanceau et al. “Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow”. en. In: *Journal of Fluid Mechanics* 79.2 (Feb. 1977), pp. 231–256. DOI: 10.1017/S0022112077000135. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/experimental-determination-of-the-main-features-of-the-viscous-flow-in-the-wake-of-a-circular-cylinder-in-uniform-translation-part-1-steady-flow/4283F41DAA4DF8E156788FB12879281E (visited on 09/09/2024).

[68] Marc Gerritsma et al. “Least-Squares Spectral Element Methods in Computational Fluid Dynamics”. en. In: *Advanced Computational Methods in Science and Engineering*. Ed. by Barry Koren et al. Berlin, Heidelberg: Springer, 2010, pp. 179–227. DOI: 10.1007/978-3-642-03344-5_7.

[69] Jeongyoung Park et al. “Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160”. en. In: *KSME International Journal* 12.6 (Nov. 1998), pp. 1200–1205. DOI: 10.1007/BF02942594. URL: https://doi.org/10.1007/BF02942594 (visited on 10/30/2024).

[70] Mark N. Linnick et al. “A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains”. In: *Journal of Computational Physics* 204.1 (Mar. 2005), pp. 157–192. DOI: 10.1016/j.jcp.2004.09.017. URL: https://www.sciencedirect.com/science/article/pii/S0021999104004127 (visited on 09/07/2024).

[71] Mustapha Darif et al. “A new accurate solution method for steady forced convection in an unbounded flow around a circular cylinder in continuum and slip regimes”. In: *International Communications in Heat and Mass Transfer* 159 (Dec. 2024), p. 107997. DOI: 10.1016/j.icheatmasstransfer.2024.107997. URL: https://www.sciencedirect.com/science/article/pii/S0735193324007590 (visited on 09/06/2024).

[72] X. D. Niu et al. “Simulation of flows around an impulsively started circular cylinder by Taylor series expansion- and least squares-based lattice Boltzmann method”. In: *Journal of Computational Physics* 188.1 (June 2003), pp. 176–193. DOI: 10.1016/S0021-9991(03)00161-X. URL: https://www.sciencedirect.com/science/article/pii/S002199910300161X (visited on 09/11/2024).

[73] Tapan K. Sengupta et al. “Dynamical system approach to instability of flow past a circular cylinder”. en. In: *Journal of Fluid Mechanics* 656 (Aug. 2010), pp. 82–115. DOI: 10.1017/S0022112010001035. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/dynamical-system-approach-to-instability-of-flow-past-a-circular-cylinder/384149082FF54AFAC64C574204EAD31C (visited on 09/11/2024).

[74] Q. M. Al-Mdallal et al. “Forced streamwise oscillations of a circular cylinder: Locked-on modes and resulting fluid forces”. In: *Journal of Fluids and Structures* 23.5 (July 2007),

pp. 681–701. DOI: 10.1016/j.jfluidstructs.2006.11.001. URL: https://www.sciencedirect.com/science/article/pii/S088997460600137X (visited on 09/11/2024).

[75] Y. Lecointe et al. “On the use of several compact methods for the study of unsteady incompressible viscous flow round a circular cylinder”. In: *Computers & Fluids* 12.4 (Jan. 1984), pp. 255–280. DOI: 10.1016/0045-7930(84)90009-4. URL: https://www.sciencedirect.com/science/article/pii/0045793084900094 (visited on 09/11/2024).

[76] Mouna Laroussi et al. “Triggering vortex shedding for flow past circular cylinder by acting on initial conditions: A numerical study”. In: *Computers & Fluids* 101 (Sept. 2014), pp. 194–207. DOI: 10.1016/j.compfluid.2014.05.034. URL: https://www.sciencedirect.com/science/article/pii/S0045793014002412 (visited on 08/14/2024).

[77] Hubert L. Meitz et al. “A Compact-Difference Scheme for the Navier–Stokes Equations in Vorticity–Velocity Formulation”. In: *Journal of Computational Physics* 157.1 (Jan. 2000), pp. 371–403. DOI: 10.1006/jcph.1999.6387. URL: https://www.sciencedirect.com/science/article/pii/S0021999199963878 (visited on 09/16/2024).

[78] F. H. Wang et al. “Flow patterns of cross-flow around a varicose cylinder”. en. In: *Journal of Visualization* 8.1 (Mar. 2005), pp. 49–56. DOI: 10.1007/BF03181602. URL: https://doi.org/10.1007/BF03181602 (visited on 11/04/2024).

[79] A. Belov et al. “A new implicit algorithm with multigrid for unsteady incompressible flow calculations”. In: *33rd Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.1995-49. URL: https://arc.aiaa.org/doi/abs/10.2514/6.1995-49 (visited on 09/10/2024).

[80] C. Liu et al. “Preconditioned Multigrid Methods for Unsteady Incompressible Flows”. In: *Journal of Computational Physics* 139.1 (Jan. 1998), pp. 35–57. DOI: 10.1006/jcph.1997.5859. URL: https://www.sciencedirect.com/science/article/pii/S0021999197958599 (visited on 09/10/2024).

[81] S. Marrone et al. “An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers”. In: *Journal of Computational Physics* 245 (July 2013), pp. 456–475. DOI: 10.1016/j.jcp.2013.03.011. URL: https://www.sciencedirect.com/science/article/pii/S0021999113001885 (visited on 09/11/2024).

[82] Farzeen Zehra et al. *Comparative Analysis of C++ and Python in Terms of Memory and Time*. en. Dec. 2020. DOI: 10.20944/preprints202012.0516.v1. URL: https://www.preprints.org/manuscript/202012.0516/v1 (visited on 11/03/2024).

[83] Giovanni Petrone. *Unleashing the Power of Multiple GPUs for CFD Simulations*. en-US. URL: https://www.ansys.com/blog/unleashing-the-power-of-multiple-gpus-for-cfd-simulations (visited on 11/02/2024).

A

Stream Function Calculation

This appendix contains a procedure for calculating stream function Ψ using velocity field u. The stream function, $\Psi(x, y)$, is a scalar function used to describe two-dimensional, incompressible flow fields. The key property of the stream function is that the velocity components $u_x(x, y)$ and $u_y(x, y)$ can be derived from the stream function as:

$$u_x = \frac{\partial \Psi}{\partial y}, \quad u_y = -\frac{\partial \Psi}{\partial x} \tag{A.1}$$

This relationship ensures that the continuity equation for incompressible flow is automatically satisfied. The following steps outline the procedure for calculating the stream function from a known velocity field.

The spatial grid is defined on which the stream function will be computed. Let the grid points be defined as (x_i, y_j) where $i = 1, 2, \ldots, N_x$ and $j = 1, 2, \ldots, N_y$.

The stream function can be computed using numerical integration methods. The fourth-order Runge-Kutta method is particularly effective for this purpose due to its balance between accuracy and computational cost.

To compute $\psi(x, y)$ across the grid following steps were taken. The stream function was initialized at a reference point with $\psi(x_0, y_0) = 0$. The loop is made through each grid point (x_i, y_j) and the stream function is computed using the following Runge-Kutta formulas:

$$\psi_{i+1,j} = \psi_{i,j} + \frac{\Delta x}{6} (k_1 + 2k_2 + 2k_3 + k_4) \tag{A.2}$$

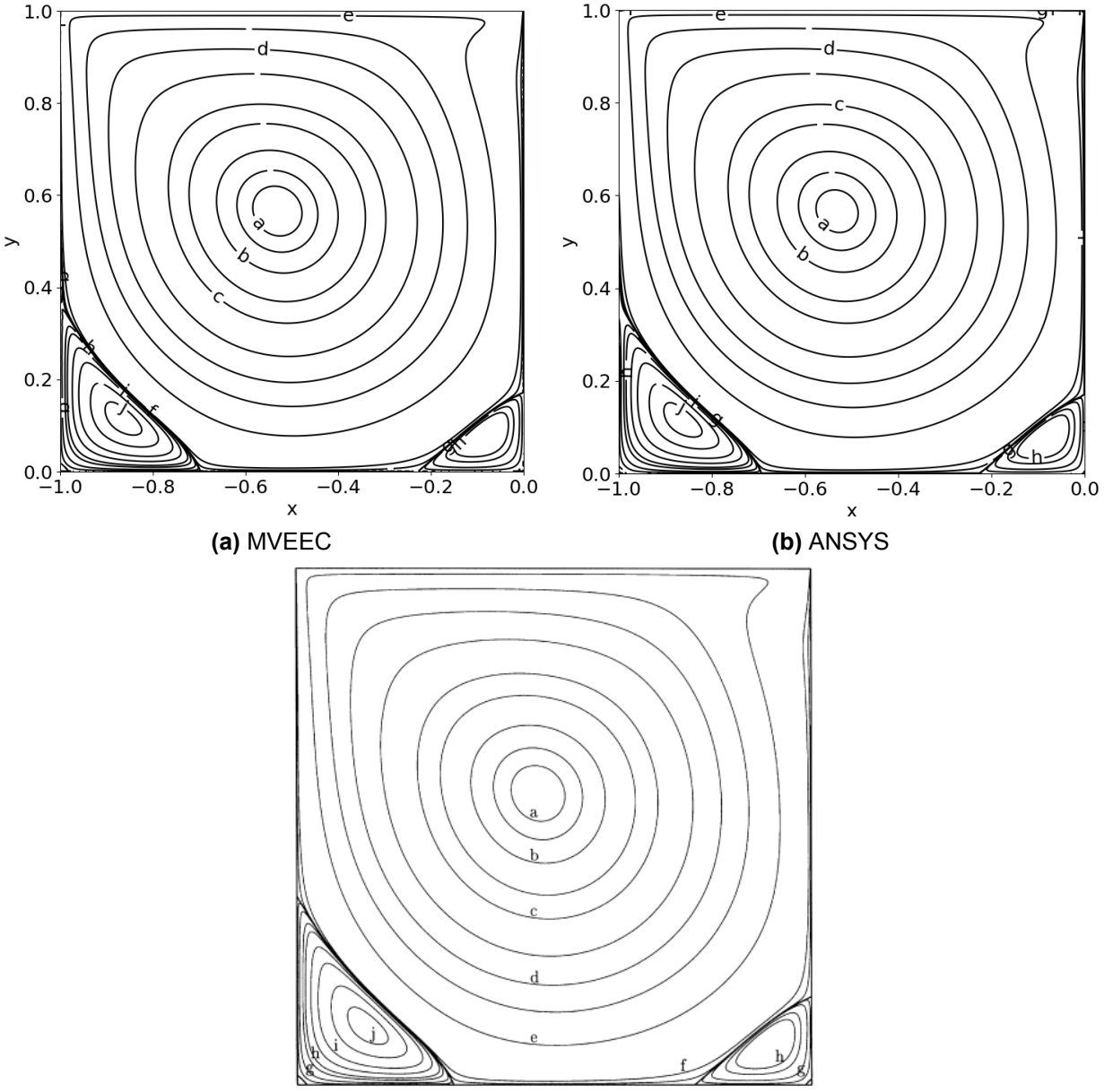
$$\psi_{i,j+1} = \psi_{i,j} - \frac{\Delta y}{6} (k_1 + 2k_2 + 2k3 + k_4) \tag{A.3}$$

where:

$$k_1 = u_{i,j}, \quad k_2 = u_{i,j} + 0.5\Delta y \cdot v_{i,j}, \quad k_3 = u_{i,j} + 0.5\Delta y \cdot v_{i,j}, \quad k_4 = u_{i,j} + \Delta y \cdot v_{i,j} \tag{A.4}$$

Here, k_1, k_2, k_3, and k_4 are intermediate slopes calculated at different points, which are then combined to estimate the stream function at the next grid point.

The example stream function results for the lid cavity can be seen Figure A.1 with reference from [57].



(a) MVEEC

(b) ANSYS

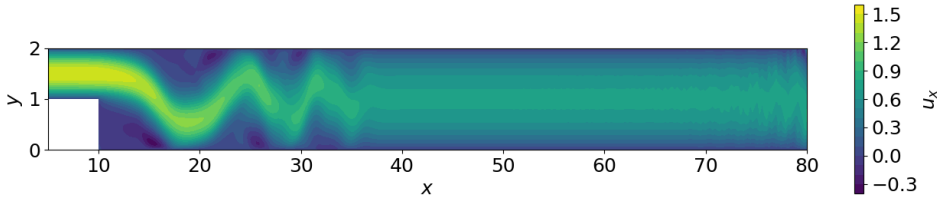
(c) Botella

Figure A.1: MVEEC, ANSYS and Botella stream function Ψ field for lid cavity flow $Re = 1000$, [57].

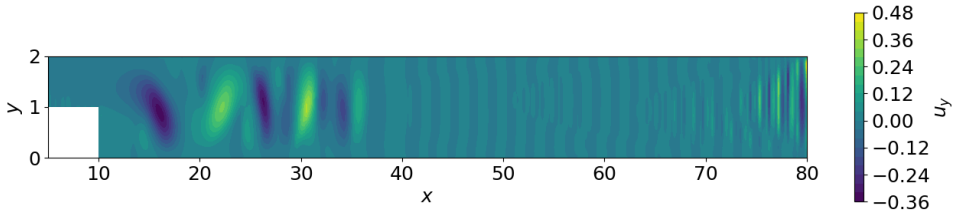
B

Backward facing step at higher polynomial degree N

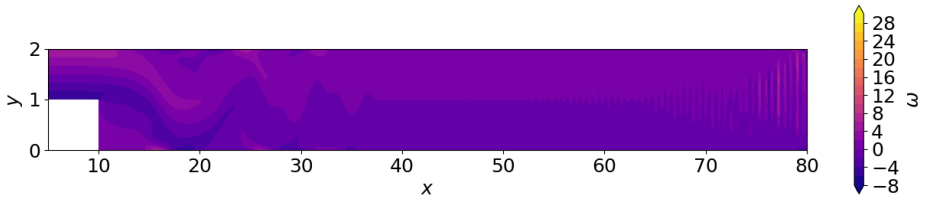
The Figure B.1 shows contour plots for horizontal velocity u_x, vertical velocity u_y and vorticity ω for $N = 3$, generated using the MEEVC method. The timestep is around $t = 45$. As can be seen on the plots, noticeable disturbances are present near the outlet. Most visible in the vertical velocity u_y plot.



(a) Horizontal velocity u_x.



(b) Vertical velocity u_y.



(c) Vorticity ω.

Figure B.1: MEEVC contour plots for $Re = 800$ with $N = 3$.

The disturbances at the outlet progressively intensified over time, resulting in the divergence of the solution and preventing the attainment of a steady-state condition. This outcome was unexpected, particularly given that a steady-state solution was achieved with $N = 2$. Several modifications were implemented in an effort to achieve convergence for the $N = 3$ simulation. First, the outlet length was extended to $l_{outlet} = 120$ to ensure a fully developed flow at the outlet. Second, the mesh near the outlet was refined to better capture potential disturbances. Third, $N = 4$ was tested to rule out any influence from the use of an odd polynomial degree. Then, outlet BC was changed to the outflow so the velocity was prescribed instead of the static pressure. Lastly, the top boundary condition was changed from no-slip to free-slip. In a no-slip condition, the fluid velocity at the boundary is forced to be zero, which can generate shear stresses and contribute to instabilities. Despite these efforts, none of the adjustments succeeded in altering the outcome of the simulation.