Technische Universität München

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Statik

Verification Methodology for Computational Wind Engineering Prediction of Wind Loads on Structures

Ahmed K. S. A. Abodonya

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Prof. Dr. Ir. Bert Blocken
3. Prof. Dr. Riccardo Rossi

Die Dissertation wurde am 14.05.2019 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 13.03.2020 angenommen.

Schriftenreihe des Lehrstuhls für Statik TU München

Band 42

Ahmed K. S. A. Abodonya

VERIFICATION METHODOLOGY FOR COMPUTATIONAL WIND ENGINEERING PREDICTION OF WIND LOADS ON STRUCTURES

München 2019

Veröffentlicht durch

Kai-Uwe Bletzinger
Lehrstuhl für Statik
Technische Universität München
Arcisstr. 21
80333 München

Telefon: +49(0)89 289 22422
Telefax: +49(0)89 289 22421
E-Mail: kub@tum.de
Internet: www.st.bgu.tum.de

ISBN: 978-3-943683-53-0

Abstract

In this thesis, a new credibility assessment framework is developed for computational wind engineering (CWE) simulations. The framework is mainly developed for testing code implementation correctness and estimation of the discretization uncertainty for eddy-resolving, and unsteady simulations. The framework is composed of two main milestones.

First, a modular and flexible procedure for code verification is developed with the ability to test black-box codes. The code verification procedure focuses on the consistency of the code implementation and convergence of field variables. The procedure for code verification consists of analytical benchmarks, either exact or manufactured, with increasing complexity to test the implementation of each term in the Navier-Stokes equation.

Second, the reliability assessment framework has a guideline for the quantification of discretization error/uncertainty. More precisely, guidelines are defined for solution verification. The discretization error/uncertainty estimation is based on the Richardson Extrapolation approach. A solution biased uncertainty estimator is used to account for using unstructured grids, non-uniform refinement, and non-asymptotic solutions. The newly developed framework has a new definition for the measurement of grid size, the manipulation of simulation data with anomalous behavior, and the safety factor definition in the uncertainty quantification of the discretization error. The assessment methodology is suited to both well- and ill-behaved sequences of simulations.

The performance of the assessment methodology is checked with a glimpse of validation with experimental data. Finally, it can be concluded that the developed verification methodology is highly qualified to judge the quality of CWE simulations. Moreover, the generality and modularity of the framework make it applicable to any software environment regardless of the discretization scheme. Consequently, the methodology encourages further research on the identification of the reliability of CWE simulations.

Zusammenfassung

In dieser Arbeit wird ein neues Rahmenwerk zur Glaubwürdigkeitsbewertung für rechnergestützte Windsimulationen (CWE) entwickelt. Der Rahmen wird hauptsächlich für die Prüfung der Korrektheit der Code-Implementierung und die Abschätzung der Diskretisierungsunsicherheit für wirbelauflösende und instationäre Simulationen entwickelt. Das Framework besteht aus zwei Hauptmeilensteinen.

Erstens wird ein modulares und flexibles Verfahren zur Code-Verifikation entwickelt, das die Möglichkeit bietet, Black-Box-Codes zu testen. Das Code-Verifikationsverfahren konzentriert sich auf die Konsistenz der Code-Implementierung und die Konvergenz der Feldvariablen. Das Verfahren zur Codeverifizierung besteht aus analytischen Benchmarks, entweder exakt oder hergestellt, mit zunehmender Komplexität, um die Implementierung jedes Terms in der Navier-Stokes-Gleichung zu testen.

Zweitens verfügt das Rahmenwerk zur Glaubwürdigkeitsbewertung über einen Leitfaden zur Quantifizierung von Diskretisierungsfehlern/Unsicherheiten. Genauer gesagt, werden Richtlinien für die Verifizierung der Lösung definiert. Die Schätzung des Diskretisierungsfehlers/der Unsicherheit basiert auf dem Richardson-Extrapolationsansatz. Ein lösungsverzerrter Unsicherheitsschätzer wird verwendet, um die Verwendung unstrukturierter Gitter, ungleichmäßiger Verfeinerung und nicht asymptotischer Lösungen zu berücksichtigen. Der neu entwickelte Rahmen hat eine neue Definition für die Messung der Gittergröße, die Behandlung von Simulationsdaten mit anomalem Verhalten und für die Definition des Sicherheitsfaktors bei der Unsicherheitsquantifizierung des Diskretisierungsfehlers. Die Bewertungsmethodik eignet sich sowohl für gut als auch für schlecht verhaltene Simulationsfolgen.

Die Leistungsfähigkeit der Bewertungsmethodik wird mit einem Blick auf die Validierung mit experimentellen Daten überprüft. Abschließend kann festgestellt werden, dass die entwickelte Verifikationsmethodik hoch qualifiziert ist, um die Qualität von CWE-Simulationen zu beurteilen. Darüber hinaus macht die Allgemeingültigkeit und Modularität des Rahmens es für jede Softwareumgebung unabhängig vom Diskretisierungsschema anwendbar. Folglich fördert die Methodik weitere Forschungen zur Identifizierung der Zuverlässigkeit von CWE-Simulationen.

Acknowledgments

I would like to express my sincere appreciation to my supervisors, Prof. Dr.-Ing. Kai-Uwe Bletzinger and PD Dr.-Ing. habil. Roland Wüchner for their continuous interest, support, and guidance during this work. I would like to thank the examiners for their efforts in reviewing my dissertation.

I would like to thank Dr. Jordi Cotela for developing the VMS element in KRATOS Multiphysics, his feedback and many fruitful discussions on all thesis matters, and proofreading my thesis. Many thanks to Dr.-Ing Michael Andre and Máté Péntek for having many productive conversations on the CWE Simulations. Mainly, I would like to thank Mohamed Khalil for proofreading my thesis.

To my family, my wife Reem, and my son Adam I am eternally grateful for the many years of patience and unfailing support.

Ahmed K. S. A. Abodonya
Technische Universität München
April, 2019

List of Symbols

$\boldsymbol{\sigma}$	Cauchy Stress Tensor
$\boldsymbol{f}$	External Force
$\boldsymbol{I}$	Second Order Identity Tensor
$\boldsymbol{u}$	Fluid Velocity
Δt	Temporal Discretization Size
Δx	Spatial Discretization Size
Δ	Filter Width
ϵ_h	Discretization Error
η	Kolmogorov length scale
γ	Boundary Edge
$\hat{\phi}_{(x,t)}$	Analytical/MMS Solution
ν	Molecular Viscosity
ν_t	Turbulent Viscosity
ν_{air}	Air Kinematic Viscosity
ν_{eff}	Effective Viscosity
ν_{num}	Numerical Viscosity
Ω	Fluid Domain

ϕ^{γ}	Numerical Solution Variable
ϕ_h	Discrete Solution at Grid (h)
$\phi_{(x,t)}$	Numerical Solution Variable
ρ	Density
ρ_{air}	Air Density
$\tilde{\phi}$	Numerical Solution Variable
$\tilde{p}$	Observed Order of Convergence
C_D	Drag Force Coefficient
C_f	Friction Coefficient
C_L	Lift Force Coefficient
C_p	Pressure Coefficient
C_Y	Cross-Wind Force Coefficient
C_{D_f}	Friction Coefficient
Exp	Experimental Data
F_s	Factor of Safety
f_{KH}	Kelvin_Helmholtz Frequency
f_{vs}	Vortex_Shedding Frequency
$gen-wind$	Numerically Generated Wind
$I_{Z_{ref}}$	Stream-Wise Turbulence Intensity at Reference Height
k_{num}	Numerical Turbulent Kinetic Energy
k_{res}	Resolved Turbulent Kinetic Energy
k_{sgs}	Sub_Grid Scale Turbulent Kinetic Energy
k_{tot}	Total Turbulent Kinetic Energy

L	Integral Length Scale
l_m	Turbulence Length Scale
L_r	Recirculation Length
L_{ux}	Stream-Wise Integral Length Scale
$LES-IQ$	Large Eddy Simulation Index of Quality
$LES-IQ_\nu$	Relative Effective Viscosity Index
n^γ	Boundary Normal
p	Fluid Pressure
p_f	Formal Order of Convergence
p_h	Spatial Order of Convergence
p_t	Temporal Order of Convergence
Pre	Predicted Exact Solution
Re	Reynolds Number
RMS	Root-Mean Square
St	Strouhal Number
T_{air}	Air Temperature
U_∞	Undisturbed Incoming Velocity
$U_{Z_{ref}}$	Stream-Wise Velocity at Reference Height
x^γ	Boundary Edge Point
Z_{ref}	Building Reference Height
3-D	Three Dimentional
ABL	Atmospheric Boundary Condition
AIAA	American Institute of Aeronautics and Astronautics

ASGS	Algebraic Sub_Grid Scales
ASME	American Society of Mechanical Engineering
BC	Boundary Condition
BCA	Bias_Corrected Accelerated
BCA-MBB	Biased-Corrected Accelerated Moving Block Bootstrap
BDF2	Second Order Backward Differentiation Formula
CF	Correction Factor
CFD	Computational Fluid Dynamics
CFL	Courant_Friedrichs_Lewy Number
CI	Confidence Interval
CON	Convection Term
CWE	Computational Wind Engineering
DE	Differential Equation
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
ERS	Eddy Resolving Simulations
FDM	Finite Difference Method
FDS	Fire Dynamics Simulator
FEM	Finite Element Method
FSI	Fluid-Structure Interaction
FVM	Finite Volume Method
GCI	Grid Convergence Index
GDU	Global Deviation Uncertainty

GMV	Grid and Model Variation
h	Cell Size
IICE	Incomplete Iterative Convergence Error
IN	Inertia Term
KH	Kelvin_Helmholtz
LES	Large Eddy Simulation
LSGCI	Least-Squares Grid Convergence Index
LSR1	Least-Squares Root by L. Eça and M. Hoekstra
LSRCONS	Least-Squares Root Conservative Safety Factor
LSRMONTE	Least-Squares Root Monte_Carlo Estimated Safety Factor
M&S	Modeling and Simulation
MBB	Moving Block Bootstrap
MES	Method of Exact Solution
MMS	Method of Manufactured Solution
MNP	Method of Nearby Problems
N-S	Navier-Stokes
ODE	Ordinary Differential Equation
OSS	Orthogonal Sub_Grid Scales
p	Order of Convergence
PDE	Partial Differential Equation
PRES	Pressure Term
QOI	Quantity of Interest
QS	Quasi_Static

RANS	Reynolds-Averaged Navier–Stokes Equations
RE	Richardson Extrapolation
SF	Safety Factor
SQA	Software Quality Assurance
SRQ	System Response Quantity
TGV	Taylor-Green Vortex
TKE	Turbulent Kinetic Energy
TKEDR	Turbulent Kinetic Energy Dissipation Rate
URANS	Unsteady Reynolds-Averaged Navier–Stokes Equations
V&V	Verification and Validation
VIS	Viscous Term
VMS	Variational Multiscale Methods

CONTENTS

We can't solve problems by using the same kind of thinking we used when we created them.

Albert Einstein

Chapter 1

Introduction

In today's world, modern architectural designs and limited spaces led to a vast spread increase in tall buildings and large-span structures, which require light-weight materials, as shown in Figure 1.1. Consequently, the emphasis is added-on safety, human comfort, and serviceability under wind loading. Load assessment, especially wind load, is very challenging for structural engineers. Thus, computational wind engineering (CWE) plays an ever-increasing role in the design, analysis, and optimization of many civil structures. Therefore, it is imperative to build confidence in the results of numerical simulation through reliability assessment.

1.1 Motivation

As reported by Blocken [3], Joel H. Ferziger stated that “The frequently heard argument ‘any solution is better than none’ can be dangerous in the extreme. The greatest disaster one can encounter in computation is not instability or lack of convergence but results that are simultaneously good enough to be believable but bad enough to cause trouble”. Both

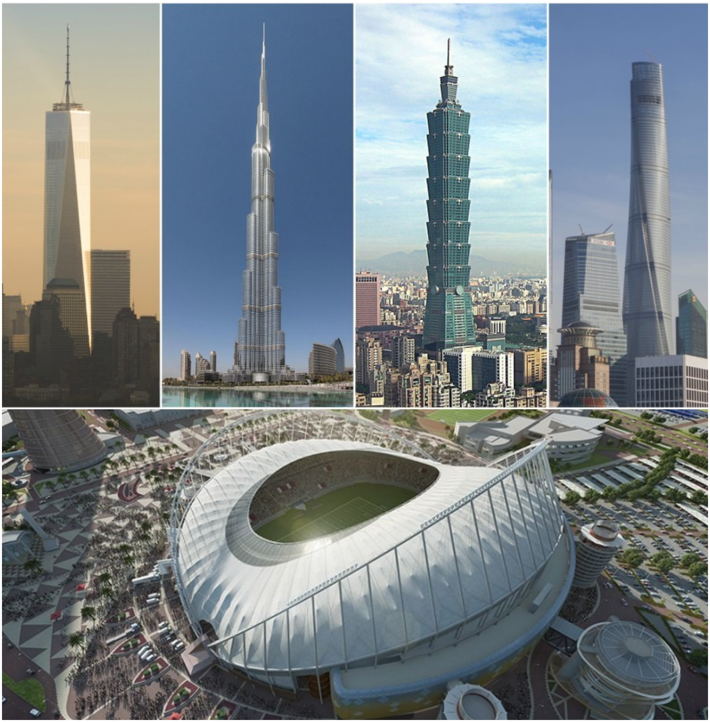


Figure 1.1: Examples of highrise buildings and membrane large span structures, which are very sensitive to wind loads ([1] [2])

experimental and numerical solutions have deficiencies. Thus, whether dealing with experimental or numerical fluid dynamics, the credibility of the results is still questionable. The shortcomings of existent experimental techniques motivated the significant advancement of numerical studies in fluid dynamics. The limitations of the experimental approach are high cost, scaling issues, limited measurement points. In the context of this thesis, only the credibility of numerical results is discussed through verification activities. Figure 1.2 shows the role of verification and validation (V&V) in establishing reliability in numerical simulations.

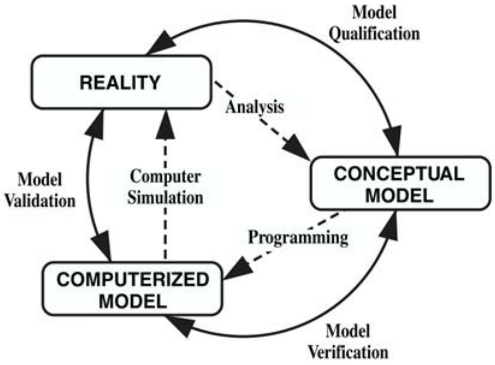


Figure 1.2: The relation between different simulation stages and required credibility assessment practices (Oberkampf et al. [4])

The accuracy of CFD in simulating steady-state flow problems is well established. Several institutions developed guidelines and standards for CFD simulations, such as the American Institute of Aeronautics and Astronautics (AIAA) and the American Society of Mechanical Engineering (ASME). These existing standards are not developed for wind engineering flow problems. Therefore, the reliability of simulating more complex flows are not well studied. In reality, wind engineering flows have a turbulent nature. Thus, eddy-resolving simulations (ERS) are extensively used by researchers to address fundamental questions of turbulence. Turbulence is a complex multi-scale phenomenon that contains a wide range of spatial and temporal scales and exhibits highly nonlinear interactions. Most natural flows are turbulent flows, which are of high interest in the engineering community. Thus, appropriate mathematical models are developed to simulate and examine flows of practical interest (Pereira [5]). Unsteady turbulent flow simulation imposes additional challenges in assessing solution reliability.

This work is mainly focused on the credibility assessment of CWE simulations. CWE is primarily defined as the use of CFD for wind engineering applications. CWE covers a wide range of simulations, such as simulation of pedestrian-level wind conditions, natural ventilation of buildings, wind loads on buildings and bridges, and sports aerodynamics. Moreover, wind loads estimation using CWE simulations is the main objective of practical structural engineering. Despite the strong potential of eddy-resolving simulations (ERS) in solving wind engineering flows, the further development of CWE in simulating natural wind flows as a research and design tool is inevitably dependent on the ability to attain reliable computations. Therefore, different sources of numerical errors and uncertainties should be adequately addressed and quantified. This process is called verification. Figure 1.3 [1] shows detailed phases for modeling and simulation activities with the role of V&V, and uncertainty quantification activities to ensure solution quality.

In Gousseau et al. [7], it is stated that "[...] despite the increasing attention given to the quantification of error and uncertainty in CFD, the techniques that have been developed for general fluid engineering problems to assess the quality of CFD simulations are still marginally used in CWE". The present work proposes a comprehensive framework for the credibility assessment of CWE simulation. Towards this end, a series of code verification and solution verification exercises are performed on well-studied representative flows with a glimpse of validation against experimental data. The role of code and solution verification in modeling phases is highlighted in Figure 1.3. Then, a boundary layer flow around a high-rise building is simulated, and a simulation credibility framework is applied to build confidence in the resulting data.

1.2 Verification of Computer Codes

The verification activities are split into code verification and solution verification, as shown and highlighted in Figure 1.3.

- Code verification is the process of determining that the numerical algorithms are correctly implemented in the computer code, and of identifying errors in the software (Oberkampf et al. [8]). The code verification activity can be performed using the Method of Manufactured Solutions (MMS) or the Method of Exact Solutions (MES) or combining both approaches (Fisch [9]).
- Solution verification is the process of determining the correctness of the input data, the numerical accuracy of the solution obtained,

[1] Revise appropriate model or experiment in Figure 1.3 implies that either the mathematical model or the physical model must be modified to comply with the reality of interest

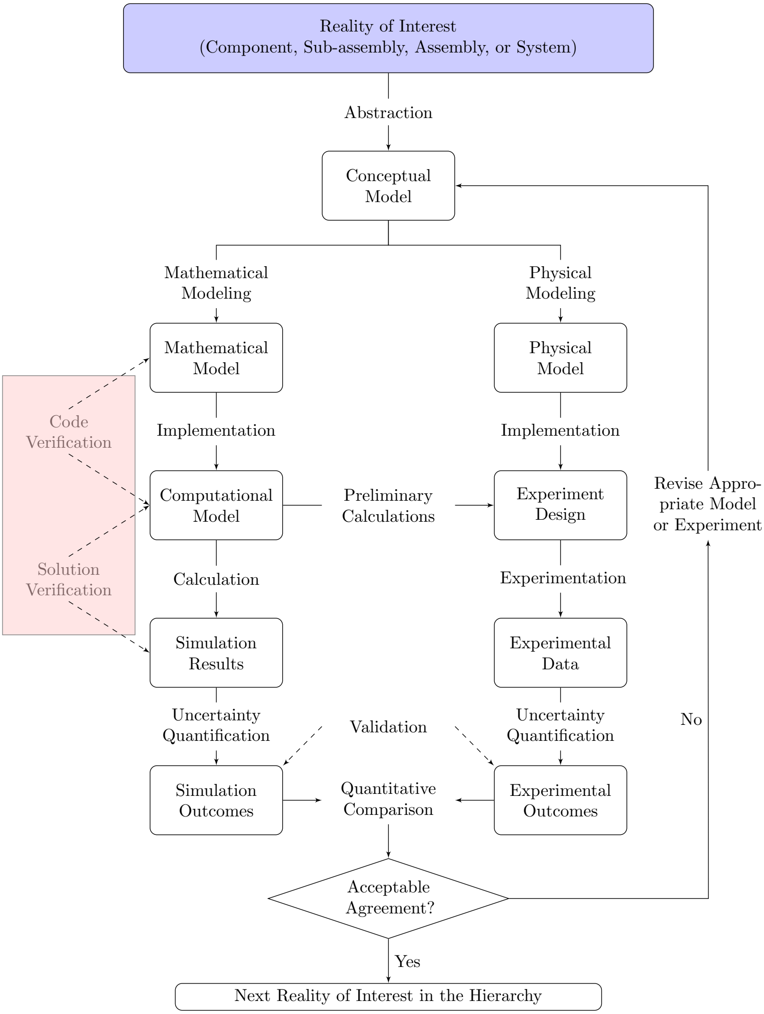


Figure 1.3: Verification, validation, and uncertainty quantification activities required in modeling and simulation phases (ASME V&V10 Committee [6])

and the correctness of the output data for a particular simulation (Oberkampf et al. [8]). It is mainly concerned with issues related to discretization errors. There are several techniques in the literature for solution verification such as: LES index of quality (LES_IQ), grid convergence index (GCI), safety factor method (SF), least-squares fitting method (LSR), etc (Oberkampf et al. [8], Phillips et al. [10], Roache [11], Eça et al. [12], and Celik et al. [13]). Although many techniques exist in the literature, there is no complete framework designed for the credibility assessment of CWE simulations, and even solution verification is still marginally used in CWE (Franke [14]).

1.3 Research Need in CWE Simulations

Murakami [15] has meticulously outlined the main difficulties in CWE: (1) the high Reynolds numbers in wind engineering applications necessitate high grid resolutions, especially in near-wall regions, as well as proper wall functions; (2) the complex nature of the 3-D flow field with impingement, separation, and vortex shedding; (3) the numerical difficulties associated with flow at sharp corners, and its consequences on discretization schemes; and (4) the inflow and outflow boundary conditions which are particularly challenging for ERS.

These difficulties are directly linked to physical modeling limitations and computational requirements in the past, but many of those limitations are still, to some extent, present today (Blocken [3]). Therefore, when thinking of the credibility of CWE simulations, the following issues should be addressed:

1. Unsteady flow: Figure 1.4 shows unsteady flow simulation around a thin half cylinder shell using an eddy-resolving simulation. The inlet boundary has a uniform flow velocity.

2. Simulating natural wind conditions, transient inlet: Figure 1.5 shows a natural wind flow simulation for the same structure in Figure 1.4. The influence of the fluctuating inlet can be clearly seen in the up-

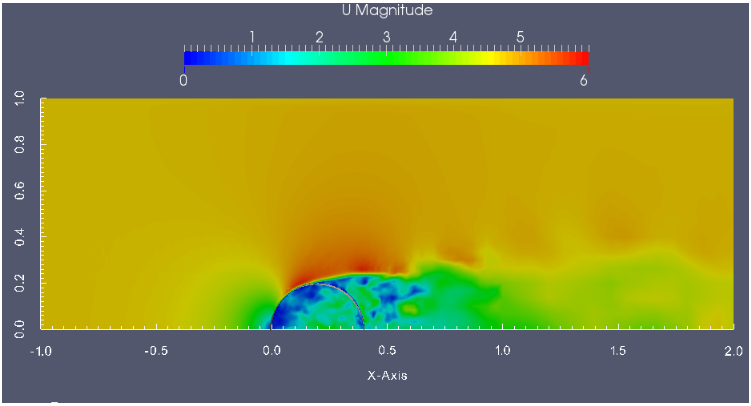


Figure 1.4: Uniform velocity inlet flow around an open half cylinder shell structure (Abodonya [16])

stream zone. Besides having a natural logarithmic wind profile, wind flow characteristics[2] around the structure are entirely different.

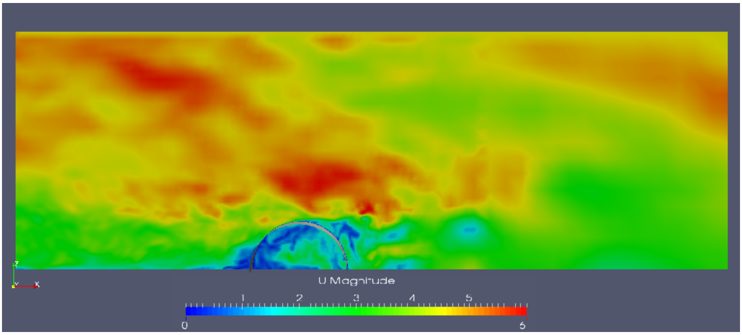


Figure 1.5: Atmospheric boundary layer inlet flow around an open half cylinder shell structure (Abodonya [16])

3. High Re: Typically simulations having $Re \geq 10^5$ are performed,

[2] It is crucial to note, in Figure 1.5, the transient nature of the approaching flow field and the large suction area at the windward side of the shell.

4. Limited computational power,
5. Using large grid sizes,
6. Discrete solutions are not in the asymptotic range. Numerical solutions should be consistent with being in the asymptotic range of convergence. In other words, as the grid is refined, the computed solution will not change significantly, and the asymptotic value (i.e., the actual numerical solution) is reached.
7. Measuring integral quantities (i.e., drag, lift, and crosswind forces) and local quantities (i.e., surface pressure and velocities),
8. Scatter in the results and oscillatory behavior,
9. An observed and formal order of convergence is not equal,
10. Using unstructured grids: Figure 1.6 shows typically used meshing approach for practical flow simulations. It is not always possible to construct structured mesh in practical flow simulations.

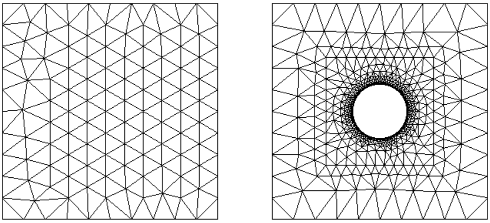


Figure 1.6: Typical unstructured triangular grids. Left figure shows general unstructured grid. Right figure shows practical relevance of non-uniform refinement

11. Using non-uniform refinement: Figure 1.6 right shows mesh around a circle, where different grid sizes are used in different zones. Refinement is typically applied in zones very close to the structure and behind the structure.
12. Eddy-resolving simulations.

Thus, a special treat for all, or some, of the challenges is required. Furthermore, there are different types of error in general CFD simulations, as shown in Figure 1.7.

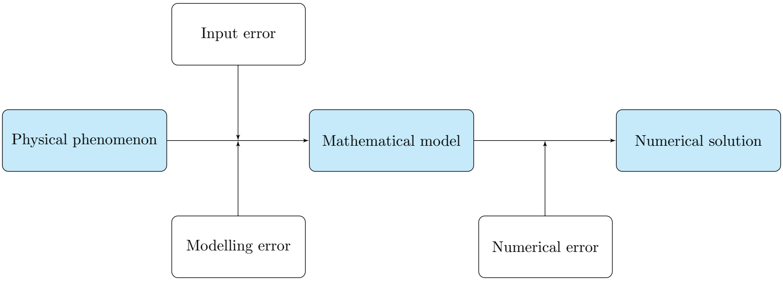


Figure 1.7: Types of errors in general CFD Simulation regenerated from (Pereira et al. [17])

For a better understanding of the error sources in CWE, a detailed error map is presented in Figure 1.8

1.3.1 Input Error

The input error results from the inconsistencies between the parameters and conditions of the physical and numerical problems. These discrepancies result from insufficient experimental information (e.g., inlet flow properties). Moreover, excessive numerical demands (domain dimensions, boundary conditions, total simulation time, turbulent flow properties, etc.) lead to limitations.

1.3.2 Modeling Error

For each physical phenomenon, several mathematical models exist. In wind engineering applications, many modeling options can be used, such as RANS, URANS, LES, DES, VMS. Each of these modeling approaches has several turbulence models. Thus, modeling error evaluation requires a comparison between numerical solutions and experimental/real data.

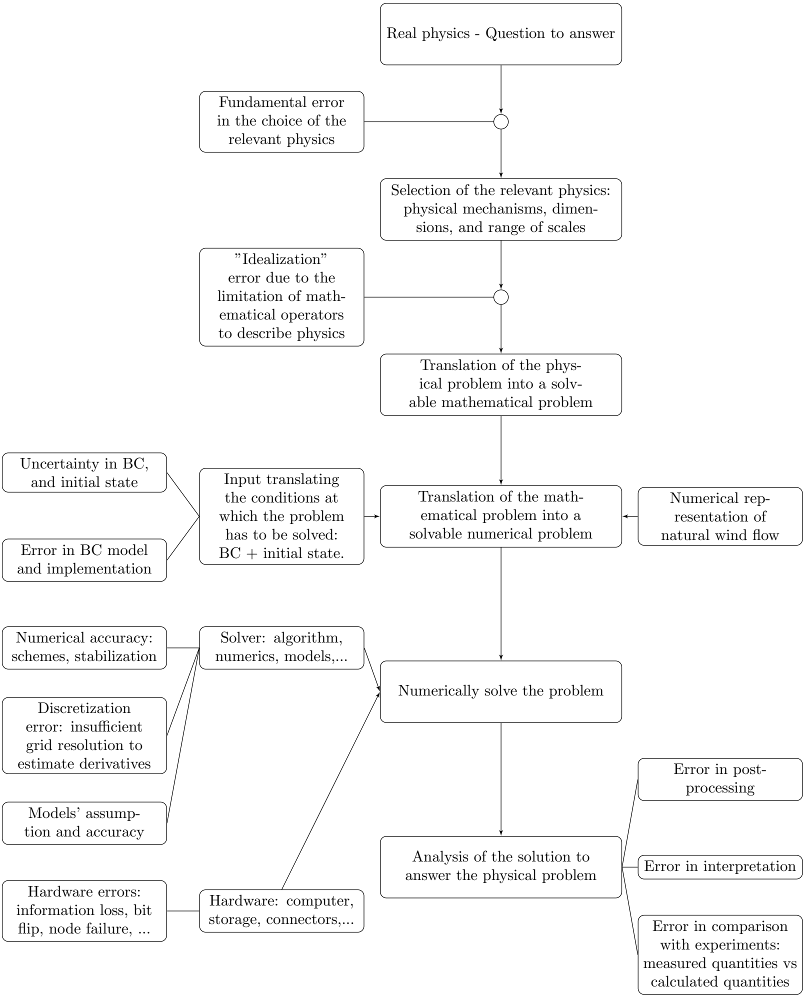

Real physics - Question to answer
Fundamental error in the choice of the relevant physics
Selection of the relevant physics: physical mechanisms, dimensions, and range of scales
"Idealization" error due to the limitation of mathematical operators to describe physics
Translation of the physical problem into a solvable mathematical problem
Uncertainty in BC, and initial state
Error in BC model and implementation
Input translating the conditions at which the problem has to be solved: BC + initial state.
Translation of the mathematical problem into a solvable numerical problem
Numerical representation of natural wind flow
Numerical accuracy: schemes, stabilization
Solver: algorithm, numerics, models,...
Discretization error: insufficient grid resolution to estimate derivatives
Numerically solve the problem
Models' assumption and accuracy
Error in post-processing
Hardware errors: information loss, bit flip, node failure, ...
Hardware: computer, storage, connectors,...
Analysis of the solution to answer the physical problem
Error in interpretation
Error in comparison with experiments: measured quantities vs calculated quantities

Figure 1.8: Error map in computational wind engineering (CWE) (Guilhem Lacaze and Joseph C. Oefelein [18])

1.3.3 Numerical Error

The numerical error is perhaps the most significant contributor to simulation results (Pereira et al. [17]). The importance of putting more emphasis on quantifying numerical error is well recognized by many researchers (Pope [19] and Eca et al. [20]). The numerical error is divided into four contributions: discretization, round-off, iterative, and statistical. The round-off, iterative, and statistical errors can be controlled to some extent by executing all computations on double-precision, using strict convergence criteria for iterative solvers, and using sufficiently large sampling time to achieve statistical convergence. However, discretization error is usually dominating the error term. It stems from the discretization of the governing equations, which is highly dependent on the quality of both spatial and temporal grid resolution, and discretization scheme (Pereira et al. [17]). In practical applications, unstructured grids are used, which is required to represent complex geometries. Unstructured grids limit the convergence order of discretization schemes, and maintaining high-quality grids is not always possible. Thus, both spatial and temporal resolutions highly influence the solution quality, making them the main contributor to numerical errors.

1.4 Research Objectives

In the scope of this work, only verification activities are of interest. Thus input and modeling errors are not considered. Given that numerical errors are the most significant contributor to uncertainty in simulation results while maintaining the same modeling and input parameters for all simulations, this research is dedicated to developing a methodology for verifying CWE simulations, particularly issues related to numerical errors. The terms error and uncertainty are frequently repeated throughout this research. In the *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* [21], the terms error and uncertainty are defined as:

- Error: ”A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge.”
- Uncertainty: ”A potential deficiency in any phase or activity of the modeling process that is due to the lack of knowledge.”

Throughout this thesis, the term uncertainty will refer to the uncertainty estimator used to account for lack of knowledge in the parameters defining the error estimator. Theodore Stathopoulos said, ”In spite of some interesting and visually impressive results produced with Computational Wind Engineering, the numerical wind tunnel is still virtual rather than real. Its potential however, is extremely high and its progress should be monitored carefully. Many more parallel studies - numerical and experimental - will be necessary in order to increase the present level of confidence in the computational results. Practitioners should be warned about the uncertainties of the numerical wind tunnel results and urged to exercise caution in their utilization. Finally, more effective efforts should be made in the numerical simulation of fluctuating flow field and the numerical evaluation of peak values of variables necessary for structural design.” Stathopoulos [22]. This work aims to check the correctness of code implementation and the quantification of numerical errors resulting from domain discretization. Therefore, a simulation credibility assessment framework is introduced in this thesis. The driving force behind this work can be summarized as following (Eça [23]):

- CFD codes should be checked for coding bugs to ensure consistency and build reliability
- CFD simulations require the assessment of their numerical uncertainty to establish their credibility,
- Methods for uncertainty estimation based on grid refinement studies (statistically steady flows) are available in the literature,
- CFD applications are no longer restricted to (statistically) steady flows,
- In statistically unsteady flows, numerical errors are more complicated to handle. Iterative and discretization errors have more than one contribution,
- Guidelines should be developed for quality assessment of practical calculations of unsteady flows, since there is no technical reason to present numerical simulations of practical calculations without the indication of numerical uncertainty (Eça et al. [24]).

The following criteria should be satisfied to achieve the objectives of this work. The assessment framework is to:

1. comprise a series of benchmarks for code verification of eddy-resolving simulations.

2. be applicable for eddy resolving simulations.

3. be robust. The software user or software developer can use the assessment framework, without any knowledge of the implementation details. It can be used for black-box testing. Finally, the methodology should be an a-posteriori-based approach.

4. use statistical tools to identify the quality of the results by checking the statistical convergence on the measured quantity.

5. develop testing procedures for the estimation of numerical error. These procedures should include a numerical error estimator, which can comply with the unsteady nature of the problem, solutions not in the asymptotic range, using non-uniformly refined unstructured grids, and non-monotonically converging data either local or integral quantities.

Finally, the proposed framework is used to assess the simulation quality for different practical flows such as flow around a circular cylinder at $Re = 3.90x10^3$, flow around a square prism at $Re = 3.90x10^3$, and atmospheric boundary layer (ABL) flow around a high-rise building at $Re \approx 10^8$ in full scale.

1.5 Thesis Outline

The following chapters in this dissertation are structured as follows:

- Chapter 2 presents an intensive literature review of the verification activities needed to evaluate software quality. The review starts with code verification and different methods to perform code testing. Also, the limitations and hints for application are introduced. Then, the literature on solution verification is presented, and different types of numerical errors are explained. This is followed by a review of techniques found in the investigated literature for a-posteriori error estimation is presented. The primary purpose is to indicate the shortcoming of these techniques for CWE, because they are developed for different applications. The chapter concludes with a summary of all the solution verification methods presented.

- Chapter 3 presents the newly proposed integrated simulation credibility assessment methodology. The chapter starts by listing all the expected challenges in building confidence in CWE simulations. Then, the main building blocks for the methodology are presented in detail in Section 3.2 and Section 3.3. In Section 3.2, detailed steps to test the correctness of code implementation through the order of convergence studies are explained. In Section 3.3, elements contributing to numerical error are introduced. Then, methods to treat/quantify each error term are presented. The numerical error types are either avoided or quantified, as explained in that chapter. Afterward, small examples are presented to clarify the performance of the numerical error/uncertainty [3] estimation methods.

- Chapter 4 explains the theoretical background of the code that is to be verified using the methodology presented in Chapter 3. This chapter introduces the numerical technique and mathematical model used to solve the physical problem. This code is used as a demonstrative example of applying the newly proposed credibility assessment methodology and its performance.

[3] In the context of this work, uncertainty is measured to account for the doubts in error estimation.

- Chapter 5 presents a series of test cases to perform a code verification study for the monolithic variational multiscale (VMS) element implemented in the FEM-based KRATOS Multiphysics software. The benchmarks are presented in a sequence of increasing complexity. The first benchmark is widely used in the literature, Taylor-Green Vortex, with an exact solution for a decaying vortex. Therefore, it can be used to test any black-box code. If the code does not pass the tests, access to the code is mandatory to perform the remaining tests, which help to identify the source of error. Finally, these benchmarks can be used to verify any other code implementation.

- Chapter 6 shows the application of the solution verification methodology to well-studied flows in the literature, such as 3-D flow around a circular cylinder and 3-D flow around a square prism. Both cases are simulated at $Re = 3.9x10^3$. The purpose of these cases is to apply the solution verification method on flows of practical relevance after applying a successful code verification.

- Chapter 7 presents a real wind simulation, $Re \approx 10^8$, where boundary layer flow around a high-rise building is simulated. Expected challenges in assessing the quality of ERS in CWE are exercised, and the performance of the methodology is tested.

- Chapter 8 concludes the dissertation with a discussion on the performance of the proposed methodology, a summary of the main findings, and a general discussion about ERS's ability to predict wind loads with a credibility measure. Finally, suggestions for future developments are presented.

Software never was perfect and won't get perfect. But is that a license to create garbage? The missing ingredient is our reluctance to quantify quality.

Boris Beizer

CHAPTER 2

VERIFICATION: LITERATURE REVIEW

In this chapter, an extensive literature review on the verification activities is presented. First, code verification techniques are introduced with emphasis on previous usage, limitations, and advantages of each method. Second, almost all the methods used for solution verification, in other words, quality measure, are introduced with special focus on methods applicable to eddy-resolving simulations. A summary of solution verification approaches is presented at the end of the chapter. Before starting the literature review, it is imperative to note that several standards have been developed for V&V of modeling and simulation. Table 2.1 presents a historical overview of previously developed guidelines and standards for simulation quality assurance. All the proposed approaches in these standards are presented in this chapter. Moreover, a detailed reading in these standards can be found in ASME V&V10 Committee [6], *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* [21], NASA [25], ASME V&V20 Committee [26], and ANS [27].

1985	American Nuclear Society “Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry”
1998	American Institute of Aeronautics and Astronautics/Computational Fluid Dynamics Committee on Standards “Guide for the Verification and Validation of Computational Fluid Dynamics Simulations” (AIAA G-077-1998)”
2003	U.S. Department of Defense, DoD Instruction 5000.61, Defense Modeling and Simulation Office “DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)”
2006	American Society of Mechanics Engineers, V&V Standards Committee V&V10 “Guide for Verification and Validation in Computational Solid Mechanics (ASME V&V-10-2006)”
2008	National Aeronautics and Space Administration “Standard for Models and Simulations (NASA-STD-7009)”
2009	American Society of Mechanics Engineers V&V Standards Committee V&V-20 “Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer”

Table 2.1: A brief history of previously developed V&V standards

2.1 Code Verification

Numerical methods to solve *ordinary differential equations (ODE), differential equations (DE),* and *partial differential equations (PDE)* are widely used in the scientific and engineering communities. The increasing complexity of the numerical computer codes is due to the increasing physical complexity of interest phenomena. Computer codes/software complexity triggers the need for credibility and confidence. The reliability of computer code is built through *Software Quality Assurance (SQA)*. The SQA, which is mostly related to code verification application, is developed for software testing (Salari et al. [28] and Veluri [29]). Testing approaches are divided into three categories: static, dynamic, and formal testing. At this stage of building confidence in the code, the word error means a code bug that results in a static or dynamic or formal error. The three types of testing approaches are:

- Static tests mean tests performed without running the code. The code is checked for programming language consistency and being free from compilation errors,
- Dynamic test errors are detected by running the code. The code is tested for out of bounds array indices, memory leakage, coverage issues, and any other logical compiler problems (Salari et al. [28] and Marchand [30]). Dynamic testing is the base of code verification for testing of engineering software,
- Formal tests are related to identifying unnecessary calculations that slow down the code performance.

In the context of this work, dynamic tests are considered. Dynamic errors can influence code efficiency, convergence, and order of convergence such that (Salari et al. [28]):

1. Errors affecting the order of accuracy are the basis for code verification. The order of accuracy mistakes are causing a reduction of functional order of accuracy to zeroth, or below, and reduction to a lesser degree. The former error type can be defined as a consistency mistake. Finally, a numerical scheme is named consistent, if the discretized equations approach the actual governing equations in the limit as the discretization parameters $(\Delta x, \Delta t)$ approach zero (Oberkampf et al. [8] and Fisch [9]).

2. Convergence mistakes are caused mainly by bad input and dynamic coding mistakes, which lead to solution divergence. A code is called converging if the exact solution of the discrete equations approach the exact solution of the governing equations in the limit as the discretization parameters $(\Delta x, \Delta t)$ approach zero (Oberkampf et al. [8] and Fisch [9]).

3. Efficiency errors prevent the code from attaining its theoretical performance.

Salari et al. [28] and Knupp et al. [31] classified the process of dynamic tests related to code verification into two phases: test selection and acceptance criteria. Table 2.2 summarizes the approaches for dynamic testing.

Dynamic Testing Approaches	Acceptance Criteria
Trend	Expert Judgment
Symmetry	Expert Judgment
Comparison	Expert Judgment and Percent Error
MES	Percent Error, Consistency and Order of Convergence
MMS	Percent Error, Consistency and Order of Convergence

Table 2.2: The acceptance criteria for different dynamic code verification approaches

Further discussion about trend, symmetry, and comparison can be found in Oberkampf et al. [8], Roache [11], Salari et al. [28], and Veluri [29]. In the scope of this work, MES and MMS are used as the basis for code verification based on the order of convergence studies.

2.1.1 The Method of Exact Solutions (MES)

The MES is more strict and objective than both trend and comparison tests. MES is widely used in the code verification activities following specific procedures (Salari et al. [28]):

1. Seeking exact solutions for the set of equations solved by the code. The exact solution is a closed-form mathematical expression that conforms with material properties, initial conditions, and boundary conditions, and evaluates the solution at all space and time points.

2. Running the code with corresponding inputs to compute the discrete solution.

3. Comparing the discrete solution with the exact solution and check the acceptance criterion. If the code passes the prescribed test suites, the code is verified and has a small probability of containing coding mistakes. Otherwise, a bug is suspected.

With the increasing complexity of computer codes to be verified, the MES has some limitations:

- Finding exact solutions is very complicated and sometimes impossible for governing equations, involving non-linearity, coupling between equations, complex boundary conditions, and complex domain geometry (Salari et al. [28]).
- Modern computer codes simulate two and three-dimensional domains, which are very complicated for getting an exact solution. Also, it is impractical, as it is assumed that exact solutions are obtained for infinite or semi-finite domains that can not be achieved in a computed simulation.

2.1.2 The Method of Manufactured Solutions (MMS)

Steinberg et al. [32] invented the MMS. The MMS can be seen as a derived class from the MES. The MMS is very flexible and uses analytical solutions like the MES. The manufactured solution is neither dependent on the code being verified nor on the governing equations. Thus, it is completely arbitrary (Brglez [33]). The main target for efficiently verifying a code is finding solutions that would exercise all pieces, subdivisions, and extensions of the code. The MMS has already been used successfully in fluid dynamics (Fisch [9], Marchand [30], Roy et al. [34], Eça et al. [35], Luís et al. [36], Étienne et al. [37], Yu et al. [38], Choudhary et al. [39], Roache [40], Choudhary et al. [41], Veluri et al. [42], Eça et al. [43], Étienne et al. [44], and Eça et al. [45, 46])

The MMS provides a general procedure for generating an analytical solution of conservation equations, e.g., PDE. The MMS does not necessarily yield physical realism. Instead, it is purely a mathematical exercise; the manufactured solutions do not fulfill the actual PDE. Consequently, an additional source term is added to the algebraic discretized governing equations. Thus, code accessibility is required to impose the source/force terms. The acceptance criterion is the order of convergence, at which the code is said to be verified if the observed order of convergence matches the formal order of convergence.

2.1.2.1 General Procedures of the MMS

The general procedures of the MMS are summarized in Knupp et al. [31] and are shown in Figure 2.1.

2.1.2.2 Requirements of a Good Manufactured Solution

When designing a manufactured solution to rigorously assess a numerical code, solving a complex equation, or system of equations, it is necessary to compare the numerical results with an exact, or at least a more reliable solution. Thus, it is vital to follow strict guidelines that ensure the test cases' validity and practicality. According to Fisch [9], Salari et al. [28], and Knupp et al. [31], the requirements for a good and reliable manufactured solution are that it should be

- composed of sufficiently smooth and non-trivial functions.
- general to exercise all pieces, subdivisions, and every term in the governing equations.
- having a sufficient number of non-trivial derivatives. Also, the solution derivatives should be bounded by a small constant, and not a strongly varying function of space, time, or both.
- not preventing the code from completing a successful run during testing.
- preferably composed of simple analytic functions, such as trigonometric, exponential, and polynomials.
- free from singularities.
- in a realistic range of the designed application, for example, no negative density.
- defined on a connected subset of the dimensions in space.

Finally, it is crucial to reiterate that the manufactured solution is not necessarily physically realistic because the code cannot differentiate between physical and non-physical problems.

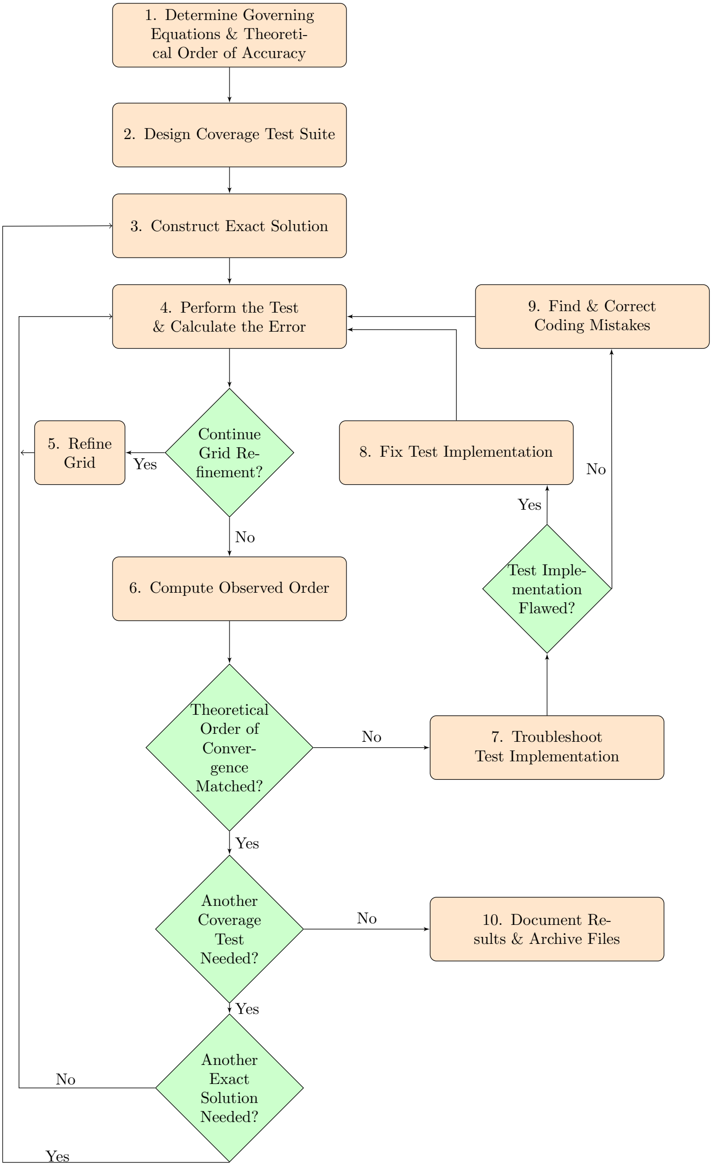

1. Determine Governing Equations & Theoretical Order of Accuracy
2. Design Coverage Test Suite
3. Construct Exact Solution
4. Perform the Test & Calculate the Error
9. Find & Correct Coding Mistakes
5. Refine Grid
Yes
Continue Grid Refinement?
8. Fix Test Implementation
No
Yes
No
6. Compute Observed Order
Test Implementation Flawed?
Theoretical Order of Convergence Matched?
No
7. Troubleshoot Test Implementation
Yes
Another Coverage Test Needed?
No
10. Document Results & Archive Files
Yes
No
Another Exact Solution Needed?
Yes

Figure 2.1: Flowchart for the code verification procedures reproduced from Knupp et al. [31]

2.1.2.3 Initial and Boundary Conditions

Every PDE solution must have auxiliary conditions. These auxiliary conditions are boundary and initial conditions. Only boundary conditions are required for space-dependent PDE *(steady-state code solution)*. On the other hand, both boundary and initial conditions are needed for space-time dependent PDE *(unsteady code solution)*. In the MMS, the boundary and initial conditions can be derived from the solution directly $\phi_{(x,t)}$ (Fisch [9], Salari et al. [28], and Knupp et al. [31]). Most of the PDE codes provide a wide range of boundary conditions to choose from in practical applications such as Dirichlet, Neumann, mixed, slip, no-slip, periodic,...etc. A detailed discussion on these different types of boundary conditions can be found in Oberkampf et al. [8], Salari et al. [28], and Knupp et al. [31]. It is recommended to verify the PDE code for all possible boundary conditions. A sequential suite of test cases is to be designed. The designed test suite implies testing, first, the interior equation implementation with Dirichlet boundary condition for all sides of the domain, then, designing test cases to verify each type of B.C available in the numerical code. The increasing complexity of the test suites renders a more straightforward recognition for coding errors. To sum up, the assessment of steady-state solution requires an initial condition significantly different from the analytical solution $\left(\hat{\phi}_{(x,t=0)}\right)$. In other words, the initial field values are set to zero to approve the code's ability to solve steady-state problems. In contrast, the assessment of unsteady solution requires the initial condition to match the analytical solution $\left(\phi_{(x,t=0)} = \hat{\phi}_{(x,t=0)}\right)$ to avoid any artificial errors. According to Fisch [9], the Dirichlet and Neumann boundary conditions at the boundary γ, at the boundary position x^γ, with the boundary normal n^γ can directly be calculated respectively derived from the manufactured solution field $\hat{\phi}_{(x,t)}$. The boundary field of a variable ϕ is abbreviated by:

$$\phi^\gamma = \phi_{(x=x^\gamma,t)} \tag{2.1}$$

Using the abbreviation in Equation (2.1), the Dirichlet boundary condition is defined in Equation (2.2)

$$\phi^\gamma = \hat{\phi}^\gamma = \hat{\phi}_{(x=x^\gamma,t)} \tag{2.2}$$

2.1.2.4 Code Verification Assessment for Steady and Unsteady Simulations

Generally speaking, the discretization error is evaluated by comparing the variable $\phi_{(x,t)}$ to the manufactured solution $\hat{\phi}_{(x,t)}$. A detailed description of the evaluation of global error and order of convergence will be explained in Section 3.2.1 and Section 5.3. The testing of steady-state simulations is relatively simple. The error of the converged solution is evaluated with grid refinement to compute the observed order of convergence.

The testing of unsteady simulations is challenging because the simulation code has both spatial and temporal orders of convergence. The unsteady simulation assessment can be performed on three levels (Fisch [9]):

- Assess the spatial convergence order only: a steady-state solution can be used, in which the time contribution to the PDE vanishes,
- Assess the temporal convergence order only: the spatial contribution to the PDE should vanish. Two approaches can achieve the assessment of temporal convergence order. Firstly, create an arbitrary manufactured solution that equally covers all terms of the PDE, and use a superfine grid for all time refinement steps. This way assures a minimal contribution from the spatial discretization errors compared to temporal discretization errors. Second, a manufactured solution is chosen that fulfills the solution requirements only over time, and guarantees the solution is mesh-independent,
- Assess the space and time order of convergence together: this approach has the advantage that correlated space-time errors can additionally be assessed. Moreover, it requires equal formal order of convergence in both space and time. Also, the refinement ratios in space and time must be of the same order. Finally, the manufactured solution should fulfill the PDE solution requirement in space and time.

2.1.2.5 Strengths and Limitations of the MMS

The strengths and limitations are summarized in this section, and further discussion can be found in Oberkampf et al. [8], Roache [11], Salari et al.

[28], and Knupp et al. [31]. The strengths as mentioned in Marchand [30] are:

- Most code capabilities can be verified with the MMS procedure. The use of an arbitrary solution that can be applied to an arbitrary geometry and flexibility in the BCs is the main factor behind this method's strength,
- The method allows choosing any geometry of interest because the solution domain can be selected after the manufactured solution has been generated,
- A symbolic toolbox can be used to generate source terms,
- Easily evaluable analytic functions are used to construct the manufactured solution,
- Code debugging can be achieved by elimination, in case of errors,
- A wide variety of numerical techniques can be verified using the MMS,
- The MMS is self-correcting. Source terms implementation errors can be detected by the MMS if existing.

The limitations, as mentioned in Marchand [30], are summarized as:

- The MMS procedure is not simple and needs access to the simulation code if the implementation of the numerical technique lacks volumetric source terms,
- MMS can not detect all coding mistakes, because not all mistakes affect accuracy, such as efficiency mistakes. These mistakes, however, do not prevent the method from showing correctness,
- There is no clear methodology in the verification of mixed-order methods.

The MMS is the most widely used technique in the literature. Table 2.3 shows the usage of the MMS in the previous years in verifying different software environments and discretization schemes.

1985	The MMS was invented by Steinberg and Roache
2004	MMS for Euler/Navier-Stokes codes compressible flow (FVM)
2006	MMS for FSI code verification
	MMS verification of turbulence models
2008	MMS for unsteady flow solvers (FDM)
2009	MMS was used as a basis for testing solution verification techniques (FVM)
2012	MMS for FSI codes (FVM)
2013	MMS was applied for Computational Electromagnetic codes (FEM)
2014	MMS was applied for partitioned FSI environment RANS and URANS (FVM)
	MMS for URANS unsteady solver with turbulence model (FVM)
2015	MMS for FSI code 2-D and 3-D (RANS and URANS) (FVM)
	MMS for BC of compressible and incompressible flows (FVM)
	MMS for Multi-phase flows

Table 2.3: Brief chronological development of code verification using MMS

2.2 Solution Verification

According to Oberkampf et al. [8], verification in general means, "the process of assessing software correctness and numerical accuracy of the solution to a given mathematical model". Verification activity is divided into code verification and solution verification. The strong need for a solution verification approach is for two reasons: the first is that the exact solution is unknown. The second is that the knowledge of the error is as significant as the solution knowledge. In this section, an extensive literature review of solution verification approaches is presented with an emphasis on the applicability of these methods to unsteady turbulent flows.

The primary role of solution or calculation verification is to estimate the numerical errors resulting from a mathematical model’s discretization. In solution verification, the solution to the mathematical model is not known. Thus, the role of solution verification is to estimate the numerical error, and not simply evaluate the error, as in code verification. Numerical errors are calculated with a measure of uncertainty [1]. Since the quality of solutions used to estimate the error is not guaranteed, numerical errors are defined with uncertainty. Therefore, calculation verification includes not only methods for error estimation, but also uncertainty quantification. There are defined procedures by which numerical estimates can be converted to numerical uncertainty estimates.

The numerical solutions of continuum models unquestionably lead to errors in the calculated results. The challenge imposed by using numerical solutions is error estimation. Sources of numerical errors can be generally listed as follows:

1. Round-off errors.
   Round-off error is an inevitable type of error arising due to the use of finite arithmetic on computers (Denis et al. [47]). Round-off errors can be minimized by using more significant digits (DeBonis [48]).

2. Statistical sampling errors.
   Statistical sampling errors only appear when dealing with unsteady simulations for measuring both local and integral system response quantities *(SRQ)* such as pressure, drag, lift coefficients, turbulent kinetic energy...etc.

3. Iterative solver errors.
   Iterative error is defined in Oberkampf et al. [8] as the difference between the current approximate solution to an equation or system of equations, and the exact solution. It is resulting from solving the algebraic equation system using iterative methods. The use of a direct solver can abstain iterative errors, but this approach has become impractical due to efficiency reasons. Consequently, iterative solvers are the usual approach. This error can be avoided by imposing strict convergence criteria and a large number of iterations.

4. Discretization errors.
   Discretization errors are the result of transforming the continuum equation into discrete equations in space and time. As described in Oberkampf et al. [8], discretization errors are formally defined as the

[1] This is the reason why numerical error estimation is referred to as numerical uncertainty

difference between the exact solution to the discrete equations and the exact solution to the mathematical model:

$$\epsilon_h = \phi_h - \tilde{\phi} \tag{2.3}$$

Because discretization error has a locally-generated and a transported component, it is considered the most challenging type of numerical error, to be accurately estimated. Also, it represents the most considerable contribution to the total numerical error.

There are several approaches for calculation verification in the literature, which will be introduced in Section 2.2.1.

2.2.1 Solution Verification: Available Approaches

A variety of discretization error estimators have been developed where some have been developed for specific discretization schemes and others are general (Phillips [49]). Discretization error estimators can be either recovery or residual-based methods. As described in Phillips [49]:

- Residual-based methods estimate the discretization error using information about the discretization scheme and the current problem, including defect correction, error transport equations, and adjoint methods (Ainsworth et al. [50]). Compared to recovery methods, residual-based methods are tremendously more code intrusive.

- Recovery methods are based on the post-processing of a solution to approximate the discretization error. An example of a recovery method is the grid refinement, which uses refined meshes to estimate the discretization errors. The quality of such an error estimate is highly dependent on the solution quality. Solutions should be in the asymptotic range to conserve error estimation quality, which is impossible to achieve in practical applications. Therefore, the error estimation term has been changed to uncertainty estimation to bound the exact solution to the PDEs with some +/- band around the numerical solution (Phillips [49] and Phillips et al. [51]).

The methods used for Solution verification can be classified into four main categories (Gant [52]):

- Rules of thumb,
- Techniques based on prior RANS results,
- Single-grid estimators,
- Multi-grid estimators

Table 2.4 represents solution verification (i.e discretization error estimate) development over time. The information in Table 2.4 are mainly extracted from Oberkampf et al. [8], Roache [11], Eça et al. [12], Phillips [49], Rider et al. [53], Roache [54], Richardson [55], Roy et al. [56, 57], and Xing [58]. Each of the methods introduced in Table 2.4 is briefly explained in subsequent sections.

2.2.1.1 Rules of thumb for plumes

Rules of thumb are some rules developed for appraising appropriate grid resolution in modeling fire plumes prior to undertaking any LES simulations. The rules are provided in the manual for the Fire Dynamics Simulator *(FDS)* (Gant [52] and McGrattan et al. [59]). The advantage of this approach is having a reference while designing the computational grid. This approach lacks robustness, where it is highly dependent on the simulated problem and does not directly account for discretization errors. Further discussion on these rules can be found in Gant [52] and Smardz [60].

2.2.1.2 Turbulence length scales and cell size

Turbulence is usually defined using two characteristic length scales:

- The Kolmogorov length scale (η) which relates to the smallest dissipative turbulent eddies,
- The integral length scale (L) which relates to the largest eddies

In the premises of this approach, the turbulence length scales are used in the grid design for eddy-resolving simulations. The estimation of the turbulence length scales can be achieved by running a steady RANS simulation. The work of Addad et al. [61] and Baggett et al. [62] based the grid design on this approach. The cell size was two to five times smaller than

1911		*The Richardson Extrapolation was first proposed and applied*	Multi-Grid Estimators
	1994	Grid Convergence Index *(GCI)* introduced	
	2001	Correction Factor Method *(CF)* introduced	
	2002	Global Averaging Method introduced	
	2003	The Method of Nearby problem introduced	
	2004	CF method modified	
	2005	LES Index of Quality introduced *(LES_IQ)*	
		Grid and Model Variation *(GMV)* introduced	
	2009	The Factor of Safety method *(SF)* introduced	
	2010	SF method modified	
	2011	The Method of Nearby problem improved	
	2014	The Least-Squares Fits *(LSGCI)* method introduced	
	2015	Grid and Model Variation modified	
	2016	Robust Verification introduced	
1997		*Turbulence length scales and cell size*	Single-Grid Estimators
2002		*The Rules of thumb method is first used for Fire Dynamics simulations*	
		Sub-grid Activity parameter introduced	
2005		*Magnitude of sub-grid scale viscosity*	
		Turbulent Kinetic Energy Resolution introduced	
2009		*Relative Effective Viscosity Index*	

Table 2.4: Brief chronological development of solution verification approaches

the turbulence length scale (l_m). The turbulence length scale is calculated from the RANS model as follows:

$$l_m = c_\mu^{3/4} \frac{k^{3/2}}{\epsilon} \tag{2.4}$$

where:

$c_\mu = 0.09$
k : turbulent kinetic energy
ϵ : dissipation rate

The usage of the above approaches to assess the LES grid size based on prior RANS simulations is also implied in some best practice guidelines, such as best practice guidelines for the Use of CFD in nuclear reactor safety (Mahaffy et al. [63]). It notes: "As a general rule of thumb either the code logic or the user should ensure that four to five mesh cells are available to span (in each direction) the smallest eddy resolved by the Navier-Stokes solution." The main advantage of using this approach is that it is easy and cheap before running LES simulations. There are several drawbacks in using this approach:

1. the uncertainty associated with finding the length scales from a RANS simulation
2. it does not directly account for discretization errors
3. some of the ratios and values used in this approach are user-dependent and do not have sufficient scientific proof

Further discussion on this approach can be found in Gant [52], Addad et al. [61], Baggett et al. [62], Gant [64], and Pope [65].

2.2.1.3 Single Grid Estimators

Magnitude of the Subgrid-Scale Viscosity The magnitude of the subgrid scale viscosity can be expressed as the ratio between the turbulent viscosity to the molecular viscosity ($\frac{\nu_t}{\nu}$). Celik et al. [66] suggested that as a rough rule of thumb the ratio of effective viscosity to the molecular viscosity ($\frac{\nu_{eff}}{\nu}$) is approximately 20 for good LES and 1.0 for DNS, where (ν_{eff}) is the sum of the turbulent, numerical and molecular viscosities ($\nu_{eff} = \nu_t + \nu_{num} + \nu$). The numerical viscosity is estimated from the turbulent viscosity and the relation is dependent on the Reynolds number. Celik et al. [67] suggested for a high turbulence Reynolds number $Re_t = \frac{k^2}{\nu\epsilon}$ a ratio of ($\frac{\nu_{eff}}{\nu} \approx 20$), while for $Re_t = 300$ a ratio of ($\frac{\nu_{eff}}{\nu} \approx 5$). Moreover, Klein [68] and Hadžiabdić et al. [69] simulated flows at 4000 and 20000 Reynolds number with reasonably good predictions by keeping ($\frac{\nu_t}{\nu} \leq 3$) The great advantage of this method is that it requires only one simulation. Meanwhile, it has several defects, such as ratios being dependent on Reynolds number, some of the quantities being estimated not computed, which adds doubts to the credibility of the method, and not accounting for the numerical dissipation.

Relative Effective Viscosity Index The Relative Effective Viscosity Index ($LES-IQ_\nu$) is a derived measure from the approach of measuring the Subgrid-Scale Viscosity. According to Celik et al. [70], the ($LES-IQ_\nu$) is calculated as:

$$LES-IQ_\nu = \frac{1}{1+\alpha_\nu(\frac{\nu_{eff}}{\nu})^n} \tag{2.5}$$

where the constants $\alpha_\nu = 0.05$ and $n = 0.53$ are determined from studying homogeneous isotropic turbulence. According to Celik et al. [70], having $LES-IQ_\nu \geq 0.8$ is proposed as a good LES and a value greater than 0.95 as a DNS. There is no actual application of this index in the literature, but only the theory is presented in Gant [52], Celik et al. [67], and Celik et al. [70].

Subgrid Activity Parameter In this estimate, the quality of LES is assumed to be firmly defined by the subgrid parameterization and the numerical contamination of the smaller retained flow structures. Consequently, the subgrid-activity parameter (s) represents the total simulation error. The parameter measures the relative turbulent dissipation rate ($0 \leq s \leq 1$) and the subgrid resolution (r) (Geurts et al. [71]). The subgrid-activity parameter is calculated as following:

$$s = \frac{\overline{\epsilon_t}}{\overline{\epsilon_t}+\overline{\epsilon_\mu}} \tag{2.6}$$

where:

$\epsilon_\mu = \nu S_{ij}\frac{\partial U_i}{\partial x_j} \rightarrow$ *the molecular dissipation*

$\epsilon_t = -\tau_{ij}\frac{\partial U_i}{\partial x_j} \rightarrow$ *the turbulent dissipation*

τ_{ij} : sub-grid scale stress

$\frac{\partial U_i}{\partial x_j}$: velocity gradient

S_{ij} : strain rate

ν : kinematic viscosity

Following Geurts et al. [71] and Brandt [72], the subgrid activity parameter can be estimated using an equation simplified from Equation (2.6) without any assumptions:

$$s = \frac{\overline{\nu_t S^2}}{\overline{(\nu + \nu_t) S^2}} \tag{2.7}$$

Moreover, Celik et al. [66] proposed a simplified approximation for the subgrid activity parameter as:

$$s = \frac{\nu_t}{\nu + \nu_t} \tag{2.8}$$

This technique shows the importance of sub-grid scales in the flow and requires one LES. On the other hand, it requires the calculation of additional time-averaged parameters, is insensitive to grid resolution, and depends on Reynolds number.

Turbulent Kinetic Energy Resolution The resolution of the turbulent kinetic energy is one of the decisive elements in judging the quality of an eddy-resolving simulation. The total turbulent kinetic energy can be decomposed into three components (Pope [65, 73], Garcia et al. [74], and Gant [75]):

$$k_{tot} = k_{res} + k_{sgs} + k_{num} \tag{2.9}$$

k_{res} is the turbulent kinetic energy from resolved velocity fluctuations and can be computed as follows:

$$k_{res} = \frac{1}{2}\left(\overline{\mathrm{u}^2} + \overline{\mathrm{v}^2} + \overline{\mathrm{w}^2}\right) \tag{2.10}$$

k_{sgs} is the component from the sub-grid scale turbulence model. The doubts in this approach come from calculating this parameter because there is no general agreement on how to compute it. Gant [75] examined three different proposals to compute this quantity:

$$k_{sgs} = \frac{3}{2}\left(\frac{\nu_t}{\sqrt[3]{Vol}}\right)^2 \tag{2.11}$$

$$k_{sgs} = 3.6\left(\frac{\nu_t S^2 (Vol)^{\frac{1}{3}}}{\pi}\right)^{\frac{2}{3}} \tag{2.12}$$

$$k_{sgs} = \frac{l^2 S^2}{C_E} \tag{2.13}$$

where:

Vol : cell volume
S : strain invariant
ν_t : subgrid-scale eddy viscosity
C_E : constant stress-energy ratio assumed ≈ 0.3
l : Smagorinsky length scale $l = C_s\delta$
C_s Smagorinsky constant

Further discussion and simplification of these three definitions can be found in Gant [52], Gant [64], Pope [65], and Gant [75]. Two further developments can be found in the work of Klein [68] and Lilly [76]. All these different approaches were developed to estimate the fraction of the resolved-to-modeled turbulent kinetic energy from:

$$\frac{k_{res}}{k_{tot}} = \frac{k_{res}}{k_{res} + k_{sgs}} \tag{2.14}$$

According to Pope [65, 73] and Kempf et al. [77], at least 80% resolution of the turbulent kinetic energy is required to have an acceptable eddy-resolving simulation. The main advantage of this approach is the need for only one simulation. Meanwhile, it depends heavily on the assumptions in the calculation of the k_{sgs}, which makes it application-dependent. Moreover, Gant [52] stated two drawbacks of this approach in assessing LES quality. Firstly, it is implicitly assumed that as the grid is refined, the turbulent kinetic energy (TKE) resolved increases, which is not necessarily true. This assumption is not valid in some cases where the TKE decreases with refinement; thus, negative values are obtained. This assures that this approach is not a good indication of mesh resolution. Secondly, numerical dissipation is ignored.

2.2.1.4 LES Index of Quality Measure

This method can be a single-grid estimator or a multi-grid estimator, depending on the definition of the total turbulent kinetic energy calculation. The LES Index of Quality ($LES-IQ_k$) is developed based on the resolution of the turbulent kinetic energy, which is evolved from the single grid estimator approach explained in Section 2.2.1.3. The previous approach's challenge was the computation of the total turbulent kinetic energy, especially the sub-grid scale component. The ($LES-IQ_k$) proposed by Celik et al. [66] and Celik et al. [67] is calculated from:

$$LES-IQ_k = \frac{k_{res}}{k_{tot}} = \frac{k_{res}}{k_{res}+a_k h^p} \tag{2.15}$$

The error term in equation Equation (2.15) is a result of a Richardson extrapolation concept through performing two simulations on two different grid resolutions. The main purpose of the usage of Richardson extrapolation is the estimation of the total turbulent kinetic energy. Celik et al. [70] presented a different approach to predict the total turbulent kinetic energy from DNS or experiments instead of Richardson extrapolation. The necessary steps in order to calculate the basic ($LES-IQ_k$) are summarized in Gant [52] as follows:

1. Perform LES calculations using two separate grids: coarse and fine. Hereafter subscript “1” refers to fine mesh results and subscript “2” to coarse mesh results.

2. Calculate the resolved turbulent kinetic energy with both grids, i.e. find k_1^{res} and k_2^{res} , from Equation (2.10).

3. Calculate the characteristic grid cell size, h, for both grids. For structured meshes using:

$$h = \sqrt[3]{(\delta x \delta y \delta z)} \tag{2.16}$$

   where δx, δy and δz are the cell dimensions, and for unstructured meshes[2]:

$$h = \sqrt[3]{(Vol)} \tag{2.17}$$

[2] The cell size measure defined in Equation (2.17) is justifiable for grids with uniform mesh densities but not for meshes with local refinement.

where Vol is the cell volume.

4. Calculate the $(LES-IQ_k)$ on the fine grid from:

$$LES-IQ_k^f = \frac{1}{1+\left(1-\frac{k_2^{res}}{k_1^{res}}\right)\left[\frac{1}{(\alpha^p-1)}\right]} \tag{2.18}$$

and on the coarse grid from:

$$LES-IQ_k^c = \frac{1}{1+\left(\frac{k_1^{res}}{k_2^{res}}-1\right)\left[\frac{\alpha^p}{(\alpha^p-1)}\right]} \tag{2.19}$$

Reasonably good results were obtained in some flows tested by Gousseau et al. [7], Celik et al. [66], Celik et al. [67], and Klein [68]. Having $(LES-IQ_k) \geq 80\%$ means a good grid resolution. Some limitations were noticed by Klein [68] as this approach should not be used for laminar or transitional flows. Another drawback is having some negative values when having k_{res} larger than the DNS turbulent kinetic energy. Celik et al. [66] proposed a correction to $LES-IQ_k$ calculation as follows:

$$LES-IQ_k = 1-\frac{|k_{tot}-k_{res}|}{k_{tot}} \tag{2.20}$$

The main advantage of this procedure is using Richardson extrapolation, which takes into account the grid resolution. In contrast, reliability depends on the grid resolution being within the asymptotic range, which urges the need for an uncertainty measure.

2.2.1.5 Richardson Extrapolation Based Methods

All the methods presented in this section are multi-grid estimators. The Richardson extrapolation-based methods are the most widely used in discretization error estimation. Richardson extrapolation was first introduced in 1911 by Richardson [55]. The robustness of the method to any scientific computation with any discretization scheme (e.g., finite volume, finite difference or finite element) accelerated its success. Because it is an a-posteriori error estimator, the method is not code invasive and relatively easy to implement. The basic idea of Richardson extrapolation is knowing the formal rate of convergence with mesh refinement and discrete solutions on two systematically refined meshes (Richardson [55]). Then, Richardson extrapolation can be used to estimate the exact solution. In mathematical notation, having ϕ defined as the local or global solution variable on a mesh spacing h (Oberkampf et al. [4, 8] and Oberkampf et al. [78]),

$$\epsilon_h = \phi_h - \tilde{\phi} \tag{2.21}$$

where:

ϵ_h : is error at mesh spacing h
ϕ_h : discrete solution
$\tilde{\phi}$: exact solution

The numerical solution ϕ_h can be expanded using a Taylor series expansion about the exact solution,

$$\phi_h = \tilde{\phi} + \frac{\partial \tilde{\phi}}{\partial h} h + \frac{\partial^2 \tilde{\phi}}{\partial h^2} \frac{h^2}{2} + \frac{\partial^3 \tilde{\phi}}{\partial h^3} \frac{h^3}{6} + O(h^4), \tag{2.22}$$

or simply a power series in h,

$$\phi_h = \tilde{\phi} + \alpha_1 h + \alpha_2 h^2 + \alpha_3 h^3 + O(h^4), \tag{2.23}$$

$$\epsilon_h = \phi_h - \tilde{\phi} = \alpha_1 h + \alpha_2 h^2 + \alpha_3 h^3 + O(h^4), \tag{2.24}$$

Equation (2.24) is called the standard Richardson extrapolation. Richardson extrapolation can be generalized to $p^{th} - order$ accurate schemes as following:

$$\epsilon_h = \phi_h - \tilde{\phi} = \alpha_p h^p + \alpha_{p+1} h^{p+1} + \alpha_{p+2} h^{p+2} + ..., \tag{2.25}$$

Equation (2.25) can be written having computations on different grid levels with refinement factor $r = \frac{h_{coarse}}{h_{fine}} > 1$ as

$$\epsilon_h = \phi_h - \tilde{\phi} = \alpha_p h^p + \alpha_{p+1} h^{p+1} + o(h^{p+2}) \tag{2.26}$$

$$\epsilon_{rh} = \phi_{rh} - \tilde{\phi} = \alpha_p (rh)^p + \alpha_{p+1} (rh)^{p+1} + o(h^{p+2}) \tag{2.27}$$

Then the generalized Richardson extrapolation estimate $\overline{\phi}$ (Oberkampf et al. [8]):

$$\overline{\phi} = \phi_h + \frac{\phi_h - \phi_{rh}}{r^p - 1} \tag{2.28}$$

Finally, the discretization error can be simply estimated as:

$$\overline{\epsilon_h} = \overline{\phi} - \phi_h = \frac{\phi_h - \phi_{rh}}{r^p - 1} \tag{2.29}$$

The credibility of this error estimator is restricted by five assumptions (Oberkampf et al. [8]):

1. Discrete solution in the asymptotic range
2. A uniform spacing of meshes over the domain
3. Systematic mesh refinement
4. Smooth solutions
5. Other sources of numerical errors are minimal

In practical computations, it is not possible to satisfy all the restrictions imposed using the Richardson extrapolation. There are two components included in these approaches. The first is the discretization error estimation. The second is the discretization uncertainty estimation. The two components must be combined to account for the uncertainties originating from having meshes not in the asymptotic range, and definition of the (h) and (p) parameters. Consequently, the uncertainties associated with the violation of the Richardson assumptions are to be quantified. A discretization uncertainty estimator is required to increase the reliability and integrity of the Richardson extrapolation-based methods. Therefore, several approaches are introduced in the literature, such as Grid Convergence Index (GCI), Global Averaging Method, Correction Factor Method (CF), Safety Factor Method (SF), Least-Squares Method (LSR) and Grid and Model Variation (GMV). Finally, all the methods introduced in the literature are variants of the GCI.

The Grid Convergence Index (GCI) In 1994, Roache proposed the grid convergence index (GCI) as the first error estimator based on Richardson extrapolation with uncertainty. The GCI can basically be described as the absolute value of Richardson extrapolation error estimate multiplied by a safety factor of three (Oberkampf et al. [8], Roache [11], Phillips [49], and Phillips et al. [51]). The GCI was developed as a means for uniform reporting of grid convergence studies having solutions in the non-asymptotic range. The GCI for the fine grid numerical solution is defined as

$$GCI = \frac{F_s}{r^p - 1}\left|\frac{\phi_2 - \phi_1}{\phi_1}\right| \tag{2.30}$$

The GCI is modified as follows in most recent implementations to avoid problems when the solution values are near zero:

$$GCI = \frac{F_s}{r^p - 1}\left|\phi_2 - \phi_1\right| \tag{2.31}$$

where:

$F_s = 3$ factor of safety when having two grids solutions only
r : refinement ratio
p : order of convergence

The order of convergence (p) in Equation (2.31) is assumed to be equal to the formal order of convergence ($p = p_f = 2$). Roache proposed some modifications on the definition of F_s when having solutions on more than two grids and the observed order of convergence ($\tilde{p}$) is near the formal order of convergence (p_f). The F_s and p in Equation (2.31) are defined according to Table 2.5.

$\left\lvert\frac{\tilde{p}-p_f}{p_f}\right\rvert$	F_s	P
≤ 0.1	1.25	p_f
> 0.1	3.0	$min\left(max\left(0.5, \tilde{p}\right), p_f\right)$

Table 2.5: GCI factor of safety values (Oberkampf et al. [8])

The procedure of computing $\tilde{p}$ is based on the Richardson extrapolation. Let's assume a $p^{th} - order$ accurate scheme with numerical solutions on three meshes, fine (h_1), medium (h_2) and coarse (h_3), and a constant grid refinement ratio (r),

$$r = \frac{h_2}{h_1} = \frac{h_3}{h_2} > 1, \tag{2.32}$$

thus we can write

$$h_1 = h, \quad h_2 = rh, \quad h_3 = r^2 h \tag{2.33}$$

Using the generalized Richardson discretization error expansion from Equation (2.26),

$$\phi_1 = \tilde{\phi} + \alpha_p h^p + \alpha_{p+1} h^{p+1} + o(h^{p+2}), \tag{2.34}$$

$$\phi_2 = \tilde{\phi} + \alpha_p (rh)^p + \alpha_{p+1} (rh)^{p+1} + o(h^{p+2}), \tag{2.35}$$

$$\phi_3 = \tilde{\phi} + \alpha_p (r^2 h)^p + \alpha_{p+1} (r^2 h)^{p+1} + o(h^{p+2}), \tag{2.36}$$

The observed order of convergence ($\tilde{p}$) is calculated from Equations 2.34, 2.35, and 2.36 by neglecting high order terms and set $p = \tilde{p}$. Then, we can subtract ϕ_2 from ϕ_3 and ϕ_1 from ϕ_2. One can write

$$\phi_3 - \phi_2 = \alpha_{\tilde{p}} (r^2 h)^{\tilde{p}} - \alpha_{\tilde{p}} (rh)^{\tilde{p}} = \alpha_{\tilde{p}} r^{\tilde{p}} h^{\tilde{p}} \left(r^{\tilde{p}} - 1 \right), \tag{2.37}$$

$$\phi_2 - \phi_1 = \alpha_{\tilde{p}} (r^h)^{\tilde{p}} - \alpha_{\tilde{p}} h^{\tilde{p}} = \alpha_{\tilde{p}} h^{\tilde{p}} \left(r^{\tilde{p}} - 1 \right), \tag{2.38}$$

Dividing Equation (2.37) by Equation (2.38) and taking the natural log of both sides, the observed order of convergence can be defined as

$$\tilde{p} = \frac{ln\left(\frac{\phi_3 - \phi_2}{\phi_2 - \phi_1}\right)}{ln(r)} \tag{2.39}$$

Further discussion on this approach can be found in Oberkampf et al. [8], Roache [11], Phillips [49], and Gant [52].

Global Averaging Method This method is based on the GCI. Cadafalch et al. [79] proposed this method.Only a factor of safety $F_s = 1.25$ is used in the evaluation of the GCI using Equation (2.31). The method is summarized in Oberkampf et al. [8] as:

1. Interpolate the solutions from three systematically refined meshes onto a common post-processing grid with a higher-order interpolation method.

2. Classify the nodes in the common mesh as

$$\textit{Richardson nodes}: \quad \left(\phi_3 - \phi_2\right)\left(\phi_2 - \phi_1\right) > 0 \tag{2.40}$$

$$\textit{Oscillatory nodes}: \quad \left(\phi_3 - \phi_2\right)\left(\phi_2 - \phi_1\right) < 0 \tag{2.41}$$

3. Compute the local observed order of convergence $\tilde{p}$ for all of the Richardson nodes according to Equation (2.39).

4. Compute a global observed order of accuracy from step 3 as

$$\tilde{p} = \frac{1}{N} \sum_{i=1}^{N} min\left(max\left(0.5, \tilde{p}_i\right), p_f\right) \tag{2.42}$$

5. Compute the local GCI values at the Richardson nodes with the global order of convergence from step 4.

Further discussion can be found in Oberkampf et al. [8], Phillips [49], Phillips et al. [51], and Cadafalch et al. [79].

Correction Factor Method (CF) and Safety Factor Method (SF) The correction factor method (CF) and safety factor method (SF) are similar in that both are functions of the observed order of accuracy ($\tilde{p}$). The CF was initially proposed in Wilson et al. [80]. After the criticism from Roache [54], the method was modified in Wilson et al. [81] as

$$\overline{\epsilon}_{CF} = \frac{\phi_2 - \phi_1}{r^{\tilde{p}} - 1} \tag{2.43}$$

$$U_{CF} = \begin{cases} \left[9.6(1-CF)^2+1.1\right]\left|\overline{\epsilon}\right| & \text{if } 0.875 < CF < 1.125 \\ \left[2\left|1-CF\right|+1\right]\left|\overline{\epsilon}\right| & \text{if } 0 < CF \leq 0.875 \text{ and } CF \geq 1.125 \end{cases} \tag{2.44}$$

where the correction factor is computed as

$$CF = \frac{r^{\tilde{p}}-1}{r^{p_f}-1} \tag{2.45}$$

Moreover, the factor of safety method was proposed in Xing et al. [82]. Later, the method was adapted by Xing Tao [83] to remove the singularity at twice the formal order of convergence by adding a third coefficient. The FS method is defined as

$$\overline{\epsilon}_{FS} = \frac{\phi_2-\phi_1}{r^p-1} \tag{2.46}$$

$$U_{FS} = \begin{cases} \left[FS_1 p + FS_0\left(1-p\right)\right]\left|\overline{\epsilon}_{FS}\right| & \text{if } 0 < p \leq 1 \\ \left[FS_1 p + FS_2\left(1-p\right)\right]\left|\overline{\epsilon}_{FS}\right| & \text{if } p > 1 \end{cases} \tag{2.47}$$

where: $FS_0 = 2.45$, $FS_1 = 1.6$, $FS_2 = 14.8$ and $p = \frac{\tilde{p}}{p_f}$

The Least-Squares Method (LSR1) This approach was developed by Eça and Hoekstra in 2002 to estimate numerical uncertainties in solutions not in the asymptotic range. The central concept of this method is filtering out the noise from the observed order of accuracy calculation using the least-squares fitting approach over four or more mesh levels (Oberkampf et al. [8]). The error function is defined based on the series expansion from the generalized Richardson extrapolation defined in Equation (2.25). The method had several developments, as shown in the literature (Eça et al. [46, 84, 85]), but the final version was introduced in Eça et al. [12]. The procedure is summarized in Figure 2.2.

The method can be divided into two main parts (Eça et al. [12]):

1. The first part is the error estimation. Power series expansions are used for error estimation. Power series expansions are defined as a function of the typical cell size, further are fitted to the simulations data using least-squares regression analysis. The procedure includes several formulations, such as weighted and non-weighted fits for expressions with different exponents in the series's leading term. Finally, the selection of the best error estimate is based on the standard deviation of the fits.
2. The second part is the uncertainty estimation. The error estimation is multiplied by a safety factor to account for data convergence behavior. The safety factor equals 1.25 for monotonic data convergence and a fit standard deviation smaller than an averaged data range parameter (reliable error estimation). Otherwise, the safety factor is three.

Therefore, the method is said to be an error estimation with a measure of uncertainty. Further discussion on this method and testing can be found in Pereira et al. [17], Xing et al. [86], Eça et al. [87], Negrato et al. [88], Pereira et al. [89], Vink et al. [90], and Tolias et al. [91], which showed adequate performance in predicting numerical discretization uncertainty.

The Grid and Model Variation (GMV) The GMV method was first proposed by Klein [68]. It is based on the Richardson extrapolation. In this approach, not only are grid effects considered, but the sub-grid scale model is also varied. The error is defined as

$$\phi_\Delta - \tilde{\phi} = c_n h^n + c_m \Delta^m + o\left(h^{n+1}, \Delta_s^{m+1}\right) \tag{2.48}$$

where:

$n = 2$ order of numerical scheme
m : order of modeling error
h : grid size
Δ : filter width
ϕ_Δ : mean LES solution
$c_n h^n$: numerical error
$c_m \Delta^m$: modeling error

In Klein's approach, two assumptions are made: high order error terms are ignored, and second-order accurate schemes are used so that $n = m = 2$.

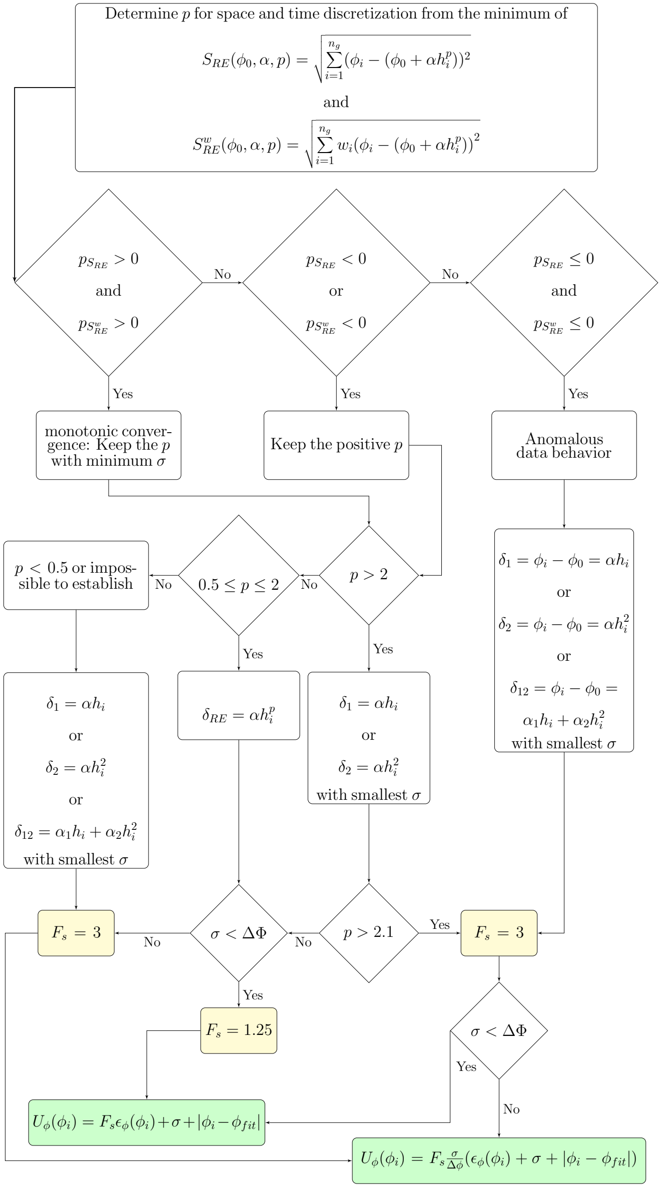

Determine p for space and time discretization from the minimum of
$S_{RE}(\phi_0, \alpha, p) = \sqrt{\sum_{i=1}^{n_g} (\phi_i - (\phi_0 + \alpha h_i^p))^2}$
and
$S_{RE}^w(\phi_0, \alpha, p) = \sqrt{\sum_{i=1}^{n_g} w_i (\phi_i - (\phi_0 + \alpha h_i^p))^2}$
$p_{S_{RE}} > 0$ and $p_{S_{RE}^w} > 0$
No
$p_{S_{RE}} < 0$ or $p_{S_{RE}^w} < 0$
No
$p_{S_{RE}} \leq 0$ and $p_{S_{RE}^w} \leq 0$
Yes
Yes
Yes
monotonic convergence: Keep the p with minimum σ
Keep the positive p
Anomalous data behavior
$p < 0.5$ or impossible to establish
No
$0.5 \leq p \leq 2$
No
$p > 2$
$\delta_1 = \phi_i - \phi_0 = \alpha h_i$ or $\delta_2 = \phi_i - \phi_0 = \alpha h_i^2$ or $\delta_{12} = \phi_i - \phi_0 = \alpha_1 h_i + \alpha_2 h_i^2$ with smallest σ
Yes
Yes
$\delta_1 = \alpha h_i$ or $\delta_2 = \alpha h_i^2$ or $\delta_{12} = \alpha_1 h_i + \alpha_2 h_i^2$ with smallest σ
$\delta_{RE} = \alpha h_i^p$
$\delta_1 = \alpha h_i$ or $\delta_2 = \alpha h_i^2$ with smallest σ
$F_s = 3$
No
$\sigma < \Delta\Phi$
No
$p > 2.1$
Yes
$F_s = 3$
Yes
$F_s = 1.25$
$\sigma < \Delta\Phi$
Yes
No
$U_\phi(\phi_i) = F_s \epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|$
$U_\phi(\phi_i) = F_s \frac{\sigma}{\Delta\phi}(\epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|)$

Figure 2.2: Flowchart for the solution verification approach for the numerical uncertainty estimation following Eça et al. [12]

Based on these assumptions, only three simulations are needed to estimate the numerical discretization error. Further explanation can be found in Klein [68] and test cases can be found in Gousseau et al. [7] and Brandt [72]. The use of Richardson-based extrapolation in predicting LES quality is not straightforward and needs further developments. Consequently, more work had been done in that direction (Xing [58]) and tested in Dutta et al. [92, 93]. Finally, the drawback of the methods proposed by Xing [58] is the need for at least five simulations, and it requires some control over the filter width, which is not always affordable in commercial CFD codes.

2.2.1.6 The Method of Nearby Problems (MNP)

The MNP is developed by Roy et al. [94] to estimate the discretization error using only one grid. The MNP is a form of defect correction that is examined to generate exact solutions to partial differential equations and as a discretization error estimator. The procedures can be summarized as in Roy et al. [94]:

1. Establish accurate numerical solution
2. Generate analytic curve fit to solution from step 1
3. Generate analytic source terms
4. Numerically solve “nearby” problem
5. Evaluate error estimators

Further explanation and test cases of the MNP can be found in Roy et al. [56, 57], Roy et al. [94], and Roy et al. [95].

2.2.2 Summary of Solution Verification Methods

In this section, all previously discussed methods are summarized in Table 2.6, including the main advantages and restrictions on applying these methods for eddy-resolved simulations *(ERS)*.

Description	Restrictions on Application to Eddy-Resolving Simulations	Advantages
	Turbulence Length Scales and Cell Size	
Design of the grid based on prior RANS simulations to estimate both Kolmogorov and integral length scales	• Restriction on the quality of RANS prediction, • Does not account for discretization error, • Some of the parameters used in the calculations are expert defined quantities, • Can not be used in transitional or laminar regions.	• Cheap and easy to compute, • Grid is generated before running ERS.
	Magnitude of the Subgrid-Scale Viscosity	
It is a single grid estimator. It is the ratio between the turbulent to the molecular viscosity	• Reynolds number dependent, • Does account for effects of numerical dissipation, • Does not account for discretization errors.	• Easy to compute, • Needs only one ERS.
	Relative Effective Viscosity Index	
It is a single grid estimator and derived from the Magnitude of the Subgrid-Scale Viscosity	• Few examples of its application in the literature because of the effective viscosity definition, • Assumes effect of numerical dissipation is the same as modeled dissipation, • Does not account for discretization errors.	• Provides initial estimate of quality based on performing one ERS.
	Subgrid Activity Parameter	
It is defined as the ratio between turbulent to the total dissipation	• Requires calculation of additional time-averaged parameters, • Depends on Reynolds number, • Relatively insensitive to grid resolution, actually does not account for discretization error.	• Shows importance of subgrid-scales in the flow, • Requires one ERS.

Description	Restrictions on Application to Eddy-Resolving Simulations	Advantages
	Turbulent Kinetic Energy Resolution	
It is the ratio between the resolved to total turbulent kinetic energy	• Requires calculation of additional time-averaged parameters, • Assumption having total turbulent kinetic energy is greater than the resolved is not always valid, • Does not account for discretization errors.	• Indicates the proportion of the resolved turbulent kinetic energy, • Needs only one ERS.
	LES Index of Quality	
It is the ratio between the resolved to total turbulent kinetic energy based on multi-grid simulations and Richardson extrapolation	• Assumes second order numerical error, • Assumption that the total turbulent kinetic energy is greater than resolved is not valid in some cases, • Reliability depends on grid resolution being within the asymptotic range.	• Uses Richardson extrapolation, • Reasonably good accuracy reported in some flows.
	The Grid Convergence Index	
Richardson extrapolation based method which Estimates the discretization uncertainty with a safety factor	• Assumes second order of convergence and refinement factor, • Not handling variations between observed and formal order of convergence, • Issue of non-monotone convergence.	• Accounts for the uncertainty associated with non-asymptotic solutions, • Requires three grids.
	Global Averaging Method	
Richardson extrapolation based which Estimates the discretization uncertainty by averaging monotonically converging nodes	• Requires more computations, • Is not valid in transitional and turbulent flows, • Inconsistency in the sense that oscillatory and boundary data points are excluded.	• Accounts for non-asymptotic solutions, • Less conservative than GCI.

Description	Restrictions on Application to Eddy-Resolving Simulations	Advantages
	Correction Factor and Safety Factor Methods	
Depend on Richardson Extrapolation with factors depending on observed order of convergence	• Treatment of oscillatory, lower and upper limits for the convergence order, • Noisy data in calculating the observed order of convergence.	• Accounts for non-asymptotic solutions and discretization errors.
	The Least-Squares Fitting Method	
Using least-squares fitting approach to filter out the noise from the observed order of convergence	• Treatment of non-monotonically converging data.	• Accounts for non-asymptotic solutions and discretization errors.
	The Grid and Model Variation	
Based on Richardson extrapolation. Separate modeling and discretization errors	• Requires at least five simulations, • Needs control on the filter width.	• Quantification for both modeling and discretization errors separately.
	The Method of Nearby Problems	
Estimation of the discretization error using one grid	• Very complex in dealing with 3D-unsteady simulations.	• Requires one simulation.

Table 2.6: Summary for different techniques used in the context of solution verification to quantify simulation quality

2.3 Summary

Overall, it can be clearly noted from Table 2.1 that the available standards are not developed for CWE simulations, but some of the presented methods can be adapted for CWE. Firstly, the MMS can be used for code verification. The MMS has shown to be a very efficient approach for detecting implementation mistakes since it has been applied to many codes, as shown in Table 2.3. Secondly, the LSR1 is shown to be a very promising numerical uncertainty estimator for solution verification, see Table 2.6. The LSR1 has been used for solution verification for several applications in the literature. The method uses a safety factor to account for uncertainty in the error estimation. Moreover, the LSR1 can be defined as the least-squares fitting for power series expansions as a function of the typical cell size.

The frequently heard argument 'any solution is better than none' can be dangerous in the extreme. The greatest disaster one can encounter in computation is not instability or lack of convergence but results that are simultaneously good enough to be believable but bad enough to cause trouble.

Joel H. Ferziger

CHAPTER 3

Proposed Simulation Credibility Assessment Methodology

In this chapter, the newly developed simulation credibility assessment framework is introduced. As presented in Chapter 2, verification activities are the basis for *SQA*. In this work context, verification activities are performed for unsteady turbulent simulations with high Re. Moreover, natural wind conditions are used, which means turbulent inlet and eddy-resolving simulations are used. All the above properties describe a *Computational Wind Engineering (CWE)* simulation. The challenges imposed in CWE reliability are summarized as follows

- Unsteady flow,
- Simulating natural wind conditions,
- Transient inlet,
- High Re,

- Limited computational power,
- Using large grid sizes,
- Discrete solutions are not in the asymptotic range,
- Measuring integral quantities (i.e drag, lift, and crosswind forces) and local quantities (i.e surface pressure and velocities at reference height),
- Scatter in the data and anomalous behavior[1],
- Non-matching observed and formal order of convergence,
- Using unstructured grids,
- Using non-uniform refinement,
- Eddy-resolving simulations.

Therefore, a special emphasis is needed for each component of the verification procedures. The mainstream of the new credibility methodology is shown in Figure 3.1. In the subsequent sections, the main blocks are explained in detail.

The procedure preliminaries are presented in Section 3.1. The two main blocks ***(1) and (2)*** in Figure 3.1 are presented in Section 3.2 and Section 3.3 respectively. The proposed credibility methodology defines a variety of code verification test cases, some fulfill the governing equations, and some are for code debugging. Meanwhile, for uncertainty estimation, particular emphasis is placed on defining an appropriate measure for typical grid size (h) for unstructured non-uniform meshes used in practice. Furthermore, the safety factor (F_s) is defined based on the uncertainty in the identification of (h). Finally, statistical convergence error is quantified, which is rarely considered in publications on verification activities.

[1] Anomalous data behavior occurs if the observed order of convergence $(\tilde{p})$ is less than or equal to zero.

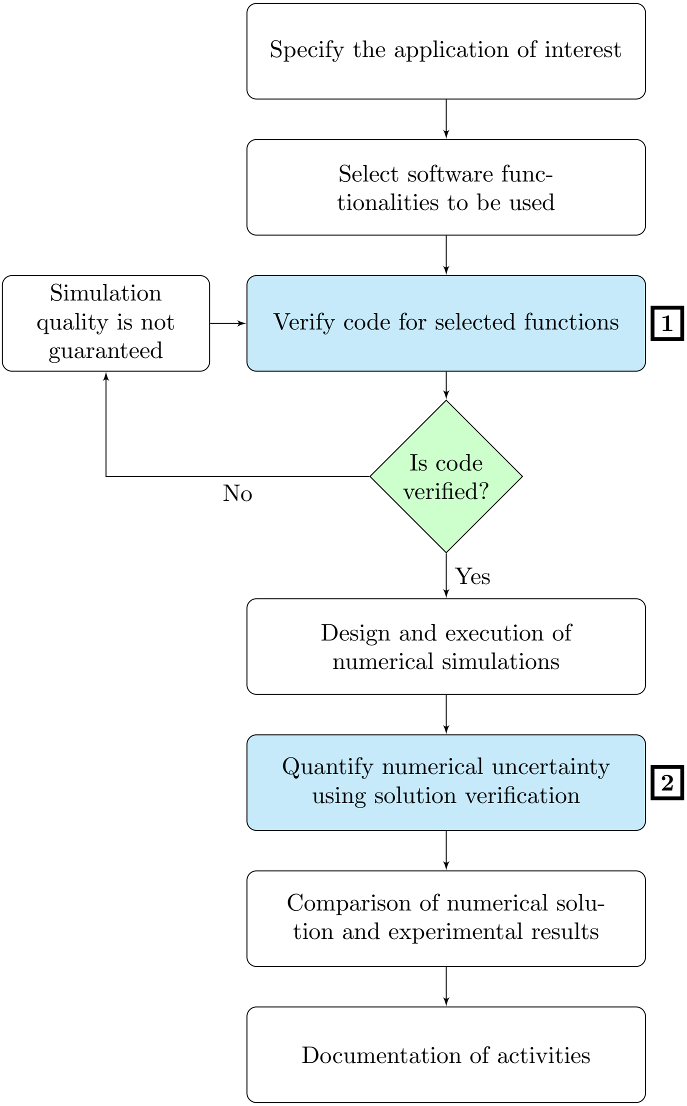

Specify the application of interest
Select software functionalities to be used
Simulation quality is not guaranteed
Verify code for selected functions
1
Is code verified?
No
Yes
Design and execution of numerical simulations
Quantify numerical uncertainty using solution verification
2
Comparison of numerical solution and experimental results
Documentation of activities

Figure 3.1: Flowchart for the verification procedures main building blocks

3.1 Procedure Preliminaries

Specify the application of interest It is crucial to specify the application target to identify all the challenges and limitations of the simulation. In the context of this work, CWE simulations are used for calculating loads on tall-slender structures (i.e., high-rise buildings).

Select software functionalities to be used CFD application of KRATOS Multiphysics [2] is used as test code for clarification of the method applicability. The variational multi-scale element is considered using a monolithic solver. More information about the simulation code will be presented in Chapter 4.

Verify code for selected functions This part is concerned with examining the correctness of the code implementation. The required steps are presented in Section 3.2. The newly developed benchmarks and code verification procedures are presented in Chapter 5. Once the code is verified and passed all the designed benchmarks, physical problems can be solved, and numerical errors can be estimated.

Simulation quality is not guaranteed If, after performing the code verification procedures, the acceptance criteria are not satisfied, verification can not be completed because the code has some bugs, and code credibility can not be established.

Design and execution of numerical simulations After building confidence in the code according to the method explained in Section 3.2, simulations for real test cases can be started. Then, numerical uncertainties associated with the simulation can be evaluated according to the procedure explained in Section 3.3.

Quantify numerical uncertainty using solution verification Now, the numerical uncertainty for the application of interest is to be quantified.

[2] http://www.cimne.com/kratos/
https://github.com/KratosMultiphysics
More information can be found in Dadvand [96] and Cotela [97]

Achieving this target requires a unique treatment of the challenges imposed in CWE simulations. Thus, a new framework with new definitions is introduced in Section 3.3 to quantify numerical uncertainty, especially uncertainty due to discretization and statistical convergence.

Comparison of numerical solution and experimental results This can be considered the final step. The discrete solution with the estimated uncertainty can be compared to experimental results if existing.

3.2 Code Verification Assessment Framework

In this section, block ***(1)*** of the verification procedure is presented in Figure 3.2. It is very similar to the procedure shown in the literature Section 2.1.2.1. The main difference is the design of the code verification test cases. A combination of MES and MMS is used in code verification to initiate a test case for codes with no access to the equations system. It is very important to indicate that all the recommendations presented in Section 2.1.2 must be realized in the design of test cases. Recommendations presented in Section 2.1.2.3 are followed for the decisions on initial and boundary conditions. Chapter 5 presents an application of the code verification procedures, where test cases design is shown, and instructions on how to perform refinement are presented to verify a simulation code.

3.2.1 Calculation of the Global Error

The numerical solution is composed of the values of the variables dependent on discrete locations obtained from spatial and temporal discretization algorithms. The errors in computer simulations are physical modeling errors, discretization errors, incomplete iterative convergence error (IICE), programming mistakes, and computer round-off errors (Oberkampf et al. [8] and Fisch [9]). In this work scope, the discretization error is dominating since all other errors are kept minimal. Several measures are possible to compute the discretization error. In the MMS, the exact solution is defined on the continuum. Thus, the exact solution can be computed at the same locations in time and space as the numerical solution (Knupp et al. [31]). Let n be a grid point, $\hat{\phi}_n$ be the analytical solution, ϕ_n be the discrete so-

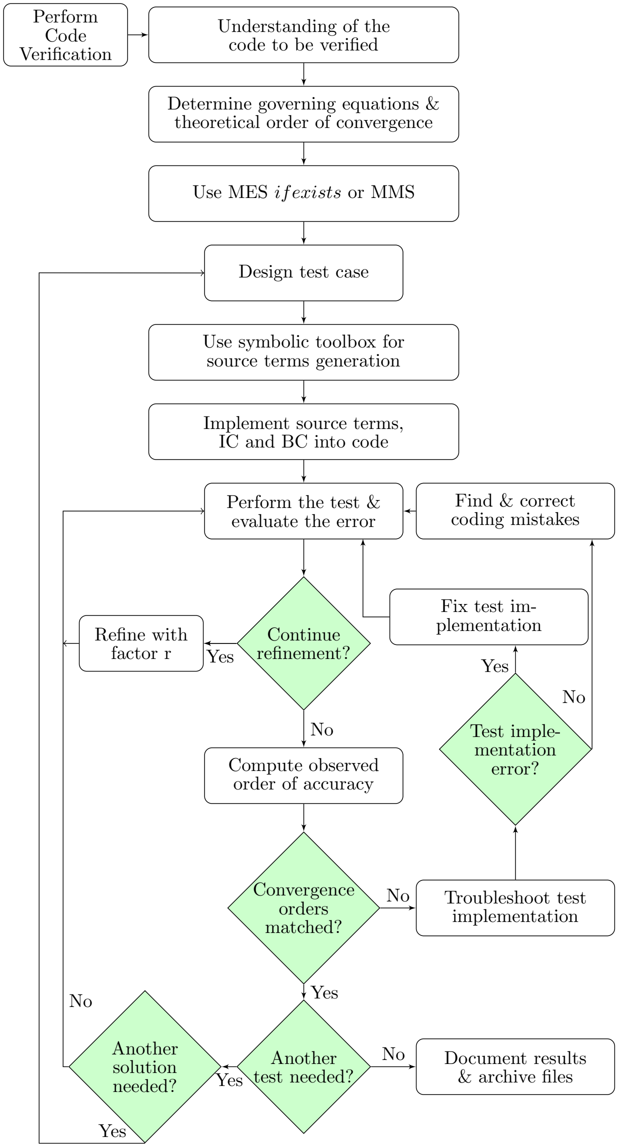

Perform Code Verification
Understanding of the code to be verified
Determine governing equations & theoretical order of convergence
Use MES $if exists$ or MMS
Design test case
Use symbolic toolbox for source terms generation
Implement source terms, IC and BC into code
Perform the test & evaluate the error
Find & correct coding mistakes
Fix test implementation
Refine with factor r
Yes
Continue refinement?
Yes
No
Test implementation error?
No
Compute observed order of accuracy
Convergence orders matched?
No
Troubleshoot test implementation
Yes
No
Another solution needed?
Yes
Another test needed?
No
Document results & archive files
Yes

Figure 3.2: Flowchart for the code verification procedures adapted from Oberkampf et al. [8], Roache [11], and Knupp et al. [31]

lution. Both solutions are evaluated at the same point in time and space $\left(x_n, y_n, z_n, t\right)$. The normalized global error is evaluated on the L_2 norm by

$$E_{L_2} = \sqrt{\frac{\sum_n \left(\hat{\phi}_n - \phi_n\right)^2 \alpha_n}{\sum_n \alpha_n}} \tag{3.1}$$

where α_n is some local volume measure[3]. Moreover, if this local measure is constant, such as in a uniform grid, the normalized global error (sometimes referred to as the RMS error) reduces to

$$E_{L_2} = \sqrt{\frac{1}{N} \sum_n \left(\hat{\phi}_n - \phi_n\right)^2} \tag{3.2}$$

where N is the number of grid points. Another very sensitive norm, that can be used to obtain the global error, is the infinity norm which is defined by

$$E_{\inf} = m\underset{N}{a}x |\hat{\phi}_n - \phi_n| \tag{3.3}$$

It is important to include all the grid points especially near boundary points in calculating the global error using either the RMS or infinity norms (Knupp et al. [31]).

3.2.2 Order of Convergence

When performing code verification with the order of convergence as the acceptance criteria, it is crucial to differentiate between formal and observed order of convergence.

3.2.2.1 Formal Order of Convergence

The formal order of convergence is also known as the overall theoretical order of accuracy of the fundamental discretization scheme used by the code (Knupp et al. [31]). Sometimes, it is not easy to retrieve this information from the code documentation, and even the code developer may not

[3] it can be defined as the tributary volume around a node.

know it. Thus, it is recommended to derive the theoretical order of convergence if it is not available. If the theoretical order of convergence cannot be obtained, the acceptance criteria must be changed to convergence check only. The convergence check is not as reliable as the order of convergence check. Thus, convergence check test power in finding code bugs is not guaranteed which weakens the code verification process.

3.2.2.2 Observed Order of Convergence

The observed order of convergence is obtained from the discrete solution on different grid sizes. The discretization error is a function of the grid size h (Knupp et al. [31]).

$$E = E(h) \tag{3.4}$$

The discretization error is proportional to h^p, and $p > 0$ is the discretization algorithm theoretical order of convergence. Thus, the discretization error for a consistent method can be written as in Equation (3.5):

$$E = \alpha h^p + H.O.T \tag{3.5}$$

where α is a constant and $H.O.T$ are high order terms. The error description in Equation (3.5) is applicable to any consistent discretization method such as finite element, and finite volume, but a sufficiently smooth solution is required. Assuming that the discrete solutions on two different grids, fine mesh h_1 and coarse mesh h_2, are in the asymptotic range, the first term in the discretization error dominates the high order terms, and the discretization error is estimated by Equation (3.6) or Equation (3.7).

$$E(h_2) \approx \alpha h^p \tag{3.6}$$

$$E(h_1) \approx \alpha \left(\frac{h}{r} \right)^p \tag{3.7}$$

where r is the refinement ratio. By dividing Equation (3.6) by Equation (3.7), the ratio of the discretization error writes:

$$\frac{E(h_2)}{E(h_1)} = r^p \tag{3.8}$$

From Equation (3.8), the observed order of convergence p can be estimated using Equation (3.9).

$$p = \frac{log\left(\frac{E(h_2)}{E(h_1)}\right)}{log(r)} \tag{3.9}$$

3.3 Solution Verification Assessment Guidelines

Solution verification for CWE simulations is not a straightforward task. It requires data measures that can represent mean flow, turbulence-statistics, and eddy-statistics. Thus, the first crucial step for assessing numerical models is finding reliable, reproducible reference data that provide detailed information about important flow parameters, against which the model performance and uncertainty can be assessed (Hertwig et al. [98, 99]). In this section, block ***(2)*** of the verification procedure in Figure 3.1 is introduced. As concluded from the literature review in Section 2.2, the main emphasis for solution verification was discretization error/uncertainty estimators with minimal focus on unsteady CFD and marginal usage in CWE (Franke [14]). The motivations towards the use of solution verification in unsteady flows simulations are (Eça [23]):

- CFD simulations require the assessment of their numerical uncertainty to establish their credibility,
- Methods for uncertainty estimation based on grid refinement studies (statistically steady flows) are available in the literature,
- CFD applications are no longer restricted to (statistically) steady flows,
- In statistically unsteady flows[4], numerical errors are more complicated to handle. Iterative and discretization errors have more than one contribution,
- Developing guidelines for quality assessment of practical calculations of unsteady flows, since there is no technical reason to present numerical simulations of practical calculations without the indication of numerical uncertainty (Eça et al. [24]).

[4] defines flows that do not depend on time as steady flows, whereas unsteady flows depend on time.

3.3.1 Properties of the Solution Verification Framework

Dealing with numerical errors for eddy-resolving simulations is not an easy task. The complexity of the problem leads to misleading results. The complexity comes from the interaction between numerical and modeling errors. One more factor which contributes to the complexity of the problem is the unsteadiness of the flow regime. The unsteadiness of the flow adds more sources of uncertainty such as statistical sampling error, which is typically ignored in the literature. Methods for quantification of these errors are proposed in this research. In CWE simulations, representation of natural wind flow is an additional complexity added to the solution quantification process. Consequently, the grid design should be good enough to preserve the turbulent properties of the incoming flow. First of all, let us clarify the objectives and properties of the newly proposed framework:

- Identify testing procedures for the estimation of the numerical error. Numerical error and modeling error (sub-grid scale/filter width definition error) cannot be easily separated. Therefore, we combine the two sources of errors as one term called numerical error, because both are associated with the grid size. It will not be possible to separate the numerical and modeling errors, especially when there is no input parameter to customize the filter width. It should be kept in mind, that the same order of convergence is preserved for the Navier-Stokes discretization and the sub-scale equation discretization. Moreover, the filter width is correlated to the grid size, so both error terms can be combined. For both VMS and implicit LES models, the grid size plays the role of the filter width definition. Thus, both terms grid size and filter width can be combined.

- It can be safely said that this approach is applicable to eddy-resolving simulations.

- The framework is to be robust, meaning that the software user can use it without any knowledge of the implementation details, and it can be used for black-box testing. Finally, the methodology should be an a-posteriori-based procedure.

- The use of statistical tools with uncertainty estimates to identify the quality of the results and compare them between different simulations. It mainly indicates if the simulations are run for sufficient time to converge statistics.

- Based on the literature review in Section 2.2, LSR1 is developed to deal with data, not in the asymptotic range, which is very useful for practical applications. The LSR1 uncertainty estimation proposed by Eça et al. [12] is used as the basis for the numerical error/uncertainty estimation. The newly proposed methodology has an extension to improve the treatment of the unsteady nature of the problem, the uncertainty in estimating the cell size (h), and the definition of the order of convergence (p) for non-monotonically converging data.

3.3.2 Procedures for Solution Verification Framework

Because of the challenges imposed from the application of interest, a discretization uncertainty estimate only is not sufficient. Therefore, a measure of the statistical convergence uncertainty and monitoring of the generated wind properties from the turbulent inlet to the obstacle are needed. Also, not only first-order statistics but extreme value analysis is required. For any quantity of interest, the numerical error/uncertainty estimator shown in Section 3.3.5 can be applied. The guidelines for solution verification of CWE simulations are as follows:

1. Determine the characteristics of the application of interest.
2. Design of the computational domain, grid size, simulation time, and time step size.
3. Selection of refinement strategy. It is highly dependent on the available computational power and time frame of the simulations. It can be achieved by one or a combination of the following options:
   a) Evaluate spatial discretization error: Perform spatial refinement with the same time step for all grid levels, and keep the *CFL* number below some value ideally one.
   b) Evaluate time discretization error: Perform temporal refinement with the finest grid.
   c) Evaluate the combined space/time discretization error. If space and time discretization schemes are of the same formal order,

then refine in space and time with the same refinement ratio. In this case, the spatial and temporal discretization errors can be grouped in one error term.

d) For each grid, run simulations at different time steps. Then, spatial and temporal discretization errors can be evaluated separately.

4. Selection of the SRQ: There are different quantities of interest or measurements at different locations for each application. In CWE, the following variables are the most important:

   – Forces acting on the obstacle (Integral quantity),

   – Obstacle surface pressure (Local quantity),

   – Velocity at different locations from the inlet to the obstacle at reference height and the velocity profile (Local quantity).

5. Evaluate statistics for the SRQs: for all variables, compute the first-order statistics, mainly:

   - Mean value,
   - Standard deviation,
   - Root mean square,
   - Extreme value analysis (Minimum, Maximum).

6. Monitoring of the generated wind characteristics from the inlet to obstacle: compute the turbulent kinetic energy (TKE), turbulent kinetic energy dissipation rate (TKEDR), and both integral time and length scales for all the velocity points as follows:

   – Compute TKE:
   It can be computed for the entire domain or at selected points at each time step. Having a velocity vector $\vec{U} = \overline{U} + U'$, the turbulent kinetic energy can be defined as:

$$TKE = \frac{1}{2}\overline{U_i'^2} \tag{3.10}$$

   – Compute TKEDR:
   It can be computed using several techniques, and the simplest

is the first derivative of the TKE DeBonis [48]. Other techniques are presented in Wacławczyk et al. [100] for signals with low sampling frequency or sampling time.

$$TKEDR = -\frac{d(TKE)}{d(t)} \tag{3.11}$$

The TKEDR is computed in the discrete form using a second-order accurate central difference scheme.

– Compute integral length and time scales: It can be computed using autocorrelation function integrated to first zero crossing. Further readings in this topic can be found in Nicolaides et al. [101].

Also, power spectral densities for each grid can be compared to Kaimal spectra at a selected location.

7. Classify and evaluate sources of errors and uncertainties for each measured quantity, as shown in Section 3.3.5.3. Figure 3.3 shows different error sources and in a CWE simulation. In subsequent sections, each source is discussed in detail.

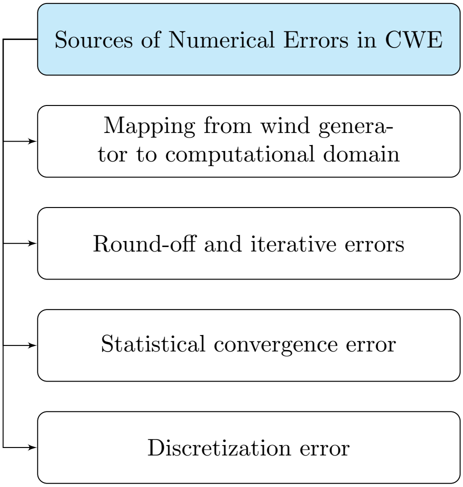

Sources of Numerical Errors in CWE
Mapping from wind generator to computational domain
Round-off and iterative errors
Statistical convergence error
Discretization error

Figure 3.3: Main sources of numerical error/uncertainty in a CWE simulation

3.3.3 Round-off and Iterative Errors

The discrete solution of a steady or unsteady flow problem is typically obtained in an iterative process. Therefore, not only discretization errors but also iterative and round-off errors are affecting the solution quality. The solution procedures of unsteady flows are challenging, where spatial and temporal resolutions have to be carefully chosen. An additional complexity originates when dealing with turbulent flows (Oberkampf et al. [8], Roache [11], Knupp et al. [31], and Eca et al. [102]). Time integration is usually performed with schemes, which require the solution of a non-linear system of equations at each time step. If convergence criteria are not well-defined in solving the non-linear equation system, the iterative error propagates to the next time step (Eca et al. [102]). It is stated by Eca et al. [102] that "a significant influence of the convergence criteria applied at each time step on the numerical accuracy of the solution was experienced by all participants". In the scope of this work, the risk of dealing with round-off and iterative errors is avoided by using machine precision and imposing stringent convergence criteria, respectively.

3.3.4 Statistical Convergence Error

In many industrial applications, the main engineering quantities of interest are time-averaged values and sometimes a fluctuation measure around the mean. To compute the system response quantities in time-varying CFD simulations of statistically stationary turbulent flows, a fully developed flow state should be first achieved (Gant [52]). There are several approaches to judge whether a flow has reached a fully-developed state. Monitoring the main system response quantity of interest and plotting its variation over time is a commonly used approach. An accurate computation can be the running average of the parameter until it reaches a steady value. On the other hand, flows are more complex in many industrial applications. Therefore, there may be subtle long-time-period oscillations that are difficult to characterize in terms of a single frequency (Gant [52]). There are some techniques presented in the literature to compute statistics with confidence (Gant [52], Gant [64], Garcia et al. [74], Haughton et al. [103], Theunissen et al. [104], Politis et al. [105], and Patton et al. [106]). Measuring statistics with confidence interval (CI) is good to account for uncertainty in the QOI resulting from simulation repetition. Statistical uncertainty provides practitioners with a rational way of evaluating the statistical error component's significance on repeated trials, which can be used for data validation with experimental or other numerical results (Garcia et al. [74]). Eventually, bootstrapping is the most widely accepted and robust approach, which does not require the repetition of simulations. It is used to compute the statistical uncertainty in the scope of this work.

3.3.4.1 Bootstrapping

Bootstrapping, first introduced in 1979 by Efron, is a re-sampling technique used to estimate statistics, such as mean and standard deviation, with uncertainty. It is extensively used in econometrics, while in recent years, fluid experimentalists have been using bootstrapping to avoid experiment repetitions to get statistical inference. The method overcomes most of the deficiencies found in standard techniques. In standard techniques, no equations are available to compute the standard error of parameters with more complicated expressions than sample mean and variance, such as turbulent kinetic energy, time and length scale, and turbulent kinetic energy dissipation rate. Bootstrapping requires no theoretical calculations and is available no matter how complicated the parameter of interest (Efron et al. [107]). Bootstrapping estimates the non-parametric standard error with a confidence interval. There are several cases where bootstrap performance is questionable, such as:

1. Extreme value estimation
2. Too small sample size
3. Long-range dependencies

The bootstrap technique consists of two components: distribution estimate (re-sampling) and estimation for a confidence interval. The two components define the general procedures for bootstrap as (Stephanie [108]):

1. Re-sample a data set X number of times,
2. Find a summary statistic (called a bootstrap statistic) for each of the x samples,
3. Estimate the standard error for the bootstrap statistic using the standard deviation of the estimated bootstrap distribution.

The bootstrap re-sampling technique can be either parametric or non-parametric. Karlis [109] defines parametric and non-parametric bootstrap as follows:

- Parametric bootstrap: it is assumed that the data belongs to a parametric family of distributions (e.g., exponential distribution, normal distribution, etc.), and parameters are estimated from the sample. Then, bootstrap samples are generated from the estimated parameters. Figure 3.4 shows the schematic representation for the parametric bootstrap where simulated samples are derived from the fitted distribution.
- Non-parametric bootstrap: it is mainly developed for random data where the statistical distribution is not known. The bootstrap samples are generated from the original sample. Figure 3.5 shows the schematic representation for the non-parametric bootstrap where

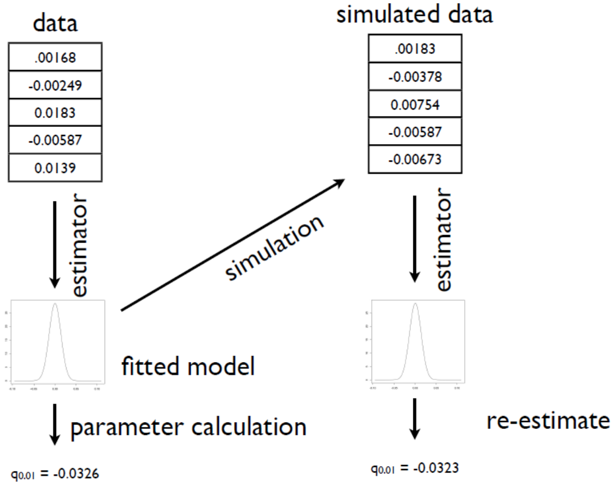


Figure 3.4: Schematic representation for the parametric bootstrap re-sampling technique (Shalizi [110])

new data is fitted by re-sampling (with replacement) from the original data.

Bootstrap confidence interval can be calculated by several techniques, such as pivotal interval, studentized interval, biased-corrected accelerated interval (BCA), etc. An extensive review of these methods can be found in Efron et al. [107], Karlis [109], and Davison et al. [111]. Overall, the BCA is shown to be the preferred method in literature because it is second-order accurate and valid for a variety of problems.

Finally, a biased-corrected accelerated moving block non-parametric bootstrap (BCA-MBB) by Efron et al. [107] is used to quantify statistical convergence uncertainty. The moving block bootstrap (MBB) was developed to preserve the real correlation in the signal to quantify the statistical uncertainty in turbulence parameters (Garcia et al. [74]). The approach was used in the literature to deal with a problem similar to the scope of CWE applications (Garcia et al. [74] and Theunissen et al. [104]). The MBB technique can be applied for a signal with N samples. A block size b is to be defined to generate all the possible contiguous blocks of length $(N - b + 1)$

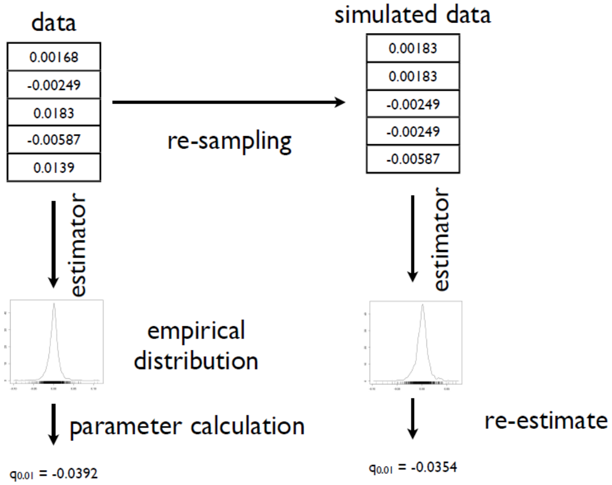


Figure 3.5: The non-parametric bootstrap re-sampling technique (Shalizi [110])

total blocks. One then randomly samples N/b blocks with replacement from the total number of blocks and processes them.

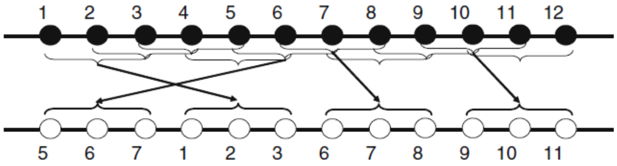


Figure 3.6: A schematic illustration of the non-parametric MBB for time signal using a block size equals to three samples. The black series represents the original time series and the white circles denote one of the bootstrap realizations using MBB. Numbers indicate the sample order in the original signal (Garcia et al. [74])

3.3.4.2 Bootstrapping Short Example

Figure 3.7 shows signals for the drag and cross-wind force coefficients for a $2-D$ flow around a square. The BCA-MBB technique[1] with 1000 samples for the mean value is applied to the force coefficient time series to estimate the mean values' statistical uncertainty (Evans [112]).

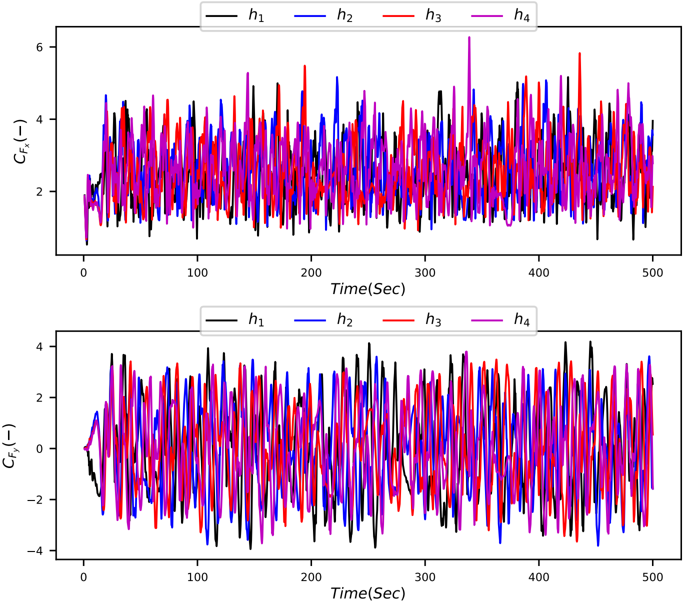


Figure 3.7: Force coefficients signals for $2-D$ flow around a square

Table 3.1 summarizes the results of the bootstrap. It can be noticed that the statistical uncertainty in the mean for C_{F_y} is very high. By looking at the signals, it can be noticed that the mean is oscillating around zero, and the variation is very high. Thus, this might require longer simulation time, a higher number of bootstrap sample size, higher sampling frequency, or a

[1] it is implemented in scikits.bootstrap python and Last accessed on January 8, 2019 `https://github.com/cgevans/scikits-bootstrap`

different approach. On the other hand, the mean statistical uncertainty for the quantity of interest C_{F_x} is around 0.7%, which was going to be neglected if the bootstrap was not performed. Finally, by changing the sampling interval, the results might improve.

Grid	C_{F_x}		C_{F_y}	
	Mean	Uncertainty [%]	Mean	Uncertainty [%]
h_1	2.4877	0.643	0.0427629	78.9
h_2	2.62258	0.687	0.0130478	65.8
h_3	2.58	0.763	0.06921	55.1
h_4	2.6277	0.776	0.02437	65.1

Table 3.1: Results of applying the BCA-MBB to evaluate the statistical mean uncertainty in time series

3.3.5 Discretization Error Estimation

The discretization error estimation technique of unsteady turbulent flows is discussed in this section. At this point, it is very crucial to clarify that the discretization error estimate has two components: first, the error estimation, and second, the uncertainty quantification. In the literature, this is referred to as discretization uncertainty estimation because the error is multiplied by a safety factor to account for solutions not being in the asymptotic range (Phillips [49]). From Section 2.2, it can be clearly seen that great emphasis was given to Multi-Grid estimators, because they are more realistic for practical applications. It can also be noticed that almost all the approaches for Multi-Grid estimators are based on the Richardson Extrapolation. The general form of the Richardson Extrapolation, introduced in Section 2.2.1.5, can be written as:

$$\epsilon_h = \phi_h - \tilde{\phi} = \alpha_1 (h)^{p_h} + \alpha_2 (\Delta t)^{p_t} \tag{3.12}$$

The error function (ϵ_h) is dependent on the typical grid size (h), spatial order of convergence (p_h) and temporal order of convergence (p_t). To study the effect of the definition of (h) and order of convergence (p) on the quantification of error, let's reduce Equation (3.12) to Equation (3.13) by assuming both spatial and temporal order of convergence are of the same order, such that ($p = p_h = p_t$).

$$\epsilon_h = \alpha (h)^p \tag{3.13}$$

Figure 3.8 left shows the typical cell size (h) on the x-axis and error function (ϵ_h) on the y-axis. The ϵ_h is calculated for different convergence orders ($p = 1, 1.5, 2$). It can be seen that the error is highly affected by value of h, especially for $p > 1.0$. Consequently, the typical grid size definition for unstructured locally refined meshes is of great impact on the error estimation. having $p = 2.0$ and locally refined mesh, where h can be any value between 0.2 m and 0.05 m, for instance, the calculated error varies from 5 % for $h = 0.05$ m to 20 % for $h = 0.2$ m. Therefore, a new definition for calculating the typical grid size (h) is mandatory and must be sensitive to local refinement.

Figure 3.8 right shows the order of convergence (p) on the x-axis and error function (ϵ_h) on the y-axis. The ϵ_h is calculated for different typical grid

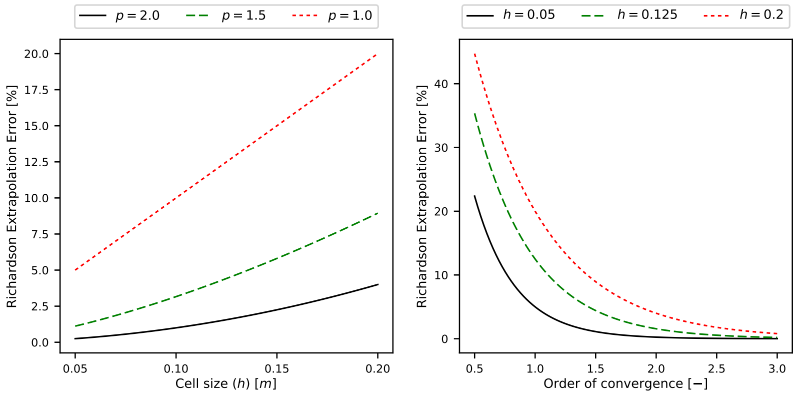


Figure 3.8: The effect of cell size and convergence order variation on Richardson Extrapolation error estimation (Assuming $\alpha = 1.0$)

sizes ($h = 0.05, 0.125, 0.2$) m. It can be seen that for the same h, the error is highly influenced by the value p where for $h = 0.05$ m, the ϵ_h has a variation of ≈ 20 %. This effect is highly affecting the error estimation for anomalous data behavior[5]. Therefore, a special treatment is crucial for the definition of p for anomalous data behavior, which is fairly seen in practical CWE simulations.

In this work, new definitions for typical grid size, treatment of anomalous data order of convergence, and definition for the uncertainty estimator are used together with a least-squares fitting approach to estimate discretization uncertainty for CWE simulations of practical relevance.

3.3.5.1 Definition of the Typical Grid Size

In practical CWE applications, the cell size (h) in Equation (3.13) does not have a unique description. Therefore, further doubts are added to the

[5] Anomalous data behavior means data that does not have a positive observed order of convergence (non-monotonically converging data)

evaluation of the discretization uncertainty. In this section, a measure for h is introduced. In the literature, h is described for any mesh type as follows:

$$h_1 = \begin{cases} \sqrt[3]{\frac{Vol}{\#ofcells}} & \text{in } 3-D \\ \sqrt[2]{\frac{Area}{\#ofcells}} & \text{in } 2-D \\ \frac{Length}{\#ofsegments} & \text{in } 1-D \end{cases} \tag{3.14}$$

Equation (3.14) is strongly valid when all mesh elements have the same size and then systematically refined. However, these meshes are not valid in practical applications. In practical CWE simulations, the domain is divided into a number of zones (n_z), and each zone has a different cell size. Typically, a mesh is refined near the structure. Therefore, a new measurement is proposed for the definition of a typical grid size. In this work, the weighted average cell size is calculated based on the importance of the cell's zone location. This approach assumes that as the computational domain is locally refined, simulation quality is improving for predicting SRQ. Moreover, the concept of mesh zones is clarified in Figure 3.9.

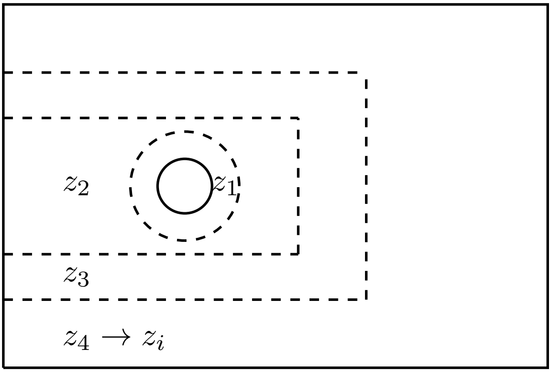


Figure 3.9: The definition of mesh zones concept used to compute the cell size

The approach is defined by:

$$h_2 = \sum_{i=1}^{n_z} w_i h_{z_i} \tag{3.15}$$

where h_{z_i} is the cell size corresponding to zone z_i, n_z is the number of zones and w_i is defined as:

$$w_i = \frac{\frac{1}{h_{z_i}}}{\sum_{i=1}^{n_z} \frac{1}{h_{z_i}}} \tag{3.16}$$

ensuring that

$$\sum_{i=1}^{n_z} w_i = 1 \tag{3.17}$$

This calculation will result in smaller grid size than the one computed from Equation (3.14). Assuming that the actual cell size can be any value between h_1 and h_2 with equal probability, the estimation of h can be described by a normal distribution with mean $\left(\mu = \frac{h_1+h_2}{2}\right)$ and standard deviation $\left(\sigma = \left|\frac{h_2-h_1}{2}\right|\right)$.

3.3.5.2 Cell Size Measure: Short Example

To clarify the concept introduced in Section 3.3.5.1, let's consider a $1-D$ case with length $L = 10$ and divided into four zones $n_z = 4$ as shown in Figure 3.10. Normally, fine regions correspond to a significant zone, where all the SRQs are evaluated.

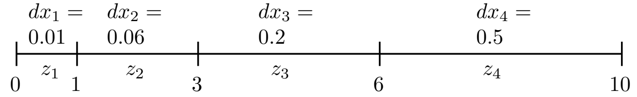


Figure 3.10: 1-D case zones division

By applying the conventional equation, the general cell size using Equation (3.14) is $h_1 = 0.06397$. On the other hand, applying the weighted approach using Equation (3.15) results in $h_2 = 0.03236$. The true typical grid size can be any value in the interval $[h_1, h_2]$ with equal probability. Therefore, the estimated typical grid size can follow a normal distribution with parameters ($\mu = 0.048165$, $\sigma = 0.015805$). In other words, the two estimates have around 35% difference in the mean estimated value in this case. Thus, the uncertainty from the estimation of h must be transferred to the discretization uncertainty, as shown in Figure 3.8.

3.3.5.3 Discretization Uncertainty Estimator

The presentation of Eça et al. [24] compared the performance of the most widely used discretization uncertainty estimators in the literature. It was concluded that the FS and LSR1 are the most acceptable for practical simulations. Hereafter, the procedure proposed by Eça et al. [12] is adapted for the estimation of the numerical uncertainty. The approach is based on using a least-squares fit for data estimated with power series expansions, which is a function of the typical grid size. The LSR1 is restructured according to the new definition of h (Section 3.3.5.1) and the assumption of p for anomalous data behavior.

Description of the Discretization Error Richardson extrapolation can be applied to numerical methods that possess power series expansion for the discretization error under sufficient conditions (Allen [113]). Discretization error ϵ_ϕ can be described as:

$$\epsilon_\phi \simeq \delta_{RE} = \phi_i - \phi_0 = \alpha h_i^p \tag{3.18}$$

where δ_{RE} is the residual, ϕ_i is the numerical solution of grid $i = 1, 2, ... n_g$, n_g is the number of grids, ϕ_0 is the predicted exact solution, α is a constant value, h_i is the typical grid size h of grid i and p is the order of convergence. Equation (3.18) is also known as the generalized form of the Richardson extrapolation. Equation (3.18) is solved for ϵ_ϕ, α, p with the least-squares sense as shown in appendix D. The estimation of ϵ_ϕ is considered reliable if Equation (3.18) is solved under the following assumptions:

1. At least three grids are needed: This assumption can be violated when the formal order of convergence is used to evaluate the error; then, only two grids are sufficient. The assumption of using p_f is not justified in practical calculations.

2. Guarantee that the leading term of the power series expansion is sufficient to estimate the error (Eça et al. [12]): This assumption is valid only if the grids are in the asymptotic range. Asymptotic range solutions are almost impossible to achieve in practical calculations.

3. Geometrical similarity must be achieved for all grids, because the density of the grid is represented by one parameter (cell size (h)): This is very difficult to achieve in practical computations where unstructured meshes are used. The definition of h for unstructured grids has been discussed in Section 3.3.5.1.

The assumptions stated above are violated in practical CWE calculations (Eça et al. [12] and Eca et al. [20]). Therefore, noisy data is produced from different grids. Equation (3.18) is not sufficient to estimate the error and more rules are needed as follows:

1. At least four grids are used to have a reasonable error estimate.

2. Assuming the code is theoretically second-order accurate, three error estimators are defined to formulate the error function as

$$\epsilon_{\phi} \simeq \delta_1 = \phi_i - \phi_0 = \alpha h_i \tag{3.19}$$

$$\epsilon_{\phi} \simeq \delta_2 = \phi_i - \phi_0 = \alpha h_i^2 \tag{3.20}$$

$$\epsilon_{\phi} \simeq \delta_{12} = \phi_i - \phi_0 = \alpha_1 h_i + \alpha_2 h_i^2 \tag{3.21}$$

Least-Squares for Error Estimation Having ($n_g \geq 4$), least-squares can be used to minimize the sum of squared deviations of the simulated ϕ_i from the estimates of their true value ϕ_0, which will treat the scatter in the data, and provide a reliable error estimate. In general, the estimation of ϕ_0 is determined from the minimum of the least-squares fitting as (Rawlings et al. [114]):

$$S_{RE}\left(\phi_0, \alpha, p\right) = \sqrt{\sum_{i=1}^{n_g} \left(\phi_i - \left(\phi_0 + \alpha h_i^p\right)\right)^2}, \tag{3.22}$$

$$S_1(\phi_0,\alpha)=\sqrt{\sum_{i=1}^{n_g}\left(\phi_i-(\phi_0+\alpha h_i)\right)^2}, \tag{3.23}$$

$$S_2(\phi_0,\alpha)=\sqrt{\sum_{i=1}^{n_g}\left(\phi_i-(\phi_0+\alpha h_i^2)\right)^2}, \tag{3.24}$$

$$S_{12}(\phi_0,\alpha_1,\alpha_2)=\sqrt{\sum_{i=1}^{n_g}\left(\phi_i-(\phi_0+\alpha_1 h_i^1+\alpha_2 h_i^2)\right)^2} \tag{3.25}$$

Assuming that fine grids are producing more reliable results, it is possible to use a weighted least-squares by minimizing the following functions:

$$S_{RE}^w(\phi_0,\alpha,p)=\sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i-(\phi_0+\alpha h_i^p)\right)^2}, \tag{3.26}$$

$$S_1^w(\phi_0,\alpha)=\sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i-(\phi_0+\alpha h_i)\right)^2}, \tag{3.27}$$

$$S_2^w(\phi_0,\alpha)=\sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i-(\phi_0+\alpha h_i^2)\right)^2}, \tag{3.28}$$

$$S_{12}^w(\phi_0,\alpha_1,\alpha_2)=\sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i-(\phi_0+\alpha_1 h_i^1+\alpha_2 h_i^2)\right)^2} \tag{3.29}$$

where the weights w_i are dependent on the cell size:

$$w_i=\frac{\frac{1}{h_i}}{\sum_{i=1}^{n_g}\frac{1}{h_i}}, \tag{3.30}$$

ensuring that

$$\sum_{i=1}^{n_g} w_i = 1 \tag{3.31}$$

The solution to the least-squares fitting is explained in appendix D. In this work, eddy-resolving simulations for unsteady flows are considered. It adds the complication of time discretization schemes. Therefore, spatial and temporal uncertainties are to be calculated. Three approaches can be used in this matter:

- First, evaluate the spatial discretization uncertainty at a fine time step with grid refinement following procedures in Figure 3.11. Then, evaluate the temporal discretization uncertainty at the finest spatial grid and apply the same procedures. In this situation at least eight simulations are required.
- Second, modify the general error function Equation (3.18) used in the LSR approach to include the temporal discretization error term as:

  $$\delta = \phi_i - \phi_0 = \alpha_1 h_i^{p_s} + \alpha_2 (\Delta t)_i^{p_t}, \tag{3.32}$$

  and the minimization function can be written, for example, as:

  $$S_{RE}\left(\phi_0, \alpha_1, \alpha_2, p_s, p_t\right) = \sqrt[2]{\sum_{i=1}^{n_g} \left(\phi_i - \left(\phi_0 + \alpha_1 h_i^{p_s} + \alpha_2 (\Delta t)_i^{p_t}\right)\right)^2} \tag{3.33}$$

  where α_1 and α_2 are constants, p_s is the spatial order of convergence, and p_t is the temporal order of convergence. In this situation at least eight simulations are required with minimum three grids.
- Third, the spatial and time discretization schemes are of the same theoretical order of convergence. Hence, the error can be evaluated by a sequential space-time refinement with the same order and the error function can be written as:

  $$\delta = \phi_i - \phi_0 = \alpha h_i^p, \tag{3.34}$$

and the minimization function can be written, for example, as:

$$S_{RE}\left(\phi_0,\alpha,p\right)=\sqrt[2]{\sum_{i=1}^{n_g}\left(\phi_i-\left(\phi_0+\alpha h_i^p\right)\right)^2} \tag{3.35}$$

In this situation at least four simulations are required.

Procedures for Error and Uncertainty Estimator The last point to consider in the solution verification framework is the complete procedure for the discretization uncertainty estimation. The procedure is summarized in Figure 3.11. These are the main issues handled by this new approach:

1. Treatment of anomalous data behavior: The key difference here is assuming that the order of convergence p_j can be any value in the range $[0.5,2]$. This range is used to have a fair error estimate instead of using fixed value such as $p_j=1$ or $p_j=2$ which are proposed in Eça et al. [12]. Then, for each value p_j in the range $[0.5,2]$, the fits S_j and S_j^w are solved. Also, the fits S_{12} and S_{12}^w are evaluated. Then, all the fits are solved for ϕ_0 and fit standard deviation σ is computed. The error function for the fits with arbitrary convergence order p_j is:

   $$\delta_{p_j}=\phi_i-\phi_0=\alpha h_i^{p_j} \tag{3.36}$$

   Then, the function for the fit with the minimum σ is used to compute the error. This improves the error estimate's ability to deal with anomalous data behavior, which can be especially experienced in local quantities. The modification can be seen in Figure 3.11. In anomalous data behavior, it will be very complicated to get the observed order of convergence using the generalized minimization problem. Consequently, a predefined order of convergence is used, which varies from 0.5 to two. The maximum p_j value is defined as the theoretical convergence order while using the minimum accepted value for the order of convergence. Finally, the fit with the minimum standard deviation is used to evaluate the discretization error.

2. Computation of the safety factor F_s: The uncertainty is based on an error estimation multiplied by a safety factor. The additional term F_{sh} in the safety factor computation is the propagation of the uncertainty

of the error function from the input parameter h_i (see Section 3.3.5.1) to the predicted exact solution ϕ_0. Monte Carlo method is used to estimate the distribution of ϕ_0. The steps for the Monte Carlo method are:

- The limits of the possible inputs are defined as described in Section 3.3.5.1,
- Use a normal probability distribution to generate random inputs,
- Perform the computation of ϕ_0 using the random input,
- Compute the mean μ_{out} and standard deviation σ_{out} for the output quantity ϕ_0.

The factor F_{sh} is computed as following:

$$F_{sh} = \frac{3\sigma_{out}}{\mu_{out}} \tag{3.37}$$

Finally, $3\sigma_{out}$ is used to ensure a 99.9% confidence interval and F_{sh} is used to calculate the safety factor as shown in Figure 3.11.

3. A safety factor (F_s) is defined to estimate uncertainty. For a monotonically converging data set with $0.5 \leq p < 2.1$, the F_s is set to 1.25, otherwise F_s is set to $1.25 + F_{sh}$. Then, a data range parameter ($\Delta\Phi$) is defined to judge on the quality of the data fit. ($\Delta\Phi$) is computed as

$$\Delta\Phi = \frac{(\phi_i)_{max} - (\phi_i)_{min}}{n_g - 1} \tag{3.38}$$

The uncertainty for a data point ϕ_i is computed as

$$U_\phi(\phi_i) = F_s \epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}| \quad if\ \sigma < \Delta\Phi, \tag{3.39}$$

$$U_\phi(\phi_i) = F_s \frac{\sigma}{\Delta\phi}(\epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|) \quad if\ \sigma \geq \Delta\Phi. \tag{3.40}$$

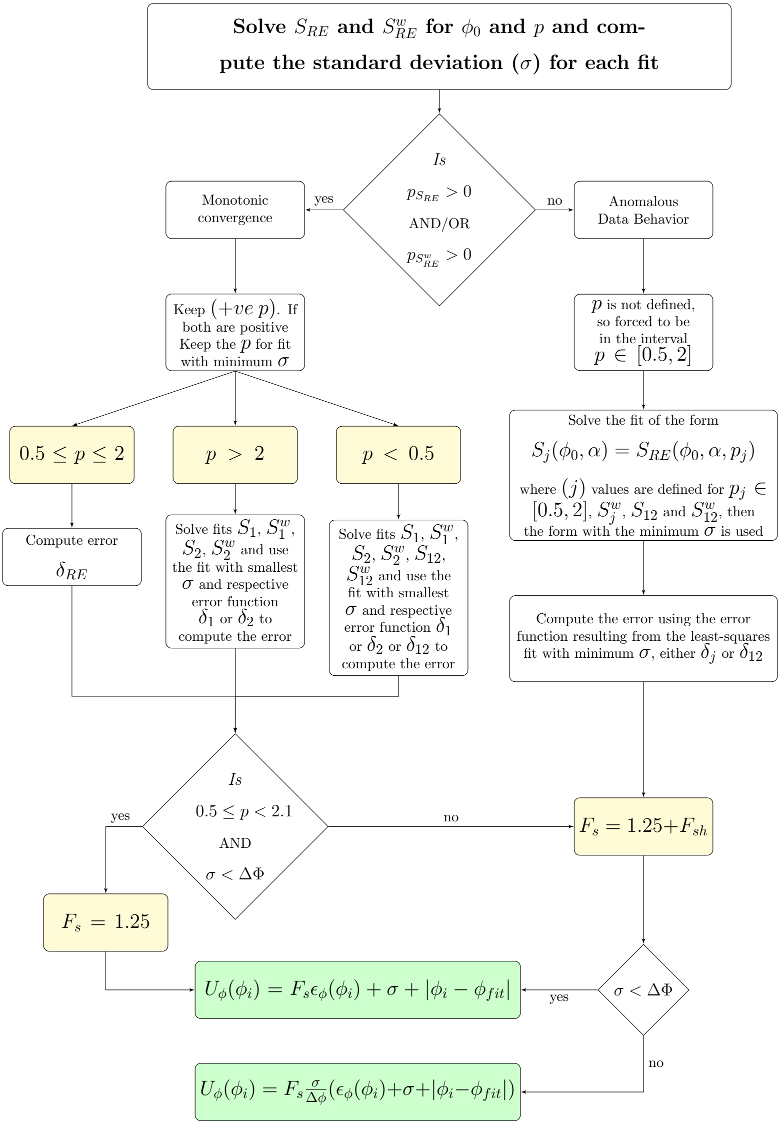

Solve S_{RE} and S^w_{RE} for ϕ_0 and p and compute the standard deviation (σ) for each fit
Is $p_{S_{RE}} > 0$ AND/OR $p_{S^w_{RE}} > 0$
yes
no
Monotonic convergence
Anomalous Data Behavior
Keep $(+ve\ p)$. If both are positive Keep the p for fit with minimum σ
p is not defined, so forced to be in the interval $p \in [0.5, 2]$
$0.5 \leq p \leq 2$
$p > 2$
$p < 0.5$
Solve the fit of the form $S_j(\phi_0, \alpha) = S_{RE}(\phi_0, \alpha, p_j)$ where (j) values are defined for $p_j \in [0.5, 2]$, S^w_j, S_{12} and S^w_{12}, then the form with the minimum σ is used
Compute error δ_{RE}
Solve fits S_1, S^w_1, S_2, S^w_2 and use the fit with smallest σ and respective error function δ_1 or δ_2 to compute the error
Solve fits S_1, S^w_1, S_2, S^w_2, S_{12}, S^w_{12} and use the fit with smallest σ and respective error function δ_1 or δ_2 or δ_{12} to compute the error
Compute the error using the error function resulting from the least-squares fit with minimum σ, either δ_j or δ_{12}
Is $0.5 \leq p < 2.1$ AND $\sigma < \Delta\Phi$
yes
no
$F_s = 1.25 + F_{sh}$
$F_s = 1.25$
$\sigma < \Delta\Phi$
yes
no
$U_\phi(\phi_i) = F_s\epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|$
$U_\phi(\phi_i) = F_s\frac{\sigma}{\Delta\phi}(\epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|)$

Figure 3.11: Flowchart for the numerical error/uncertainty estimator procedure based on the concepts in Eça et al. [12]

3.3.5.4 Application of the Discretization Uncertainty Estimator

In this section, the data available from the workshop on estimation of discretization errors based on grid refinement studies[1] is used to evaluate the performance of different discretization uncertainty estimators. The results for flow over a flat plate is considered using a Spalart-Allmaras turbulence model on the coarsest wall spacing grid *(Set 10)*. Data for $Re = 10^7$ and $Re = 10^9$ called *CaseIa* and *CaseIIIa*, respectively, are used. The detailed description for the test case can be found in appendix E. The following uncertainty estimators are used:

- GCI: is introduced in Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer (ASME V&V20 Committee [26])
- GDU: is introduced in Phillips et al. [10] and Phillips et al. [115]
- LSR1: is introduced in Eça et al. [12]
- LSRCONS: is introduced in Section 3.3.5.3 having a conservative measure using $F_s = 1.25$ or $F_s = 3$.
- LSRMONTE: introduced in Section 3.3.5.3 without any modification and assuming 20 % variation on the estimate of the typical grid size.

Applying both GCI and GDU requires grid triplets to estimate the fine grid error. The following grids are used:

- For grid h_1, the three grids h_1, h_2, h_4 are used,
- For grid h_2, the three grids h_2, h_4, h_8 are used,
- For grid h_4, the three grids $h_4, h_{5.818}, h_8$ are used.

where h_1 refers to the finest grid, and h_8 refers to the coarsest grid. Figure 3.12 shows the quantification of the error using different estimates. It can be seen that the LSR1 and LRSCONS are providing similar results, but the LSRCONS is marginally more conservative than LSR1 for the fine

[1] Last accessed on July 30, 2018 `http://web.tecnico.ulisboa.pt/ist12278/Discretization/Workshop_discretization_2017.htm`

meshes. The difference between LSR1 and LSRCONS is the definition of the order of convergence p when having non-monotonically converging data. Also, it can be noticed that LSRMONTE is less conservative than both LSR1 and LSRCONS. This comes from the definition of the F_s. Finally, the GCI and GDU show very close performance. They are highly dependent on the selection of the grid triplets, thus very sensitive to the refinement ratio, as shown in the estimation for h_4.

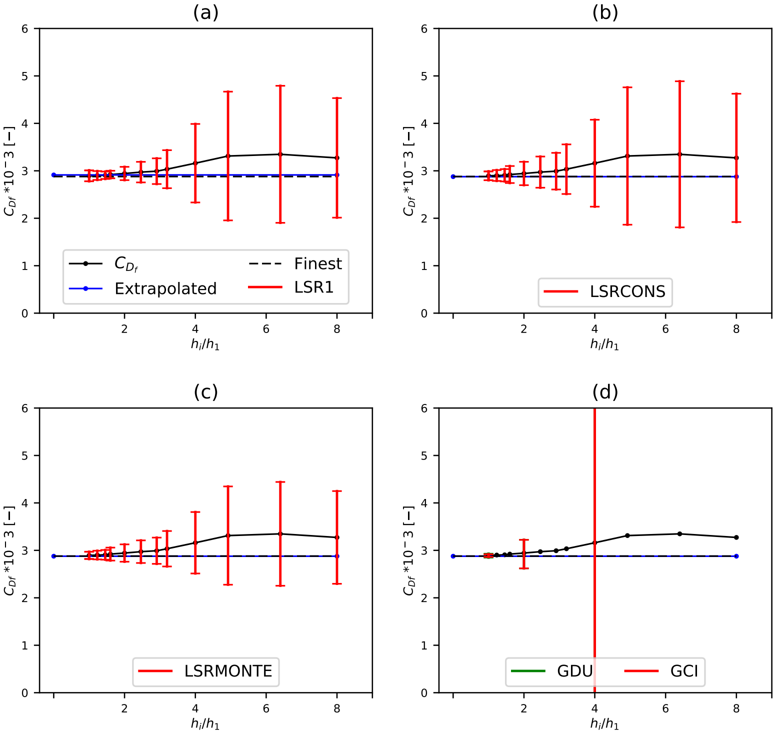


Figure 3.12: Discretization error estimation for the drag friction coefficient C_{D_f} using five approaches on all different grid levels at $Re = 10^7$

Figure 3.13 shows the results of the same analysis for simulations at $Re = 10^9$. The same observations from Figure 3.12 can be clearly seen in Figure 3.13.

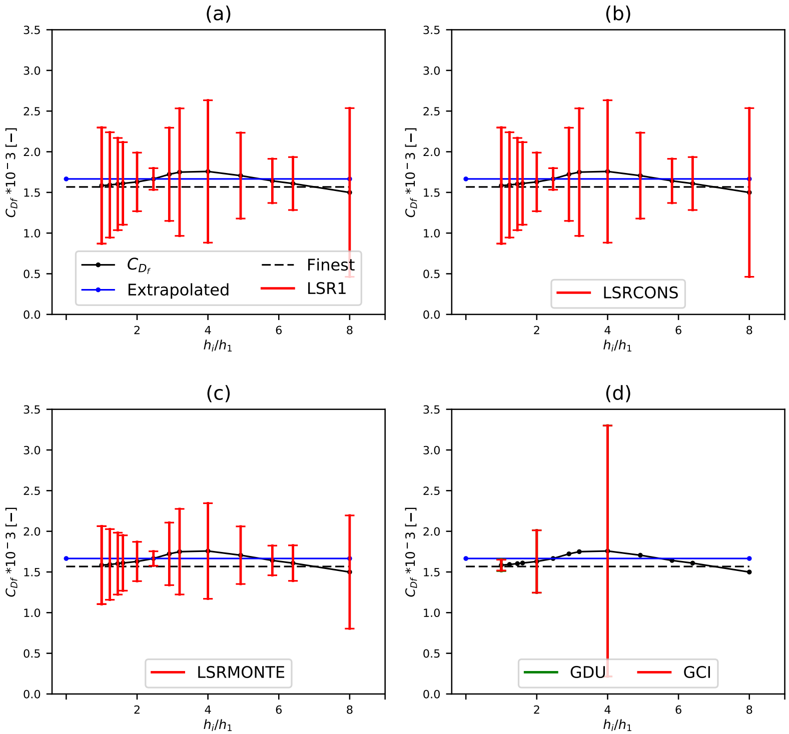


Figure 3.13: Discretization error estimation for the drag friction coefficient C_{D_f} using five approaches on all different grid levels at $Re = 10^9$

In practical applications, running more than seven simulations on different grids is not always feasible and very expensive. Thus, the analysis was performed using the recommended number of grids $n_g = 4$. It can be seen in Figure 3.14 that all the estimators have a good performance, whereas LSRMONTE is the least conservative but still bounds the DNS solution. Moreover, GCI and GDU provide highly conservative measures in case of inconsistency in the refinement ratio, especially being less than two. From Figure 3.12, Figure 3.13 and Figure 3.14, it can be clearly seen that the finest grid solution (DNS solution) is bounded by all the error estimates.

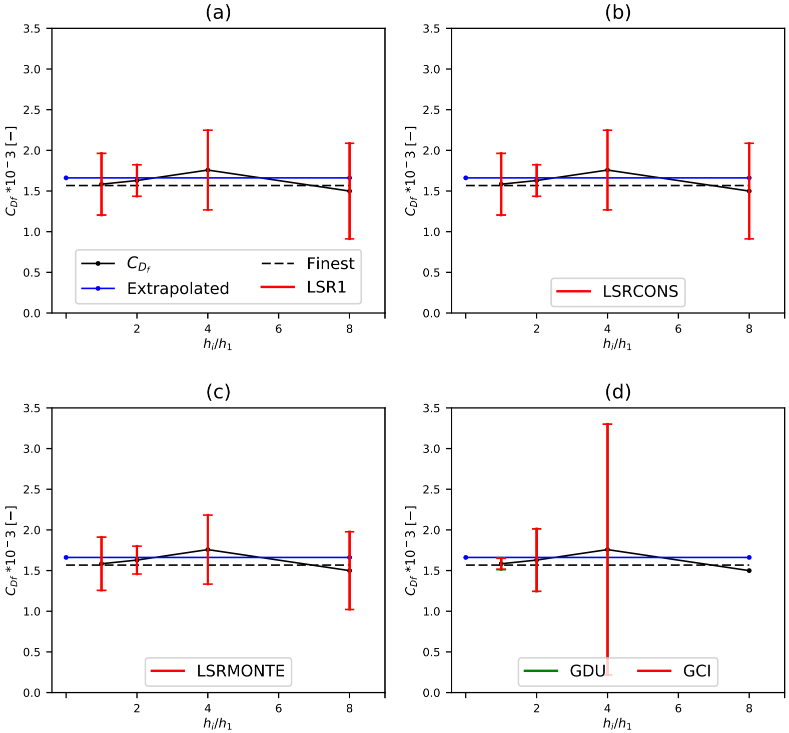


Figure 3.14: Discretization error estimation for the drag friction coefficient C_{D_f} using five approaches on four different grid levels at $Re = 10^9$

The next step is to check the performance of the estimators for local quantity. Error is estimated for the friction coefficient C_f at ten different locations from $x/l = 0.05$ to $x/l = 0.5$. The results shown are the error estimated at the finest mesh. The same conclusions drawn above can be drawn from Figure 3.15 and Figure 3.17. Moreover, the performance of GCI and GDU can be clearly seen in Figure 3.16. Finally, the performance of all the error estimators is exceptional. LSRMONTE has the advantage of being less conservative, which narrows down the true solution bounding interval, while still bounding the exact solution in all the cases, compared to other estimators. LSRMONTE also has the advantage of considering the uncertainty in the estimation of the grid cell size. This feature is very critical in cases where geometrical similarity is not satisfied, which is most probably the situation in practical applications.

In Figure 3.15 and Figure 3.17, it can be observed that at some point where anomalous data behavior is found, the LSRMONTE is less conservative than both LSRCONS and LSR1, yet it still bounds the DNS solutions.

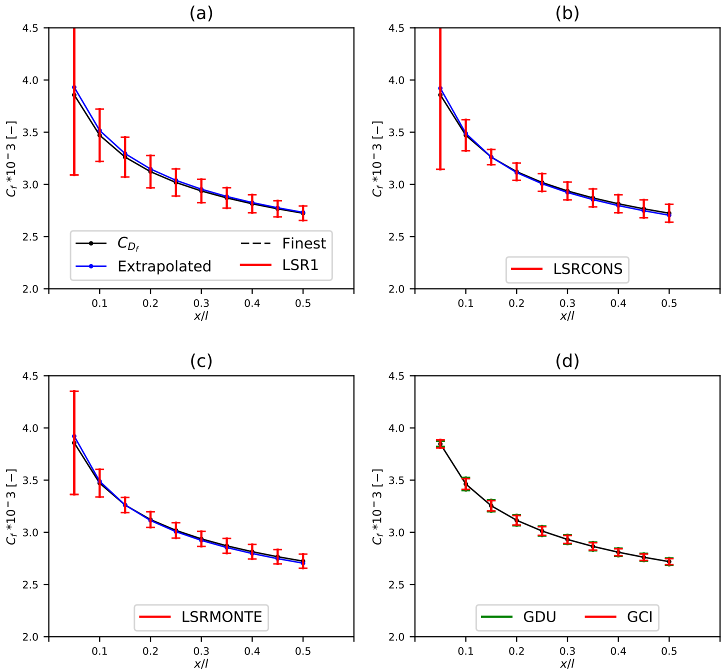


Figure 3.15: Discretization error estimation for the friction coefficient C_f using five approaches on the finest grid level at $Re = 10^7$

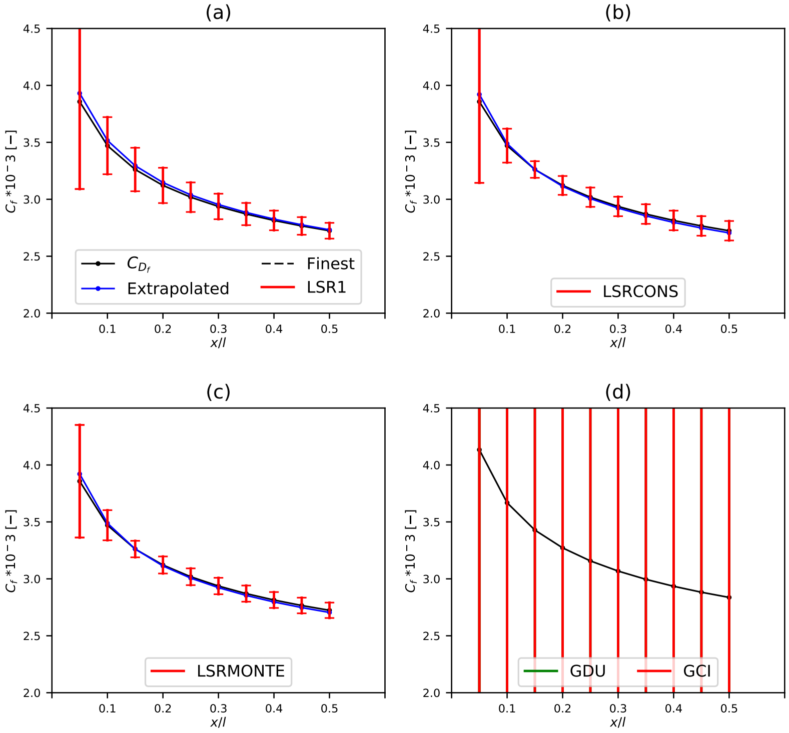


Figure 3.16: Discretization error estimation for the friction coefficient C_f using five approaches on the finest grid level at $Re = 10^7$ using unequally refined meshes for GCI and GDU

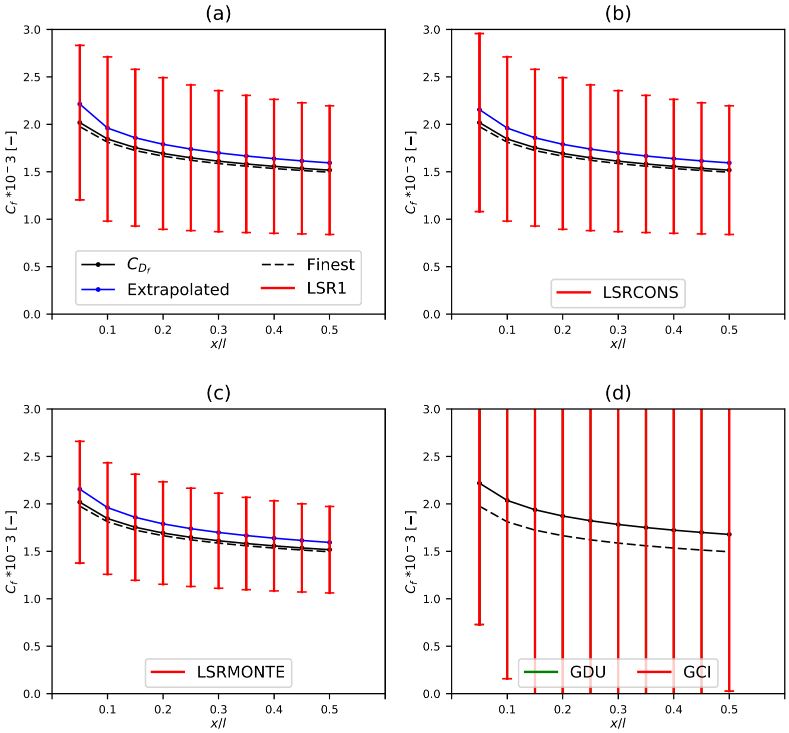


Figure 3.17: Discretization error estimation for the friction coefficient C_f using five approaches on the finest grid level at $Re = 10^9$

3.4 Summary

The newly proposed methodology reveals the importance of performing both code and solution verification to build credibility in the simulation results. Code verification guidelines are introduced, and the applicability of these guidelines with benchmarks is introduced in Chapter 5. After performing code verification, the solution verification procedure is introduced to estimate discretization uncertainty and is called (LSRMONTE). The LSRMONTE is developed with special emphasis on an appropriate definition of typical grid size, estimation of the order of convergence for anomalous data behavior, and quantification of the safety factor for uncertainty estimation. The LSRMONTE showed an outstanding performance over other methods available in the literature, as shown in Section 3.3.5.4. The performance of the LSRMONTE is further investigated in Chapter 6 and Chapter 7. The methodology is developed for CWE simulations mainly but can be possibly applied to general fluid flows. The complete methodology is summarized in appendix C.

Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.

George E. P. Box

CHAPTER 4

THE VMS SIMULATION CODE: KRATOS MULTIPHYSICS

This chapter introduces the theoretical background for the code used to clarify the usage of the developed methodology outlined in Chapter 3. KRATOS Multiphysics is a FEM-based code. The code uses a variational multiscale (VMS) method to design the stabilized finite element formulations. The VMS is based on the decomposition of the solution into resolved and unresolved parts (Codina et al. [116]). The separation is achieved by the definition of large and small scale solution spaces. The projection of the original equations onto the large scale space gives an equivalent problem that depends on the small scale variables. Meanwhile, algebraic and orthogonal models are used to project the original equations onto the small scale space, which is used to motivate a model for the effect of the small scale variables. The VMS is equivalent to the LES simulation approach to a great extent (Codina et al. [117]) because both techniques introduce a separation between the resolved and unresolved parts of the solution. The hypothesis of incompressibility is accepted in the Computational Wind Engineering (CWE) applications range. Thus, this work focuses on the incompressible case of the VMS element using algebraic sub-scales (ASGS) and orthogonal sub-scales (OSS)[1]. More information about the VMS formulation can be found in Cotela [97], Codina et al. [116], Codina [118, 119], Codina [120], Codina et al. [121], and Hanzlicek [122].

4.1 Governing Equations

The notations used in this chapter are adapted from Cotela [97]. The conservation equations for linear momentum and mass in a fluid domain Ω is given by:

$$\rho\partial_t \boldsymbol{u} + \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla\cdot\boldsymbol{\sigma} = \boldsymbol{f} \qquad in \quad \Omega, \quad t \in [0, T[\tag{4.1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad in \quad \Omega, \quad t \in [0, T[\tag{4.2}$$

where $\boldsymbol{u}$ is the fluid velocity field, ρ is the density, $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{f}$ is the external forces acting on the domain. The stress tensor $\boldsymbol{\sigma}$ can be related to the fluid velocity $\boldsymbol{u}$ and pressure p with the assumption of Newtonian fluid. The constitutive relation is described by

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\left(\nabla^s\boldsymbol{u} - \frac{1}{3}(\nabla\cdot\boldsymbol{u})\boldsymbol{I}\right) \tag{4.3}$$

where $\boldsymbol{I}$ is the second order identity tensor, μ the fluid's viscosity and $\nabla^s\boldsymbol{u}$ the symmetric gradient of velocity, which is defined as

$$\nabla^s\boldsymbol{u} = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) \tag{4.4}$$

The governing Equation (4.1) and Equation (4.2) must be associated with suitable initial and boundary conditions. The initial and boundary conditions are defined according to the following

$$\textit{Initial conditions} \quad \boldsymbol{u} = \boldsymbol{u}_0 \quad in \quad \Omega, \quad t = 0 \tag{4.5}$$

[1] The code is implemented by Dr. Jordi Cotela (Cotela [97])

$$\textit{Dirichlet Boundary} \quad \boldsymbol{u} = \boldsymbol{u}_D \quad in \quad \Gamma_D, \quad t \in [0, T[\tag{4.6}$$

$$\textit{Neumann Boundary} \quad \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} \quad in \quad \Gamma_N, \quad t = 0 \tag{4.7}$$

where $\boldsymbol{n}$ is the outer normal vector and $\boldsymbol{t}$ is the imposed traction acting along the Neumann boundary.

The weak form of the governing equations is required to obtain the finite element formulation. Equation (4.1) is multiplied by a velocity test function $\boldsymbol{w}$ and Equation (4.2) is multiplied by a pressure test function q. Then, the resulting expression is integrated over the domain Ω. The weak form of the governing equations is given by

$$\int_\Omega \boldsymbol{w} \cdot \left(\rho \partial_t \boldsymbol{u} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma}\right) d\Omega = \int_\Omega \boldsymbol{w} \cdot \boldsymbol{f} d\Omega \tag{4.8}$$

$$\int_\Omega q \nabla \cdot \boldsymbol{u} d\Omega = 0 \tag{4.9}$$

Further consideration of the complete equations for the weak form including the initial and boundary conditions can be found in Cotela [97]. Equation (4.8) and Equation (4.9) combined with initial conditions and weak form of the boundary conditions yield the weak form of the problem. All integrals must remain bounded. Thus, for any two given functions f, g we want to ensure that

$$\int_\Omega f g d\Omega < \inf \tag{4.10}$$

which are square integrable in Ω. The L^2 norm of a function is defined as

$$\|f\|_{L^2(\Omega)} = \left(\int_\Omega f^2 d\Omega\right)^{\frac{1}{2}} \tag{4.11}$$

The Hilbert space $H^1(\Omega)$ is used to verify the property of square-integrable functions for any given time instant t. The square-integrable property of a function along the problem time interval is given as $\boldsymbol{u} \in L^2\left(0, T, H^1_D(\Omega)\right)$. It is enough to enforce the L^2 norm for the pressure $p \in L^2\left(0, T, L^2(\Omega)\right)$.

For the convective term in the momentum equation, a skew-symmetric form is used as the conservation of kinetic energy is crucial in this implementation. Following the assumption of the non-conservative skew-symmetric convective term, the Galerkin weak form of the Navier-Stokes problem is stated as

Find $\boldsymbol{u} \in L^2\left(0, T, H^1_D(\Omega)\right)$, $p \in L^2\left(0, T, L^2(\Omega)\right)$ such that, $\forall \boldsymbol{w} \in H^1_0(\Omega)$, $\forall q \in L^2(\Omega)$,

$$\begin{aligned}
&\int_\Omega \boldsymbol{w}.\left(\rho\partial_t\boldsymbol{u} + \rho\frac{1}{2}\boldsymbol{u}.\nabla\boldsymbol{u}\right)d\Omega \\
&+\int_\Omega \nabla^s\boldsymbol{w} : 2\mu\left(\nabla^s\boldsymbol{u} - \frac{1}{3}(\nabla.\boldsymbol{u})\right)d\Omega - \int_\Omega \nabla.\boldsymbol{w}p\,d\Omega \\
&= \int_\Omega \boldsymbol{w}.\boldsymbol{f}\,d\Omega + \int_{\Gamma_N} \boldsymbol{w}.(t)\,d\Gamma
\end{aligned} \tag{4.12}$$

$$\int_\Omega q\nabla.\boldsymbol{u}\,d\Omega = 0 \tag{4.13}$$

The spatial discretization of this problem stated in Equation (4.12) and Equation (4.13) are not straightforward, because the discrete form is numerically unstable. This instability is necessarily true when the flow is dominated by the convection effects (nonlinear convective term) for high Reynolds number flows. A second complexity comes from the imposed compatibility of the velocity and pressure Finite Element discrete spaces by the $inf-sup$ or $Ladyzhenskaya-Babuska-Brezzi$ (LBB) condition. These instabilities can be avoided by using a stabilized formulation. The stabilized formulation has two advantages: first, a modified weak form not restricted by the LBB condition. Second, equal order interpolations can be used for velocity and pressure fields. In the scope of this work, the Variational Multi-scale (VMS) method is used and the concept is introduced in Hughes [123] and Hughes et al. [124]. The basic theory of this approach is the decomposition of the problem variables into large scale $(.)_h$ and small scale (sub-scale) $(.)_s$, which corresponds to :

$$\boldsymbol{u} = \boldsymbol{u}_h + \boldsymbol{u}_s \quad p = p_h + p_s \tag{4.14}$$

$$\boldsymbol{w} = \boldsymbol{w}_h + \boldsymbol{w}_s \quad q = q_h + q_s \tag{4.15}$$

where the scale separation in practice is closely related to the finite element mesh size h. Thus, test functions and solutions can be defined on a restricted, finite-dimensional space of admissible solutions, defined as

$$\boldsymbol{u}_h \in \boldsymbol{W}_h \subset \boldsymbol{W} \equiv L^2\left(0, T, H^1_D(\Omega)\right) \quad p_h \in Q_h \subset Q \equiv L^2\left(0, T, L^2(\Omega)\right) \tag{4.16}$$

which allows also a definition of spaces containing small scale part of the solution $\boldsymbol{w}_s \in \boldsymbol{W}_s$, $q_s \in Q_s$. Together with the definition of the large scale spaces, the complete problem space is given as: $\boldsymbol{W} = \boldsymbol{W}_h \bigoplus \boldsymbol{W}_s$, $Q = Q_h \bigoplus Q_s$. The complete derivation for the weak formulation of the VMS method and small scale equations can be found in Cotela [97]. The concept was initially introduced in Hughes [123] and further developed by Codina [118] and Codina [120]. The small scale model can be defined as Quasi-static or Dynamic small-scale models. The most general formulation can be given by

Momentum equation

$$\begin{aligned}
&\int_{\Omega} \boldsymbol{w}_h \cdot \rho \left(\partial_t \boldsymbol{u} + \frac{1}{2} \boldsymbol{a} . \nabla \boldsymbol{u}_h \right) d\Omega \\
&- \int_{\Omega} \nabla . \boldsymbol{w}_h p_h \, d\Omega + \int_{\Omega} \nabla^s \boldsymbol{w}_h : 2\mu \left(\nabla^s \boldsymbol{u}_h - \frac{1}{3} \left(\nabla . \boldsymbol{u}_h \right) \boldsymbol{I} \right) d\Omega \\
&+ \int_{\Omega} \boldsymbol{w}_h . \rho \partial_t \boldsymbol{u}_s \, d\Omega \\
&- \int_{\Sigma \Omega^e} \rho \left(\boldsymbol{a} . \nabla \boldsymbol{w}_h \right) \tau_t \left(\boldsymbol{R}^m \left(\boldsymbol{u}_h, p_h \right) - \boldsymbol{\xi}_h + \frac{\rho}{\delta t} \boldsymbol{u}_s^n \right) d\Omega \\
&- \int_{\Omega} \nabla . \boldsymbol{w}_h \tau_p \left(R^c \left(\boldsymbol{u}_h \right) - \delta_h \right) d\Omega \\
&= \int_{\Omega} \boldsymbol{w}_h \boldsymbol{f} \, d\Omega + \int_{\Gamma_N} \boldsymbol{w}_h \left(\boldsymbol{t} \right) d\Gamma
\end{aligned} \tag{4.17}$$

Mass conservation equation

$$\int_{\Omega} q_h \nabla . \boldsymbol{u}_h \, d\Omega = \int_{\Sigma \Omega^e} \nabla q_h \tau_t \left(\boldsymbol{R}^m \left(\boldsymbol{u}_h, p_h \right) - \boldsymbol{\xi}_h + \frac{\rho}{\delta t} \boldsymbol{u}_s^n \right) d\Omega \tag{4.18}$$

where:

$\boldsymbol{R}^m \left(\boldsymbol{u}_h, p_h \right)$ and $R^c \left(\boldsymbol{u}_h \right)$ represent the residual form of the Navier-Stokes equations applied to large scale variables.

$\boldsymbol{a} = \boldsymbol{u}_h + \boldsymbol{u}_s$ is a generic convection velocity used in nonlinear dependency of velocity.

$\boldsymbol{\xi}_h$ and δ_h are chosen to enforce the right hand side of momentum equation.

τ_t and τ_p are the stabilization parameters.

For the Quasi-static algebraic sub-grid scale (ASGS), the parameters $\boldsymbol{\xi}_h$ and δ_h are equal to zero. While for Quasi-static orthogonal sub-grid scale (OSS), $\boldsymbol{\xi}_h$ and δ_h are calculated as following:

$$\boldsymbol{\xi}_h = \prod_{V_h} \left(\boldsymbol{R}^m \left(\boldsymbol{u}_h, p_h \right) \right) \tag{4.19}$$

$$\delta_h = \prod_{Q_h} \left(R^c \left(\boldsymbol{u}_h \right) \right) \tag{4.20}$$

where $\prod_{V_h}$ and $\prod_{Q_h}$ are the L^2 projection onto the large scale spaces W_h and Q_h respectively. The approximated expression for the residuals are defined in the form:

$$\rho \partial_t \boldsymbol{u}_s + \frac{1}{\boldsymbol{\tau}_u} \boldsymbol{u}_s \approx \boldsymbol{R}^m \left(\boldsymbol{u}_h, p_h \right) - \boldsymbol{\xi}_h \qquad \frac{1}{\tau_p} p_s \approx R^c \left(\boldsymbol{u}_h \right) - \delta_h \tag{4.21}$$

Where the velocity second order tensor $\boldsymbol{\tau}_u = \tau_u \boldsymbol{I}$ is taken to be a diagonal matrix. The Quasi-static small scale definition for the stabilization parameters is defined as

$$\tau_u = \left(\frac{c_1 \mu}{h^2} + \frac{c_2 \rho \|\boldsymbol{a}\|}{h} \right)^{-1} \tag{4.22}$$

$$\tau_p = \frac{h^2}{c_1 \tau_u} = \mu + \rho \frac{c_2 \|\boldsymbol{a}\| h}{c_1} \tag{4.23}$$

where:

h : is an element characteristic length.

c_1 and c_2 are constants based on the flow properties and used to be 8 and 2 respectively[2]).

In Quasi-static small scales, an instability can appear for small time steps once the problem is discretized in time. The instability can be avoided if the stabilization parameter satisfies the condition

$$\delta t \geq C \tau_u \tag{4.24}$$

where δt is the time step and C is a constant. Thus, the stabilization parameter τ_u can be replaced by the expression

$$\tau_t = \left(\frac{\rho}{\delta t} + \frac{c_1 \mu}{h^2} + \frac{c_2 \rho \|\boldsymbol{a}\|}{h} \right)^{-1} \tag{4.25}$$

[2] These constants are set by the code developer (Cotela [97]

Equation (4.25) shows a dependency of the solution on the time step, even for problems that result in a stationary solution. Moreover, the Dynamic small-scales, provides solution stability, is out of the scope of this work.

4.2 Discretization

A discrete form of Equation (4.17) and Equation (4.18) is needed to solve for the velocity and pressure in space and time. Discretization in space is performed using P_1 $Lagrange$ finite elements and $WBZ-\alpha\,(BOSSAK)$ is used for the temporal discretization. The theoretical background of the implementation is adapted from Cotela [97].

4.2.1 Spatial Discretization

Defining a discrete domain Ω_h for the problem domain Ω, the standard finite element interpolation functions are used to define the large scale interpolation spaces V_h and Q_h. Thus, finite element interpolation can be used to compute the large scale velocity and pressure terms as

$$\boldsymbol{u}_h = \sum_a^{n_n} \boldsymbol{N}_a(\boldsymbol{x})\,\boldsymbol{u}_a \qquad p_h = \sum_a^{n_n} N_a(\boldsymbol{x})\,p_a \tag{4.26}$$

where:

n_n : is the number of nodes in a finite element mesh.

$\boldsymbol{u}_a$ and p_a are the nodal values for $\boldsymbol{u}_h$ and p_h respectively.

$\boldsymbol{N}_a$ and N_a are nodal standard finite element basis function.

Moreover, $\boldsymbol{U}$, $\dot{\boldsymbol{U}}$, and $\boldsymbol{P}$ are vectors of nodal values for velocity $\boldsymbol{u}_h$, acceleration $\partial_t \boldsymbol{u}_h$, and pressure p_h, respectively. The matrix form of the problem is produced by introducing the FEM discretization Equation (4.26) into the governing Equation (4.17) and Equation (4.18). In this implementation, linear shape functions are used for the finite element discretization. Therefore, terms with high order derivatives can not be evaluated. Consequently, some information is lost in computing the viscous stresses which is considered minimal in turbulent flow problems dominated by convection.

4.2.1.1 Quasi-static ASGS formulation

The quasi-static algebraic sub-grid scale (QS-ASGS) is considered the classical form of the Navier-Stokes equations. The terms $\boldsymbol{\xi}_h$ and $\boldsymbol{\delta}_h$ are equal to zero, thus neglected in Equation (4.17) and Equation (4.18). Moreover, terms involving $\boldsymbol{u}_s^n$ or $\partial_t \boldsymbol{u}_s$ can be ignored when using Equation (4.21) for describing the small scales. τ_u is used as the stabilization parameter. The discrete governing equations based on Equation (4.26) is given by Cotela [97]

$$\begin{aligned}&\left[\boldsymbol{M}+\boldsymbol{S}_m\left(\tau_u,\boldsymbol{a}\right)\right]\dot{\boldsymbol{U}}+\left[\boldsymbol{C}(\boldsymbol{a})+\boldsymbol{K}+\boldsymbol{S}_u\left(\tau_u,\boldsymbol{a}\right)+\boldsymbol{H}_u\left(\tau_p\right)\right]\boldsymbol{U}\\&+\left[\boldsymbol{G}+\boldsymbol{S}_p\left(\tau_u,\boldsymbol{a}\right)\right]\boldsymbol{P}=\boldsymbol{F}+\boldsymbol{T}+\boldsymbol{S}_f\left(\tau_u,\boldsymbol{a}\right)\end{aligned} \tag{4.27}$$

$$\boldsymbol{Q}_m\left(\tau_u\right)\dot{\boldsymbol{U}}+\left[\boldsymbol{D}+\boldsymbol{Q}_u\left(\tau_u,\boldsymbol{a}\right)\right]\boldsymbol{U}+\boldsymbol{Q}_p\left(\tau_u\right)\boldsymbol{P}=\boldsymbol{Q}_f\left(\tau_u\right)\boldsymbol{P}=\boldsymbol{Q}_f\left(\tau_u\right) \tag{4.28}$$

The discrete version of Equation (4.17) and Equation (4.18) are given by Equation (4.27) and Equation (4.28). Having a and b as the local node indices for a N finite element nodes and $\boldsymbol{A}^e$ as the elemental matrix for $\boldsymbol{A}$. The variational form of the problem elemental matrices are defined as

$$\boldsymbol{M}_{ab}^e=\int_{\Omega_e}\rho\boldsymbol{N}_a^T\boldsymbol{N}_b\,d\Omega \tag{4.29}$$

$$\boldsymbol{C}(\boldsymbol{a})_{ab}^e=\int_{\Omega_e}\frac{1}{2}\rho\left(\boldsymbol{N}_a^T\boldsymbol{a}.\nabla\boldsymbol{N}_b\right)d\Omega+\int_{\Gamma_N}\boldsymbol{N}_a^T\frac{1}{2}\rho\left(\boldsymbol{a}.\boldsymbol{n}\right)\boldsymbol{N}_b\,d\Gamma \tag{4.30}$$

$$\boldsymbol{K}_{ab}^e=\int_{\Omega_e}\boldsymbol{B}_a^T\boldsymbol{C}_\mu\boldsymbol{B}_b\,d\Omega \tag{4.31}$$

$$\boldsymbol{G}_{ab}^e=-\int_{\Omega_e}\left(\nabla.\boldsymbol{N}_a\right)^T N_b\,d\Omega \tag{4.32}$$

$$\boldsymbol{D}_{ab}^{e} = \int_{\Omega_e} N_a \nabla . \boldsymbol{N}_b \, d\Omega = -\left(\boldsymbol{G}_{ba}^{e}\right)^T \tag{4.33}$$

$$\boldsymbol{F}_{a}^{e} = \int_{\Omega_e} \boldsymbol{N}_a^T \boldsymbol{f} \, d\Omega \tag{4.34}$$

$$\boldsymbol{T}_{a}^{e} = \int_{\Gamma_N} \boldsymbol{N}_a^T \boldsymbol{t} \, d\Gamma \tag{4.35}$$

Where the constitutive matrix $\boldsymbol{C}_\mu$ in viscous matrix Equation (4.31) is given by

$$\boldsymbol{C}_\mu = \begin{bmatrix} 4\mu/3 & -2\mu/3 & -2\mu/3 & 0 & 0 & 0 \\ -2\mu/3 & 4\mu/3 & -2\mu/3 & 0 & 0 & 0 \\ -2\mu/3 & -2\mu/3 & 4\mu/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{4.36}$$

The stabilization terms are defined as

$$\boldsymbol{S}_m\left(\tau_u, \boldsymbol{a}\right)_{ab}^{e} = \int_{\Omega_e} \left(\rho \boldsymbol{a} . \nabla \boldsymbol{N}_a\right)^T \tau_u \boldsymbol{N}_b \, d\Omega \tag{4.37}$$

$$\boldsymbol{S}_u\left(\tau_u, \boldsymbol{a}\right)_{ab}^{e} = \int_{\Omega_e} \left(\rho \boldsymbol{a} . \nabla \boldsymbol{N}_a\right)^T \tau_u \rho \boldsymbol{a} . \nabla \boldsymbol{N}_b \, d\Omega \tag{4.38}$$

$$\boldsymbol{S}_p\left(\tau_u, \boldsymbol{a}\right)_{ab}^{e} = \int_{\Omega_e} \left(\rho \boldsymbol{a} . \nabla \boldsymbol{N}_a\right)^T \tau_u \nabla N_b \, d\Omega \tag{4.39}$$

$$\boldsymbol{S}_f\left(\tau_u,\boldsymbol{a}\right)_{ab}^e=\int_{\Omega_e}\left(\rho\boldsymbol{a}.\nabla\boldsymbol{N}_a\right)^T\tau_u\boldsymbol{f}d\Omega \tag{4.40}$$

$$\boldsymbol{Q}_m\left(\tau_u\right)_{ab}^e=\int_{\Omega_e}\left(\nabla\boldsymbol{N}_a\right)^T\tau_u\boldsymbol{N}_b d\Omega \tag{4.41}$$

$$\boldsymbol{Q}_u\left(\tau_u,\boldsymbol{a}\right)_{ab}^e=\int_{\Omega_e}\left(\nabla\boldsymbol{N}_a\right)^T\tau_u\rho\boldsymbol{a}.\nabla\boldsymbol{N}_b d\Omega=\left(\boldsymbol{S}_p\left(\tau_u,\boldsymbol{a}\right)_{ba}^e\right)^T \tag{4.42}$$

$$\boldsymbol{Q}_p\left(\tau_u\right)_{ab}^e=\int_{\Omega_e}\left(\nabla\boldsymbol{N}_a\right)^T\tau_u\nabla\boldsymbol{N}_b d\Omega \tag{4.43}$$

$$\boldsymbol{Q}_f\left(\tau_u\right)_{ab}^e=\int_{\Omega_e}\left(\nabla\boldsymbol{N}_a\right)^T\tau_u\boldsymbol{f}d\Omega \tag{4.44}$$

$$\boldsymbol{H}_u\left(\tau_p\right)_{ab}^e=\int_{\Omega_e}\left(\nabla.\boldsymbol{N}_a\right)^T\tau_p\nabla.\boldsymbol{N}_b d\Omega \tag{4.45}$$

4.2.1.2 Quasi-static OSS formulation

The second implementation for the VMS element is the quasi-static orthogonal sub-grid scale (QS-OSS). In contrast to the QS-ASGS, the QS-OSS includes the calculation of the projections $\boldsymbol{\xi}_h$ and $\boldsymbol{\delta}_h$ which make the small scale variables orthogonal to the large scale variables. The QS-OSS will reduce the numerical diffusion introduced in the problem. The matrix form of the QS-OSS formulation can be expressed as (Cotela [97])

$$\begin{gathered}\left[\boldsymbol{M}+\boldsymbol{S}_m\left(\tau_u,\boldsymbol{a}\right)\right]\dot{\boldsymbol{U}}+\left[\boldsymbol{C}(\boldsymbol{a})+\boldsymbol{K}+\boldsymbol{S}_u\left(\tau_u,\boldsymbol{a}\right)+\boldsymbol{H}_u\left(\tau_p\right)\right]\boldsymbol{U}\\+\left[\boldsymbol{G}+\boldsymbol{S}_p\left(\tau_u,\boldsymbol{a}\right)\right]\boldsymbol{P}=\boldsymbol{F}+\boldsymbol{T}+\boldsymbol{S}_f\left(\tau_u,\boldsymbol{a}\right)-\boldsymbol{S}_{\prod}\left(\tau_u,\boldsymbol{a}\right)-\boldsymbol{H}_{\prod}\left(\tau_u\right)\end{gathered} \tag{4.46}$$

$$\boldsymbol{Q}_m(\tau_u)\dot{\boldsymbol{U}} + \left[\boldsymbol{D} + \boldsymbol{Q}_u(\tau_u, \boldsymbol{a})\right]\boldsymbol{U} + \boldsymbol{Q}_p(\tau_u)\boldsymbol{P} = \boldsymbol{Q}_f(\tau_u) - \boldsymbol{Q}_{\prod}(\tau_u) \tag{4.47}$$

where the new terms involving the projections are defined as

$$\boldsymbol{S}_{\prod}(\tau_u, \boldsymbol{a})_a^e = \int_{\Omega_e} (\rho \boldsymbol{a}.\nabla \boldsymbol{N}_a)^T .\tau_u \boldsymbol{\xi}_h d\Omega \tag{4.48}$$

$$\boldsymbol{Q}_{\prod}(\tau_u)_a^e = \int_{\Omega_e} (\nabla N_a)^T .\tau_u \boldsymbol{\xi}_h d\Omega \tag{4.49}$$

$$\boldsymbol{H}_{\prod}(\tau_p)_a^e = \int_{\Omega_e} (\nabla.\boldsymbol{N}_a)^T \tau_p \boldsymbol{\delta}_h d\Omega \tag{4.50}$$

The computation of the projections involves an additional problem solution as shown in Equation (4.21) which is defined in the discrete form as

$$\boldsymbol{M}_\xi \Xi = \boldsymbol{R}_\xi \tag{4.51}$$

$$\boldsymbol{M}_\delta \Delta = \boldsymbol{R}_\xi \tag{4.52}$$

where Ξ and Δ represent the nodal values for $\boldsymbol{\xi}_h$ and δ_h, respectively. The nodal values are given by

$$\boldsymbol{M}_{\xi ab}^e = \int_{\Omega_e} \boldsymbol{N}_a^T \boldsymbol{N}_b d\Omega \tag{4.53}$$

$$\boldsymbol{M}_{\delta ab}^e = \int_{\Omega_e} N_a N_b d\Omega \tag{4.54}$$

$$\boldsymbol{R}_{\xi a}^e = \int_{\Omega_e} \boldsymbol{N}_a^T .\boldsymbol{R}^m(\boldsymbol{u}_h, p_h) d\Omega \tag{4.55}$$

$$\boldsymbol{R}^{e}_{\delta ab} = \int_{\Omega_e} N_a R^c \left(\boldsymbol{u}_h\right) d\Omega \tag{4.56}$$

Finally, Equation (4.51) and Equation (4.52) are solved together with Equation (4.46) and Equation (4.47) which doubles the number of nodal unknowns in the problem, but a simplified version is used based on Equation (4.53) and Equation (4.54) by a diagonal mass matrix. This simplification is used to approximate the projections instead of solving an additional system.

4.2.2 Temporal Discretization

The system of equations can be expressed in the form shown in Equation (4.57), after discretization in space.

$$\tilde{\boldsymbol{M}} \begin{bmatrix} \dot{\boldsymbol{U}} \\ 0 \end{bmatrix} + \tilde{\boldsymbol{C}} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{P} \end{bmatrix} = \tilde{\boldsymbol{F}} \tag{4.57}$$

Two temporal discretization schemes are used in this implementation $BDF2$ and $WBZ-\alpha\,(BOSSAK)$. In this section the $WBZ-\alpha\,(BOSSAK)$ will be introduced, because it is the default scheme used in the simulation code. The $WBZ-\alpha\,(BOSSAK)$ is derived from the $generalized-\alpha$ Newmark group of methods. More information about these methods can be found in Cotela [97], Wood et al. [125], and Arnold et al. [126]. The basic expression of the Newmark scheme is written in terms of velocities as the main problem's variable as

$$\dot{u}_{n+1} = \frac{1}{\gamma_N \Delta t}\left(u^{n+1} - u^n\right) - \left(\frac{1}{\gamma_N} - 1\right)\dot{u}^n \tag{4.58}$$

where γ_N is a constant parameter. The $BOSSAK$ scheme adds a relaxation factor on the acceleration of the system as Cotela [97]

$$\left(1-\alpha_B\right)\tilde{\boldsymbol{M}} \begin{bmatrix} \dot{\boldsymbol{U}}^{n+1} \\ 0 \end{bmatrix} + \alpha_B \tilde{\boldsymbol{M}} \begin{bmatrix} \dot{\boldsymbol{U}}^{n} \\ 0 \end{bmatrix} + \tilde{\boldsymbol{C}} \begin{bmatrix} \boldsymbol{U}^{n+1} \\ \boldsymbol{P}^{n+1} \end{bmatrix} = \tilde{\boldsymbol{F}} \tag{4.59}$$

Equation (4.58) is used to discretize Equation (4.59) which yields the following time-discrete problem

$$\begin{aligned}&\left(\frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}+\tilde{\boldsymbol{C}}\right)\begin{bmatrix}\boldsymbol{U}^{n+1}\\ \boldsymbol{P}^{n+1}\end{bmatrix}=\\ &\tilde{\boldsymbol{F}}-\frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}\begin{bmatrix}\boldsymbol{U}^{n}\\ 0\end{bmatrix}+\{(1-\alpha_B)\left(\frac{1}{\gamma_N}-1\right)+\alpha_B\}\tilde{\boldsymbol{M}}\begin{bmatrix}\dot{\boldsymbol{U}}^{n}\\ 0\end{bmatrix}\end{aligned} \tag{4.60}$$

The $BOSSAK$ parameter is $\alpha_B = -0.3$ together with $\gamma_N = {}^1\!/\!_2 - \alpha_B$ to ensure a second order convergence and stability of the scheme. Equation (4.60) is non-linear and the problem residual at time step $n+1$ after i non-linear iterations as

$$\begin{aligned}&\boldsymbol{R}\left(\boldsymbol{U}^{n+1,i},\boldsymbol{P}^{n+1,i}\right)=\tilde{\boldsymbol{F}}-\frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}\begin{bmatrix}\boldsymbol{U}^{n}\\ 0\end{bmatrix}\\ &+\{(1-\alpha_B)\left(\frac{1}{\gamma_N}-1\right)+\alpha_B\}\tilde{\boldsymbol{M}}\begin{bmatrix}\dot{\boldsymbol{U}}^{n}\\ 0\end{bmatrix}-\left(\frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}+\tilde{\boldsymbol{C}}\right)\begin{bmatrix}\boldsymbol{U}^{n+1,i}\\ \boldsymbol{P}^{n+1,i}\end{bmatrix}\end{aligned} \tag{4.61}$$

By using a first order Taylor decomposition to evaluate the zero of Equation (4.61) having $\boldsymbol{R}^{n+1,i+1}=0$ and an iteration increment $\boldsymbol{\delta U}^i = \boldsymbol{U}^{n+1,i+1} - \boldsymbol{U}^{n+1,i}$ to find $\boldsymbol{U}^{n+1,i+1}$, $\boldsymbol{P}^{n+1,i+1}$ as

$$\begin{aligned}&\boldsymbol{R}\left(\boldsymbol{U}^{n+1,i+1},\boldsymbol{P}^{n+1,i+1}\right)=\\ &\boldsymbol{R}\left(\boldsymbol{U}^{n+1,i},\boldsymbol{P}^{n+1,i}\right)+\frac{\partial \boldsymbol{R}\left(\boldsymbol{U}^{n+1,i+1},\boldsymbol{P}^{n+1,i+1}\right)}{\partial\left(\boldsymbol{\delta U}^i,\boldsymbol{\delta P}^i\right)}\begin{bmatrix}\boldsymbol{\delta U}^{i}\\ \boldsymbol{\delta P}^{i}\end{bmatrix}=0\end{aligned} \tag{4.62}$$

The last known variables are used to evaluate all matrices and vectors. Thus, the system matrix can be defined as

$$\frac{\partial \boldsymbol{R}\left(\boldsymbol{U}^{n+1,i+1},\boldsymbol{P}^{n+1,i+1}\right)}{\partial\left(\boldsymbol{\delta U}^i,\boldsymbol{\delta P}^i\right)}\Big|_i \approx \frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}+\tilde{\boldsymbol{C}} \tag{4.63}$$

Finally, the linear system of Equation (4.64) is solved iteratively until the residual vector $\boldsymbol{R}\left(\boldsymbol{U}^{n+1,i}, \boldsymbol{P}^{n+1,i}\right)$ are smaller than a predefined tolerance.

$$-\left(\frac{1-\alpha_B}{\gamma_N \Delta t}\tilde{\boldsymbol{M}}+\tilde{\boldsymbol{C}}\right)\begin{bmatrix}\delta \boldsymbol{U}^i \\ \delta \boldsymbol{P}^i\end{bmatrix}=\boldsymbol{R}\left(\boldsymbol{U}^{n+1,i}, \boldsymbol{P}^{n+1,i}\right) \tag{4.64}$$

4.2.3 Formal Order of Convergence

The code has a second-order formal convergence rate in space and time, since a P_1 $Lagrange$ finite elements are used in space and $WBZ-\alpha$ $(BOSSAK)$ in time. Space and time discretization schemes are of the same order of convergence and formally second order, as stated in Hanzlicek [122] and Wood et al. [125], which indicates that the time scheme is of second-order convergence. Finally, the P_1 $Lagrange$ finite element represents a second-order convergence scheme.

More than the act of testing,
the act of designing tests is
one of the best bug
preventers known.

Boris Beizer

CHAPTER 5

Code Verification: Application to KRATOS Multiphysics

Code verification is performed based on the guidelines that are introduced in Section 3.2. In the scope of this work, a series of benchmarks are developed to verify unsteady, eddy-resolving Navier-Stokes and applied to a FEM-based CFD application of *KRATOS Multiphysics*, introduced in Chapter 4. The code verification benchmarks are designed to be applicable to any arbitrary CFD code solving similar problem. The code verification activities performed are based on the Taylor-Green Vortex. In the context of this work only 2D Navier-Stokes equations for an incompressible Newtonian fluid are considered. The testing activities have an increasing complexity starting from an equal contribution of all the terms in the Navier-Stokes (N-S) equations to a term-dominating solution.

5.1 KRATOS Multiphysics Tested Functionalities

In the course of this work, KRATOS Multiphysics FEM based solver, initiated at CIMNE, is to be verified. After analyzing the discrete equations shown in Chapter 4, the code is claimed to have a second-order convergence in space and time. The actual order of convergence of the following functionalities is to be verified:

Finite Element Method

ALE Formulation

Unsteady using WBZ-α (Bossak)

Unsteady using BDF2

Incompressible Fluid

Newtonian fluid

VMS Monolithic Solver

Orthogonal Subscales Stabilization

While verifying this code, four main points are to be addressed:

1. The solution of a convection-dominated flow is always problematic in FEM-based codes, which imposes some instability in the solution. Thus, a stabilized form of the weak form is used.

2. The use of linear finite elements leads to limitations in only obtaining first-order derivatives, and adds some error to the computation of the viscous stresses.

3. The same space-time formal order of convergence discretization schemes simplifies the verification of unsteady solutions.

4. The conditions imposed on the stabilization parameters (Chapter 4) have a dependency on time step size, and also the constants, that affect the stability of the solution.

5.2 Code Verification Benchmark Workflow

The following benchmark series is developed based on the Taylor-Green vortex to test each term in the Navier-Stokes Equations. This is done to explore which term affects the code order of convergence and needs more investigation in case of unclear convergence. The approach gives the term of interest very high weight compared to others in the N-S equations. Thus, the error generated from this term is the dominating error. The workflow is described below:

1. Creation of Taylor-Green vortex exact solution using a symbolic math programming (i.e MUPAD in MATLAB)
2. Application of the source terms (if exist), boundary conditions and initial conditions to the code
3. Performing the simulations
4. Evaluating the error
5. Repeating with systematic refinement
6. Calculation of the observed order of accuracy
7. The observed order of convergence matches the theoretical order of convergence in some scenarios
8. Modify the Taylor-Green vortex manufactured solutions to assess each term in the N-S equations
9. Repeating steps 2 to 6
10. If the observed order of convergence does not match the theoretical order of convergence
11. Checking whether the test case is correctly implemented
12. If step 11 is correct, finding and correcting the errors in the code. Then, repeating testing.
13. Documenting results

5.3 Calculation of the Error and Order of Convergence

The numerical solution consists of values of the dependent variables on a set of discrete locations. To compute the discretization error, two error norms are used. Error norms can be used to determine the global error of a field ϕ in its spatial domain K. An E_2 error norm for the variable ϕ compared to the exact solution $\hat{\phi}$ in the domain **K** is defined in Equation (5.1) as

$$E_2 = ||\phi - \hat{\phi}||^2 = \sqrt{\frac{1}{K}\int_K (\phi - \hat{\phi})^2 dK} \tag{5.1}$$

The E_2 norm is chosen to have a representative mean field error of the complete domain (Fisch [9]). Moreover, the infinity (inf) norm returns the maximum absolute error over the entire domain given by Equation (5.2). Therefore, it is the most sensitive error measure, and is very proper to detect local discontinuities or singularities.

$$E_i = max|\phi_n - \hat{\phi_n}|, \quad n \in [1, N] \tag{5.2}$$

After error evaluation and simulating over several meshes, the information is used to estimate the observed order of convergence. The observed order of convergence is estimated using the following expression:

$$p = \frac{\log\left(\frac{E(h_{coarse})}{E(h_{fine})}\right)}{\log(r)} \tag{5.3}$$

5.4 Spatial and Temporal Resolutions

KRATOS Multiphysics uses triangular or quadrilateral elements in which meshes can be designed in four ways as shown in Figure 5.1:

- Structured Mesh
- Unidirectional Structured Mesh

- Alternate Diagonal Structured Mesh
- Unstructured Mesh

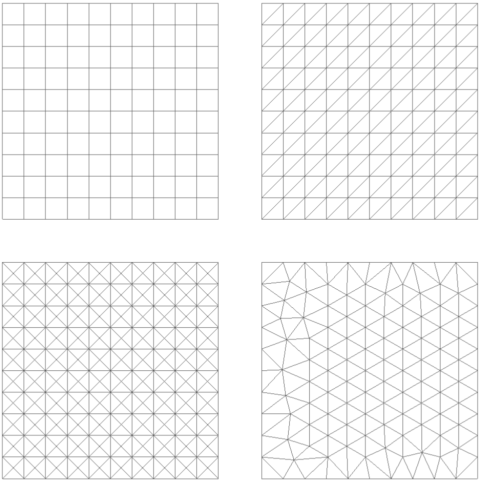


Figure 5.1: Different mesh types: top-left represents structured mesh, top-right is unidirectional structured triangular mesh, bottom-left is alternate diagonal structured mesh, and bottom-right is unstructured mesh

In the scope of the results presented below, structured and unidirectional structured meshes are tested. Both Reynolds number and Courant number are considered while deciding on the spatial and temporal resolution. CFL number is kept below one for all the simulations. The refinement ratio in both space and time is two, which complies with the code's theoretical order of convergence.

5.5 Design of Test Cases

5.5.1 Evaluation of Source Terms

The governing equations are the N-S equations introduced in Section 4.1. The manufactured solutions $(\hat{u}_x, \hat{u}_y, \hat{Pr})$ are used together with governing equations (Equation (4.1) and Equation (4.2)) to drive the equilibrium sources $\hat{\boldsymbol{s}}_f$ as

$$\hat{\boldsymbol{s}}_f^{mom} = \underbrace{\frac{\partial \hat{\boldsymbol{u}}}{\partial t}}_{\text{Inertia}} + \underbrace{\nabla \,.\, (\hat{\boldsymbol{u}}\hat{\boldsymbol{u}})}_{\text{Convection}} - \underbrace{\nabla \,.\, (2\nu\hat{\boldsymbol{D}})}_{\text{Viscous}} + \underbrace{\nabla \hat{Pr}}_{\text{Pressure}} \tag{5.4}$$

$$\hat{\boldsymbol{s}}_f^{mass} = \nabla \,.\, \hat{\boldsymbol{u}} \tag{5.5}$$

where $\hat{\boldsymbol{s}}_f^{mom}$ is the source term for the momentum conservation and $\hat{\boldsymbol{s}}_f^{mass}$ is the source term for the mass conservation.

5.5.2 Boundary and Initial Conditions

Dirichlet boundary condition is used for both velocity and pressure terms on all the boundaries because our target is the verification of the interior equations. Thus any boundary errors must be avoided. More rigorous testing can be performed for different types of BC and a combination of BC.

5.5.3 Solution Accuracy

The discretization error should be isolated from the total numerical error. Therefore, the used software is using a double-precision accuracy to minimize the round-off error. The iterative solver convergence criteria in a non-normalized version of the E_2 norm of velocity and pressure fields are set to 10^{-10} to keep the incomplete iterative convergence error (*IICE*) as small as possible. Otherwise, a direct solver is used for solving the linear system of equations; thus, the IICE will not exist.

5.5.4 Simulation Setup

Table 5.1 summarizes the simulation setup parameters.

Table 5.1: Simulation parameters used for code verification

Element:	VMS Monolithic Solver
Linear System Solver:	Super LU "Direct Solver"
Stabilization:	ASGS if not otherwise specified
Domain Size:	$X_1 = x \in [0, 2\pi]$
	$X_2 = y \in [0, 2\pi]$
	$t \in [0, 10]$

5.5.5 Taylor-Green Vortex (TGV)

The Taylor-Green vortex is an exact closed-form solution of the incompressible N-S equations (DeBonis [48], Fauconnier et al. [127], Rees et al. [128], Brachet et al. [129], Bhatt et al. [130], Fauconnier et al. [131], Mastellone et al. [132], and Bull et al. [133]). It is an unsteady flow of a decaying vortex. The solution is a periodic array of vortices that repeats itself in two Cartesian dimensions. The general form of the Taylor-Green Vortex is defined as follows:

$$\hat{u}_x = u_0 f(t) \sin(kx) \cos(ky) \tag{5.6}$$

$$\hat{u}_y = -u_0 f(t) \cos(kx) \sin(ky) \tag{5.7}$$

$$\hat{P}r = \frac{\rho u_0^2}{4} f(t)^2 (\cos(2kx) + \sin(2ky)) \tag{5.8}$$

Where:

$$f(t) = e^{-2\nu k^2 t}; \ \nu \rightarrow \text{kinematic viscosity} \tag{5.9}$$

$$k = \frac{2\pi}{L}; L \rightarrow \text{Periodic Length}; \rho \rightarrow \text{Density} \tag{5.10}$$

$$\textit{Reynolds number: } Re = \frac{L u_0}{\nu} \;\; \textit{Courant number: } CFL = \frac{u_0 \Delta t}{\Delta x} \tag{5.11}$$

The source terms $\hat{\mathbf{s}}_f^{mom}$ and $\hat{\mathbf{s}}_f^{mass}$ are evaluated using Equation (5.4) and Equation (5.5), respectively. $\hat{\mathbf{s}}_f^{mom}$ and $\hat{\mathbf{s}}_f^{mass}$ are found to be zero for the TGV, which defines the uniqueness of this benchmark. Therefore, the TGV can be used to test any code.

The rationale for choosing a TGV as a first test case for KRATOS Multiphysics VMS monolithic solver is two-fold. Firstly, the TGV is a well-established reference in the literature. Secondly, the physics of the flow field and the ease of BC constitute an excellent benchmark for eddy-resolving implementations. Three simulations are performed using TGV for different refinement strategies, as listed in Table 5.2 (TGV1, TGV2, and TGV3).

Table 5.2: Base Benchmark: Taylor Green Vortex

Material: $\nu = 0.2$, $\rho = 1.0$		
Test Case	Fields	Discretization
TGV1	Using the general form of the Taylor-Green Vortex with parameters: $k = 1.0$ $u_0 = 1.0$	Space (cells/direction): $2^{2 \xrightarrow{increment\ 1} 8}$ $\Delta t = 0.001$
TGV2		Space (cells/direction): 2^8 $\Delta t = 2^{7 \xrightarrow{decrement\ 1} 1}.10^{-3}$
TGV3		Space (cells/direction): $2^{2 \xrightarrow{increment\ 1} 8}$ $\Delta t = 2^{7 \xrightarrow{decrement\ 1} 1}.10^{-3}$

For TGV1, spatial refinement is performed, having a very fine time step. The reason is to minimize the temporal discretization error and test the spatial discretization. For TGV2, temporal refinement is performed, having a very fine spatial grid to minimize space discretization error and test the temporal discretization. Finally, for TGV3, both space and time are refined simultaneously to check the discretization in space and time. The third approach, TGV3, is valid because both space and time discretization have a theoretical second-order convergence. In this case of having equal orders, convergence studies can be performed for both space and time simultaneously.

In this benchmark, all the terms in the N-S equations have equal weights. The TGV, shown in Figure 5.2, constitutes the simplest flow for which a turbulent energy cascade can be observed numerically. Figure 5.2 shows the relation between the contribution of each term in Equation (5.4) to $\hat{\boldsymbol{s}}_f^{mom}$ at $X_2 = \pi$ and $t = 0.0$. The TGV is defined in that way to assure that the discretization errors produced by the different terms of the governing equations have the same magnitude and can, therefore, be captured by order of accuracy test.

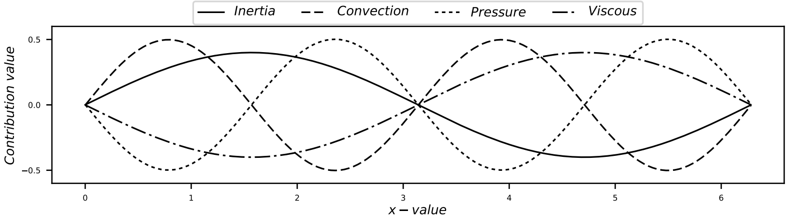


Figure 5.2: Taylor-Green Vortex functions contribution at $X_2 = y = \pi$ and $t = 0.0$

The TGV benchmark is performed for two mesh types, unidirectional triangular grid and structured quadrilateral grid. Then results are presented in Section 5.5.5.1 and Section 5.5.5.2, respectively.

5.5.5.1 Uni-Directional Triangular Grid

Using a unidirectional triangular grid (see Figure 5.1), the results produced by KRATOS Multiphysics for test cases TGV1, TGV2 and TVG3 are shown in Figure 5.3, Figure 5.4 and Figure 5.5, respectively. The left part of the named figures shows the log-log plot of the error norms of the variable fields over-refinement. The right part shows the observed order of accuracy (p) with refinement.

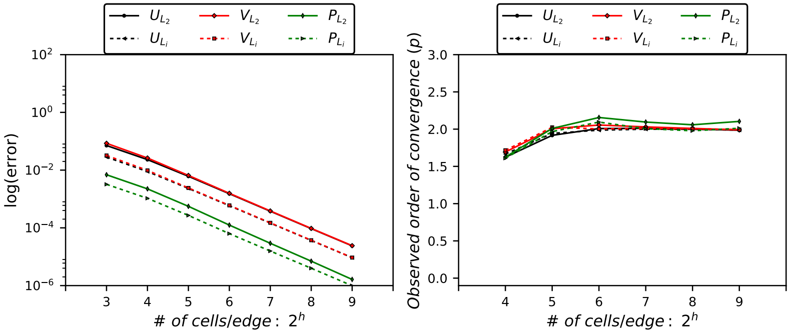


Figure 5.3: TGV1 benchmark for unidirectional triangular grid having $\Delta t = 0.001\ s$ and mesh is refined from 2^2 $cells/edge$ to 2^8 $cells/edge$

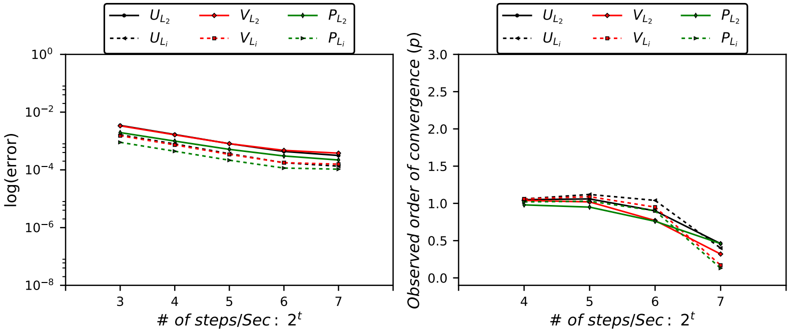


Figure 5.4: TGV2 benchmark for unidirectional triangular grid having $\Delta x = \Delta y = 2^8$ $cells/edge$ and time step size is refined from $\Delta t = 2^7.10^{-3}\ s$ to $2^1.10^{-3}\ s$

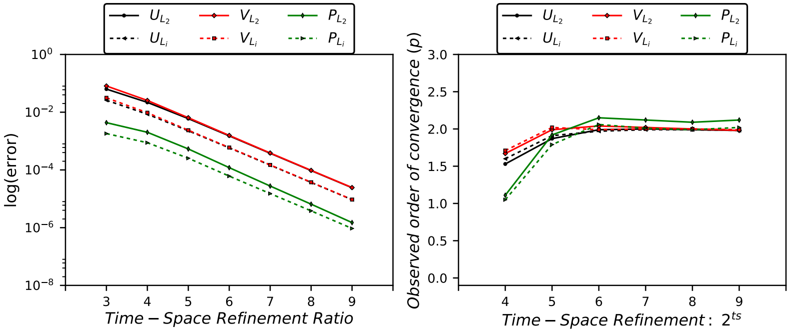


Figure 5.5: TGV3 benchmark for unidirectional triangular grid having mesh refined from 2^2 $cells/edge$ to 2^8 $cells/edge$ and time step size is refined from $\Delta t = 2^7.10^{-3}$ s to $2^1.10^{-3}$ s, simultaneously

The results shown in Figure 5.3 show that for a constant time step size ($\Delta t = 0.001$ s), the code has a second-order convergence with spatial refinement. In contrast, Figure 5.4 shows code failure in achieving second-order convergence, having constant mesh size $\Delta x = \Delta y = 2^8$ $cells/edge$ and temporal refinement. Moreover, Figure 5.5 shows a second-order convergence for space-time refinement, but the first segment of the curve has a first-order convergence for the pressure in both E_2 norm and E_i norm. As can be seen from the figures, a second-order accuracy is observed, which complies with the theoretical order of accuracy. Except for the TGV2, shown in Figure 5.4, the theoretical order of convergence can not be achieved. Therefore, more testing is required to debug the code using benchmark cases.

5.5.5.2 Structured Quadrilateral Grid

In this section, code verification activities are used to verify a new implementation in KRATOS Multiphysics. The VMS monolithic solver has been improved further to not only deal with triangular and tetrahedral elements but also quads and hexahedral meshes in FEM environment. Thus, the new implementation is tested to assure the correctness of the code implementation. A demonstrative example of a structured quadrilateral grid is presented in Figure 5.1.

First, the TGV1 benchmark is performed, and results are shown in Figure 5.6. It can be observed that the code fails when a space refinement study is performed, although it was successful for the unidirectional triangular grid. This implies a coding bug in the new implementation.

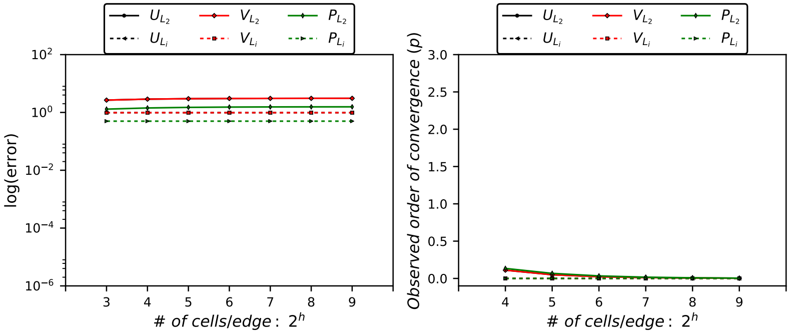


Figure 5.6: TGV1 benchmark for structured quadrilateral grid having $\Delta t = 0.001\ s$ and mesh is refined from 2^2 cells/edge to 2^8 cells/edge

The results are communicated with the code developer. A bug is found in the initialization of the solver. The bug is fixed, and the test is performed again. After fixing the bug, the code performs as expected, see Figure 5.7, and the observed order of convergence matches the formal order of convergence. Thus, it can be safely said that the code has a second order of convergence for the spatial discretization scheme, except for the first three segments of the L_i error norm curve for the pressure.

Since the test case is unsteady, the order of convergence of the time integration scheme is to be verified. The TGV3 benchmark is used to verify the coupling of the space and time discretization schemes. Figure 5.8 shows the test results. The results show a clear second-order convergence for the time-space discretization coupling, except for the first two segments of the L_i error norm curve for the pressure.

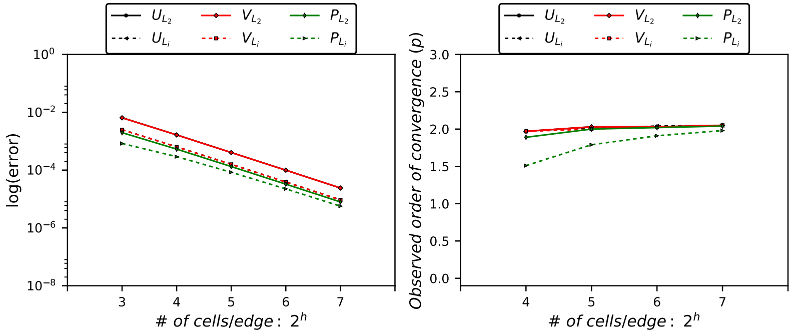


Figure 5.7: TGV1 benchmark for structured quadrilateral grid having $\varDelta t = 0.001\ s$ and mesh is refined from 2^2 cells/edge to 2^8 cells/edge: bug fixed

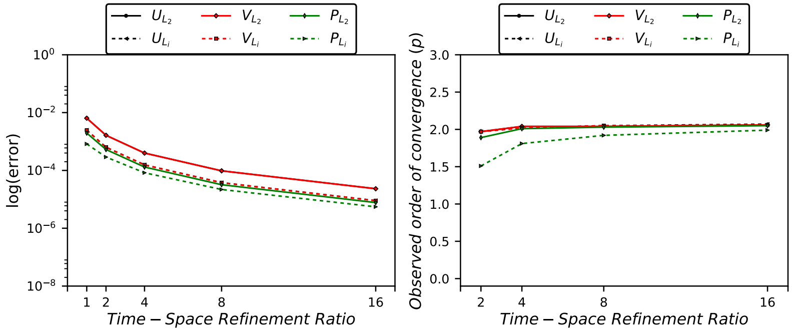


Figure 5.8: TGV3 benchmark for structured quadrilateral grid: mesh refined from 2^2 cells/edge to 2^8 cells/edge and time step size is refined from $\varDelta t = 2^7.10^{-3}\ s$ to $2^1.10^{-3}\ s$, simultaneously

The above tests do not evaluate only the time discretization scheme. Thus, the TGV2 benchmark is performed to test the effect of time refinement only. The results are shown in Figure 5.9. The order of convergence plot shows that the code fails to pass this test, which was also concluded in Section 5.5.5.1. This failure might happen for two reasons. First, the mesh is not fine enough in space, making the spatial discretization errors dominate the solution. Thus, the refinement in time has no effect. Second, a problem exists in the time discretization scheme used. Finally, Section 5.5.6 presents a more sophisticated benchmark, which is developed to test the performance of the temporal discretization scheme.

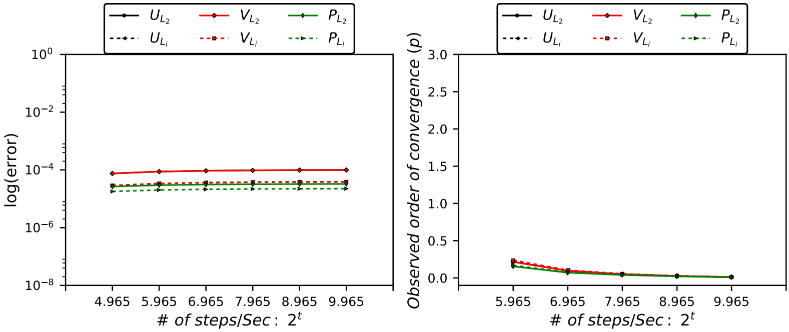


Figure 5.9: TGV2 benchmark for structured quadrilateral grid: $\Delta x = \Delta y = 2^8$ cells/edge and time step size is refined from $\Delta t = 2^7.10^{-3}\ s$ to $2^1.10^{-3}\ s$

5.5.6 Inertia Dominated Simulation

From Section 5.5.5, it can be concluded that the code performance is acceptable, but TGV2 showed a deviation from the expected performance. For time refinement only, the code performance is expected to have a second-order convergence, which was not achieved in the TGV2 benchmark for both triangular and quadrilateral grids. Consequently, this case study, shown in Table 5.3, is designed only to test the inertial term in the N-S equations.

Table 5.3: Benchmark: Inertia dominated flow

	Material: $\nu = 0.1$ $\rho = 1.0$	
Test Case	Fields	Discretization
IN1	$\hat{u}_x = \sin(t)$ $\hat{u}_y = -\sin(t)$ $\hat{P}r = \sin(t)$	Space (cells/direction): $2^{2 \xrightarrow{increment\ 1} 8}$ $\Delta t = 0.001$
IN2		Space (cells/direction): 2^8 $\Delta t = 2^{6 \xrightarrow{decrement\ 1} 1}.10^{-3}$
IN3	$\hat{u}_x = \sin(2t)\cos(y)$ $\hat{u}_y = -\sin(2t)\cos(x)$ $\hat{P}r =$ $\sin(2t)(\cos(2x)+\cos(2y))$	Space (cells/direction): $2^{2 \xrightarrow{increment\ 1} 8}$ $\Delta t = 0.001$
IN4		Space (cells/direction): $2^{2 \xrightarrow{increment\ 1} 7}$ $\Delta t = 2^{7 \xrightarrow{decrement\ 2} 1}.10^{-3}$

The benchmark design has two approaches: Firstly, the manufactured solution fields are only time-dependent functions as in cases IN1 and IN2. Secondly, the manufactured solution fields have time and space dependency in cases IN3 and IN4. In both cases, the time derivative term of the unsteady N-S equations is dominating. The contributions to $\hat{\boldsymbol{s}}_f^{mom}$, of the different terms of the governing equation, are computed from Equation (5.4). Figure 5.10 shows the evaluation of the different terms at $X_1 = X_2 = \pi$ for $t \in [0, 10]$. The benchmark satisfies the condition that Contributions from all terms are ≈ 0, except for the inertia term, as shown in Figure 5.10. Cases IN1 and IN2 are purely time-dependent, but IN3 and IN4 benchmarks have a spatial dependency. IN3 and IN4 are specially designed for code that might experience instability from having spatial derivatives equal to zero.

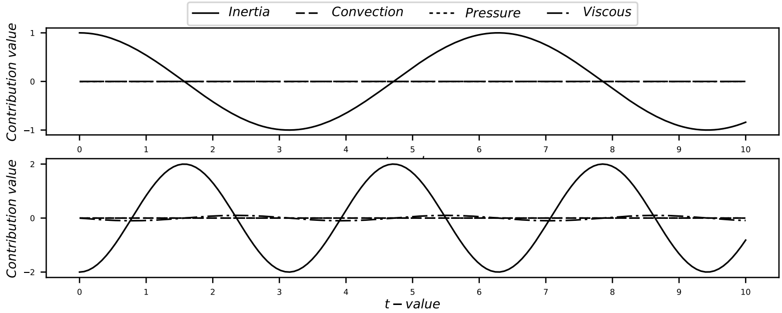


Figure 5.10: Inertia dominated functions upper figure for In1 and IN2, and lower figure for IN3 and IN4

5.5.6.1 Uni-Directional Triangular Grid

The code's performance for a time-dependent solution with test cases IN1 to IN4 is examined. Figure 5.11 and Figure 5.12 show that spatial refinement does not affect solution quality, which confirms the assumption that the solution is time-dependent.

Consequently, case IN2 is performed, where the simulation is performed with a time refinement at the finest mesh. Figure 5.13 shows that a second-order of accuracy is observed for the pressure field. Whereas for the velocity field, the observed order of accuracy is between 1.5 and 2 for some simulations. Also, the order of convergence for the velocity fields U and V using L_2 and L_i error norms decreases for smaller time steps. The performance of the code is not as expected for a time-only dependent solution. Consequently, IN4 was performed having a space-time refinement. Figure 5.14 shows an oscillating convergence behavior around two for both velocity and pressure fields. The first segment of Figure 5.14 shows a first-order convergence for the velocity fields and second order for the pressure fields, while the last segment of the curve shows a declining convergence order with refinement.

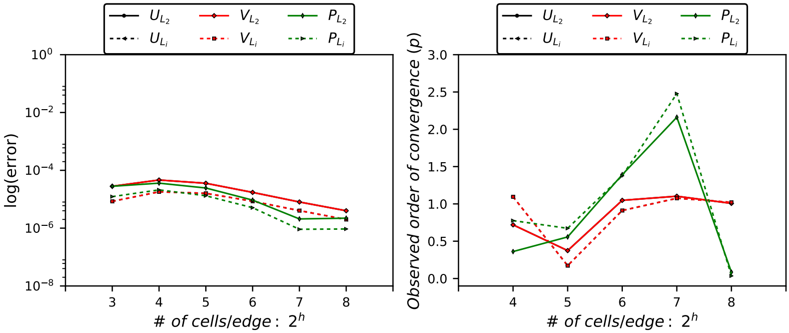


Figure 5.11: IN1 benchmark for unidirectional triangular grid having $\Delta t = 0.001\ s$ and mesh is refined from 2^2 cells/edge to 2^8 cells/edge

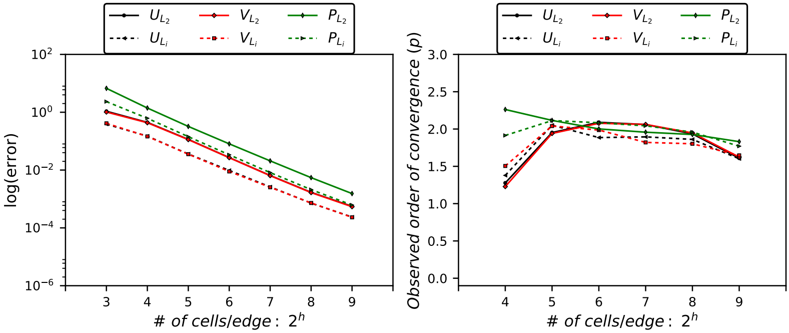


Figure 5.12: IN3 benchmark for unidirectional triangular grid having $\Delta t = 0.001\ s$ and mesh is refined from 2^2 cells/edge to 2^8 cells/edge

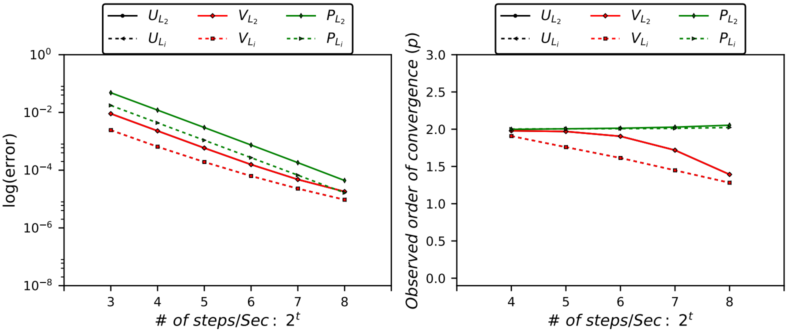


Figure 5.13: IN2 benchmark for unidirectional triangular grid having $\Delta x = \Delta y = 2^8$ cells/edge and time step size is refined from $\Delta t = 2^6.10^{-3}$ s to $2^1.10^{-3}$ s

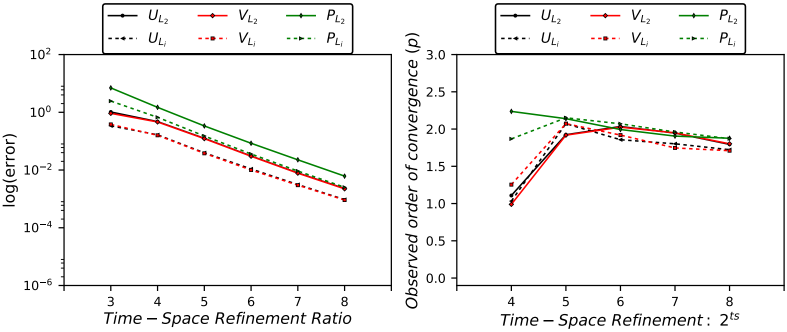


Figure 5.14: IN4 benchmark for unidirectional triangular grid having mesh refined from 2^8 cells/edge to 2^2 cells/edge and time step size is refined from $\Delta t = 2^7.10^{-3}$ s to $2^1.10^{-3}$ s, simultaneously

From cases IN1, IN2, IN3, and IN4, it can be concluded that there is some sort of instability in KRATOS Multiphysics VMS formulation for cases of only time-dependent solutions or an implementation issue with the temporal discretization scheme. Overall, IN2 benchmark is shown to be the most critical case to test the performance of the temporal discretization scheme. The code performance is deeply investigated in Section 5.5.6.2 using the IN2 benchmark.

5.5.6.2 Structured Quadrilateral Grid

The benchmark IN2 is used for the time integration scheme verification. The IN2 is shown in Table 5.3. At this stage, it is very important to clarify that the code supports two time integration schemes $WBZ-\alpha\,(Bossak)$ and $BDF2$. Both schemes are of second-order convergence, but the $WBZ-\alpha\ (Bossak)$ is the default scheme. Also, there exist two options for the stabilization formulation, ASGS (Section 4.2.1.1) and OSS (Section 4.2.1.2). ASGS is the default scheme. Therefore, the verification study starts with testing the convergence order for the ASGS with the $WBZ-\alpha\,(Bossak)$ scheme.

Figure 5.15 shows that there is no clear convergence order in the scheme. The convergence order for all the fields is decreasing with refinement. This indicates a problem in the implementation of the Bossak time integration scheme. To debug the reason for this behavior, a series of test cases with different parameters is performed. Several reasons are considered for this problem. The first reason is the definition of the α parameter used in the $WBZ-\alpha\,(Bossak)$. Therefore, the α is set to zero to treat the scheme as a $generalized-\alpha$ (Bajer [134]).

The IN2 benchmark is performed using the ASGS stabilization scheme and the $generalized-\alpha$ for time discretization scheme. Figure 5.16 shows that setting α to zero does not improve the order of convergence of the problem where the pressure field is not affected at all, which is expected, and the velocity components have a first order of convergence. Another test is performed with the default values for the parameter α in the $WBZ-\alpha\,(Bossak)$ integration scheme, but the OSS approach is used for the stabilization of the problem.

The IN2 benchmark is performed using the OSS stabilization scheme and the $WBZ-\alpha\ (Bossak)$ for time discretization. Figure 5.17 shows that

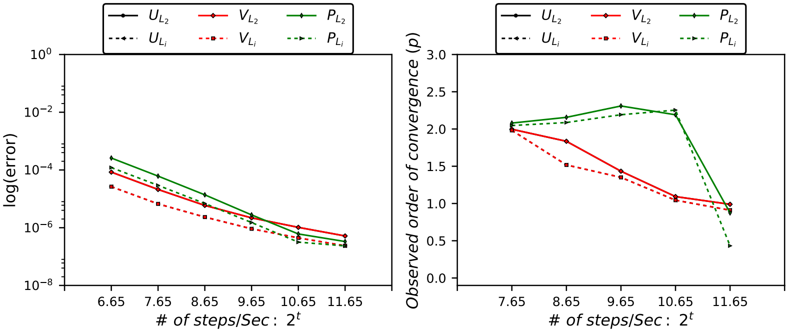


Figure 5.15: IN2 benchmark with time refinement using ASGS and $WBZ-\alpha\,(Bossak)$ on structured quadrilateral grid

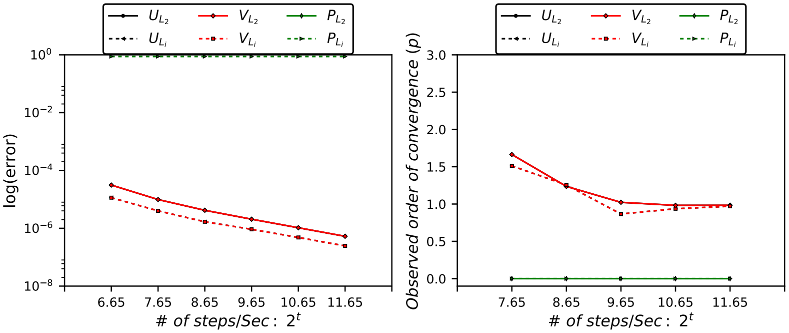


Figure 5.16: IN2 benchmark with time refinement using $generalized-\alpha$ and ASGS on structured quadrilateral grid

the OSS has an improvement over the ASGS stabilization scheme in the pressure field, where the sudden loss of accuracy at the end does not exist. Then, from observing the error development in Figure 5.15 and Figure 5.17, it is evident that the error improves in an almost second-order form for the first couple of time step sizes till a certain limit, then the convergence drops to a first-order or lower. This indicates a consistency issue or instability in the solution. The assumption of solution instability is weak since we measure the error in the L_i norm, which has a similar behavior to the euclidean norm L_2. When communicating the results with the code developer, it was said that the time step size affects the value of the stabilization parameters, also the $WBZ-\alpha\,(Bossak)$ is linearized for the ease of application. The effect of the time step size on the performance of the $WBZ-\alpha\,(Bossak)$ can be tracked in two ways: (1) using different time discretization scheme to verify it is a problem of the discretization scheme and not the stabilization scheme and (2) study the effect of varying the time step size from very coarse to very fine.

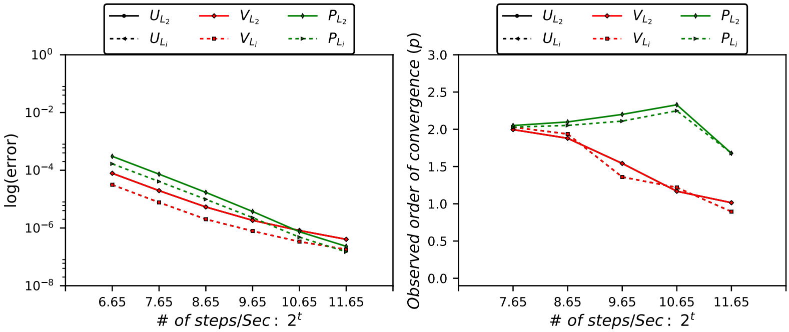


Figure 5.17: IN2 benchmark with time refinement using OSS and $WBZ-\alpha\,(Bossak)$ on structured quadrilateral grid

Therefore, another test case is performed using different time discretization scheme. The $BDF2$ time integration scheme is tested for both ASGS and OSS formulations. The results are shown in Figure 5.18 and Figure 5.19 for the ASGS and OSS approaches, respectively. It can be clearly seen that the $BDF2$ has a second order of convergence and matches the theoretical order of convergence. Thus, the $BDF2$ time integration scheme is verified.

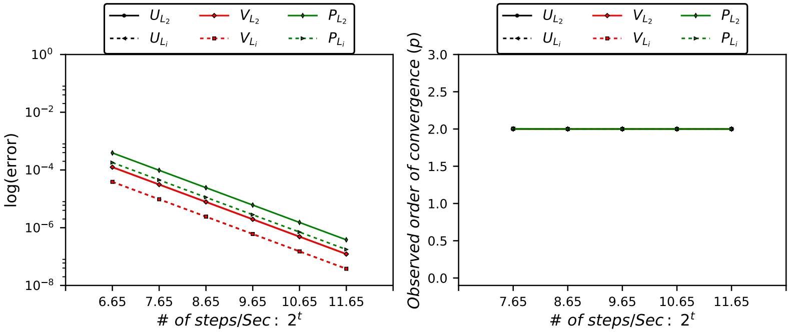


Figure 5.18: IN2 benchmark with time refinement using $BDF2$ and ASGS on structured quadrilateral grid

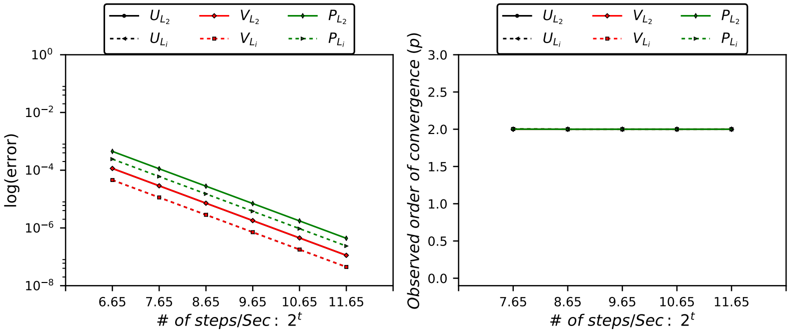


Figure 5.19: IN2 benchmark with time refinement, using $BDF2$ and OSS, for a structured quadrilateral grid

Figure 5.18 and Figure 5.19 verify the implementation of the ASGS and OSS formulations, while highlight the problem emerging from the linearization of the $WBZ-\alpha\ (Bossak)$ scheme. The $WBZ-\alpha\ (Bossak)$ scheme is verified using coarser time steps. Figure 5.20 and Figure 5.21 show that the observed order of convergence matches the theoretical order of convergence, and the code can be said to be verified. Thus, it can be concluded that the Bossak time integration scheme has a second order of convergence. Some inconsistencies in the convergence behavior are noticed when a very fine time step is used, because of the dependency of the stabilization terms on the time step size. Also, a similar behavior is observed when using different alternatives to the Bossak scheme for ASGS and OSS stabilization. The $generalized-\alpha$ scheme results in Figure 5.22 show a second-order of convergence for the velocity fields, and a zero convergence for the pressure field, as expected. The same conclusion can be drawn for a different $\alpha-value$, where $\alpha=-0.1$, as shown in Figure 5.23.

Finally, it can be concluded that the $WBZ-\alpha\ (Bossak)$ scheme has a conditional second order convergence with an accuracy limitation, as can be seen in the error plots in Figure 5.20, Figure 5.21, Figure 5.22 and Figure 5.23, based on the time step size.

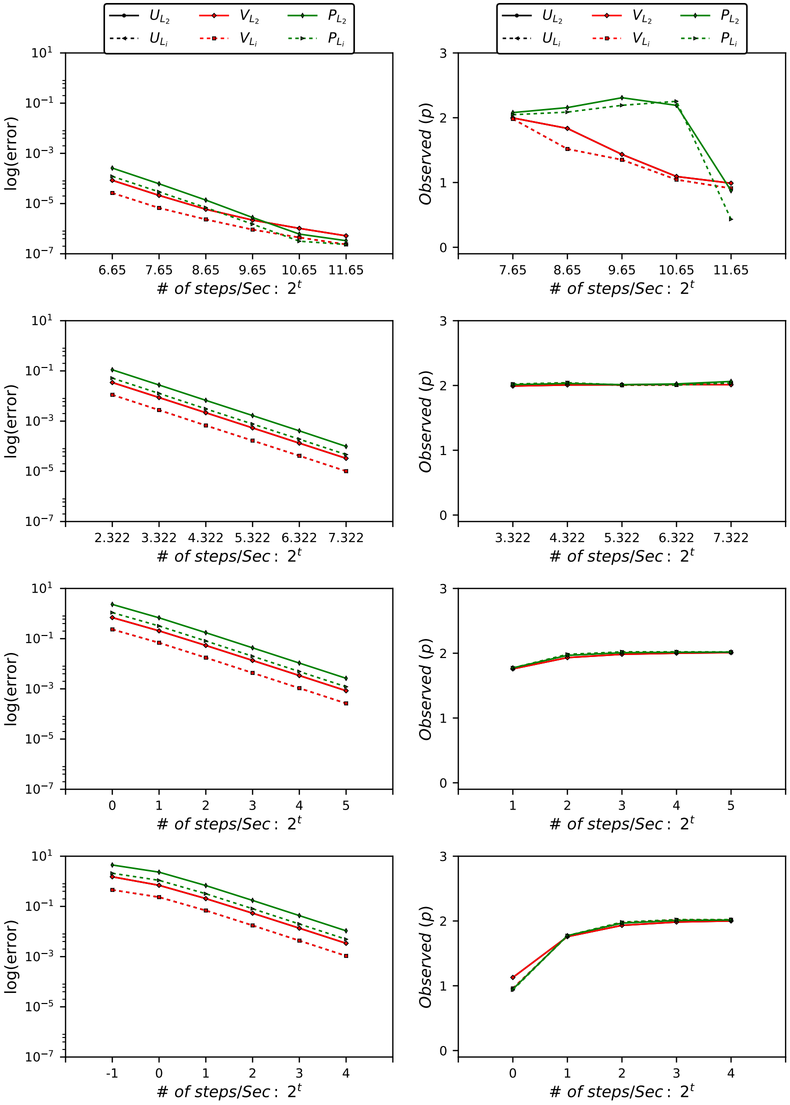


Figure 5.20: IN2 benchmark with coarse time step refinement using ASGS and $WBZ{-}\alpha\,(Bossak)$ on structured quadrilateral grid

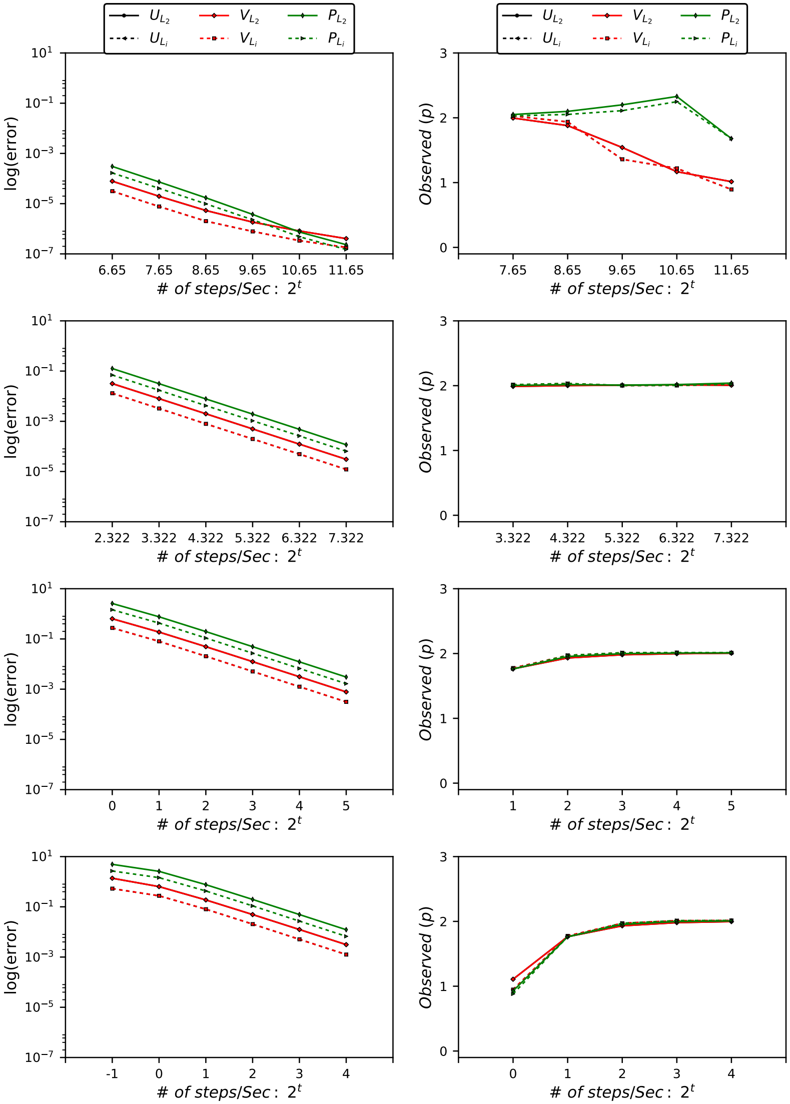


Figure 5.21: IN2 benchmark with coarse time step refinement using OSS and $WBZ-\alpha$ $(Bossak)$ on structured quadrilateral grid

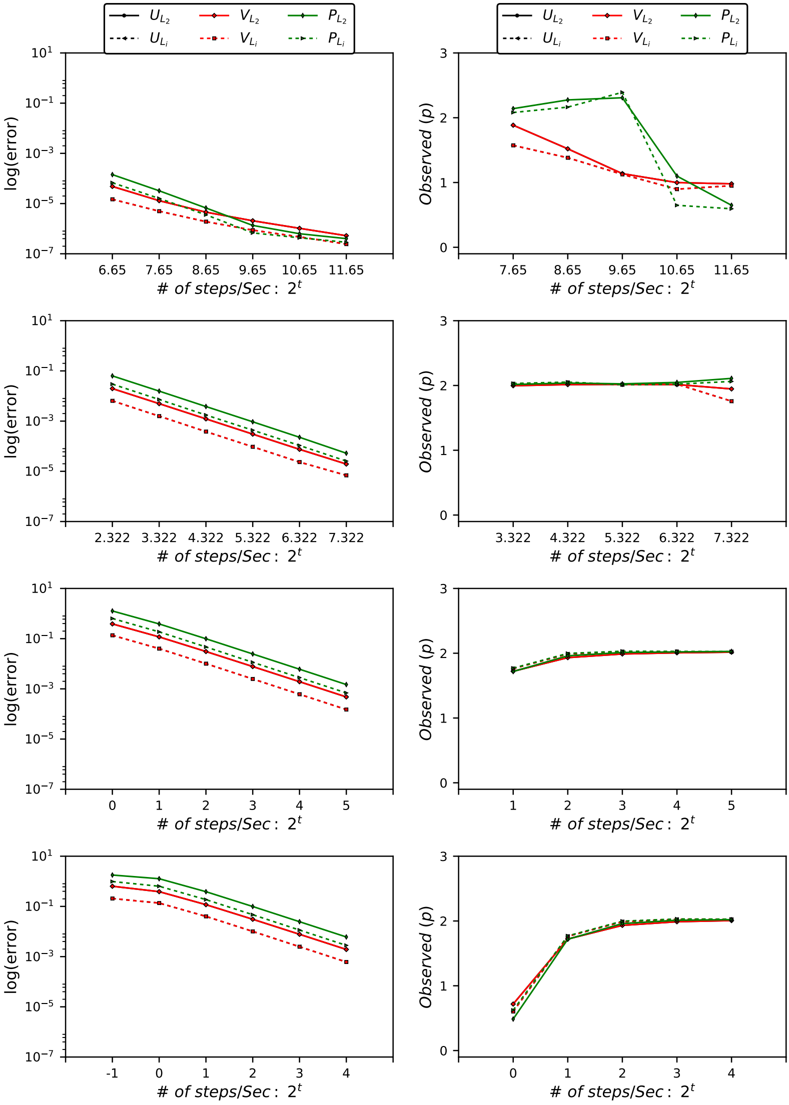


Figure 5.22: IN2 benchmark with coarse time step refinement using ASGS and generalized−α on structured quadrilateral grid

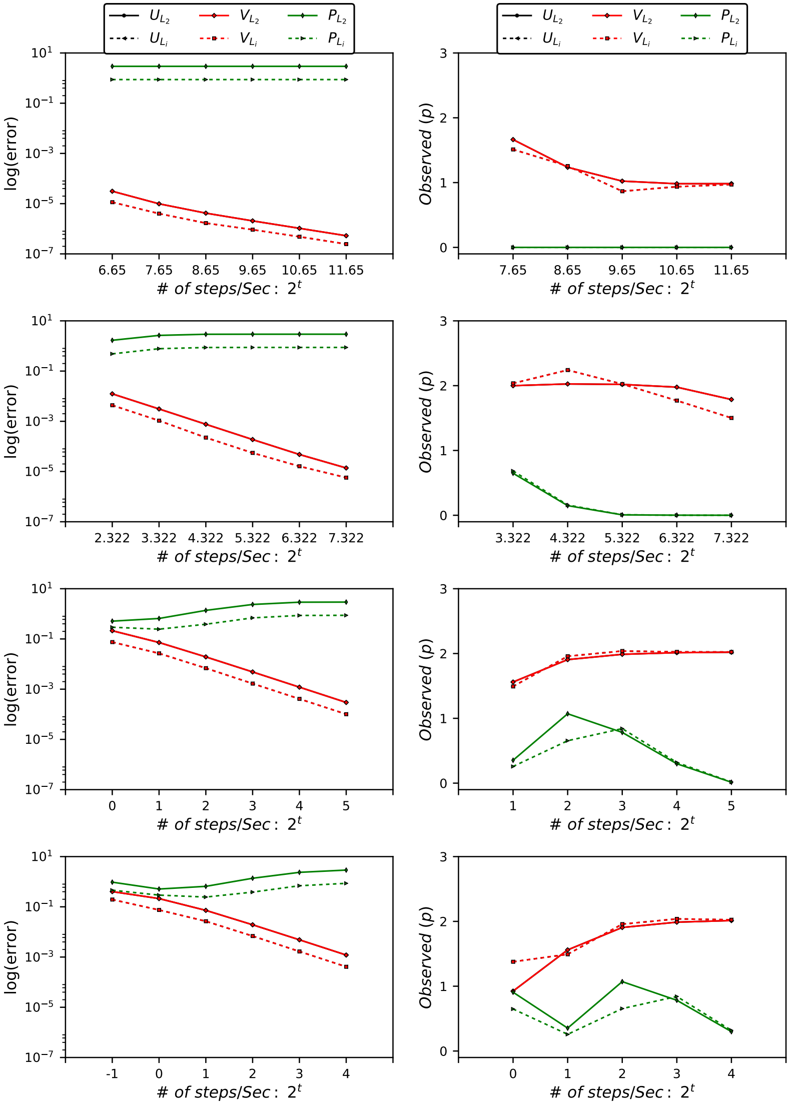


Figure 5.23: IN2 benchmark with coarse time step refinement using ASGS and Newmark–$\alpha = -0.1$ on structured quadrilateral grid

5.5.7 Pressure Dominated Simulations

The purpose of this simulation is to numerically check the code's accuracy if the pressure term in the N-S equations is dominating the flow. Unidirectional triangular grid and $WBZ - \alpha$ $(Bossak)$ time discretization scheme are used to perform the benchmark. PRES and PRES1 benchmarks are defined in Table 5.4. Then, the contribution of the different terms contributing to $\hat{\mathbf{s}}_f^{mom}$ are computed and presented in Figure 5.24. The highest contribution to the $\hat{\mathbf{s}}_f^{mom}$ is resulting from the pressure term as shown in Figure 5.24. In this study, two simulations are performed, PRES and PRES1.

Table 5.4: Benchmark: Pressure dominated flow

Material: $\nu = 0.1$ $\rho = 1.0$		
Test Case	Fields	Discretization
PRES	$\hat{u}_x = e^{-0.2t}\sin(x)\cos(y)$ $\hat{u}_y = -e^{-0.2t}\cos(x)\sin(y)$ $\hat{Pr} =$ $e^{-0.4t}(\cos(2x) + \cos(2y))$	Space (cells/direction): $2^{2\xrightarrow{increment\ 1}8}$ $\Delta t = 0.001$
PRES1		Space (cells/direction): $2^{2\xrightarrow{increment\ 1}7}$ $\Delta t = 2^{7\xrightarrow{decrement\ 2}1}.10^{-3}$

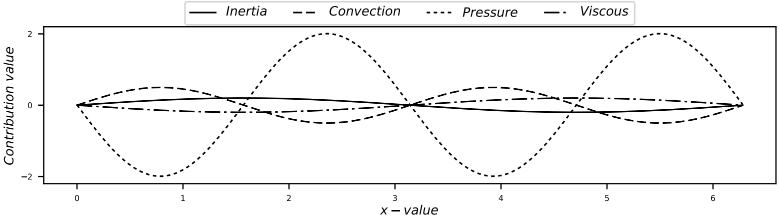


Figure 5.24: Pressure dominated functions at $X_2 = y = \pi$ and $t = 0.0$

PRES simulation is done with spatial refinement. PRES1 is performed with space-time refinement, as indicated in Table 5.4. It can be concluded that the observed order of accuracy in both cases is nearly second-order, as shown in Figure 5.25 and Figure 5.26.

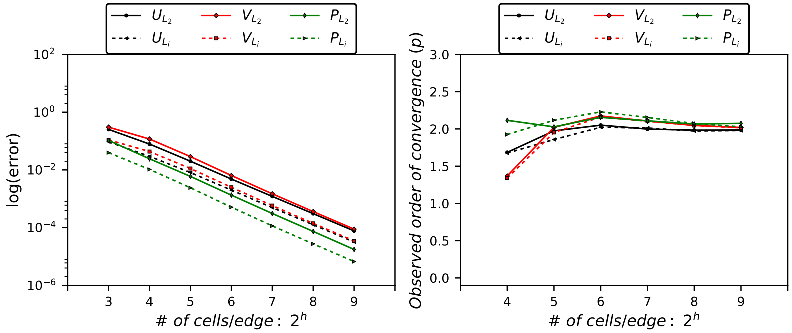


Figure 5.25: PRES benchmark for unidirectional triangular grid having $\Delta\ t = 0.001\ s$ and mesh is refined from 2^2 cells/edge to 2^8 cells/edge

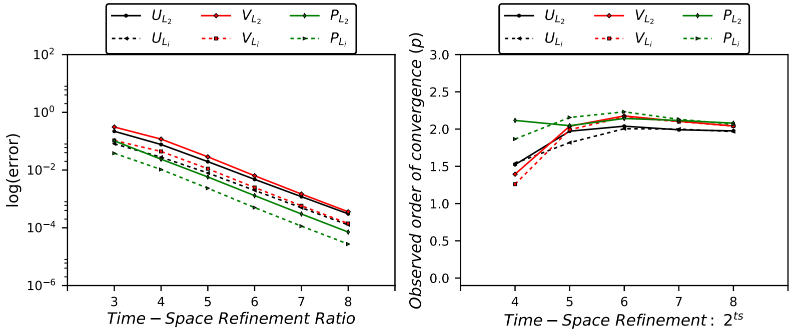


Figure 5.26: PRES1 benchmark for unidirectional triangular grid having mesh refined from $2^2\ cells/edge$ to $2^7\ cells/edge$ and time step size is refined from $\Delta t = 2^7.10^{-3}\ s$ to $2^1.10^{-3}\ s$, simultaneously

5.5.8 Convection Dominated Simulations

The purpose of this simulation is to numerically check the code's accuracy if the convection term in the N-S equations is dominating the flow. Unidirectional triangular grid and $WBZ-\alpha\,(Bossak)$ time discretization scheme are used to perform the benchmark. CON benchmark is defined in Table 5.5. Figure 5.27 shows the contribution of different terms of the governing equation to $\hat{\mathbf{s}}_f^{mom}$. From the previous cases, it can be concluded that for the KRATOS Multiphysics code, spatial and temporal discretizations are correlated in some sense, and solution accuracy is more sensitive to mesh refinement than time refinement. Consequently, only mesh refinement simulations are performed for this benchmark.

Table 5.5: Benchmark: Convection dominated flow

	Material: $\nu = 0.05\ \rho = 1.5$	
Test Case	Fields	Discretization
CON	$\hat{u}_x = e^{-0.1t}\sin(x)\cos(y)$ $\hat{u}_y = -e^{-0.1t}\cos(x)\sin(y)$ $\hat{Pr} = \frac{3}{80}e^{-0.2t}(\cos(2x)+\cos(2y))$	Space (cells/direction): $2^{2\xrightarrow{increment\ 1}8}$ $\Delta t = 0.001$

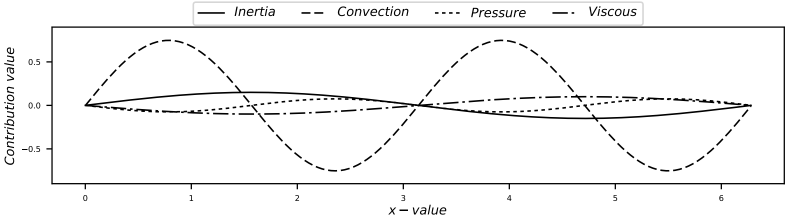


Figure 5.27: Convection dominated functions at $X_2 = y = \pi$ and $t = 0.0$

Figure 5.28 shows the code performance having a convective flow problem. It can be concluded that the observed order of accuracy is nearly second-order, which matches the theoretical order of convergence. Thus, the code is verified.

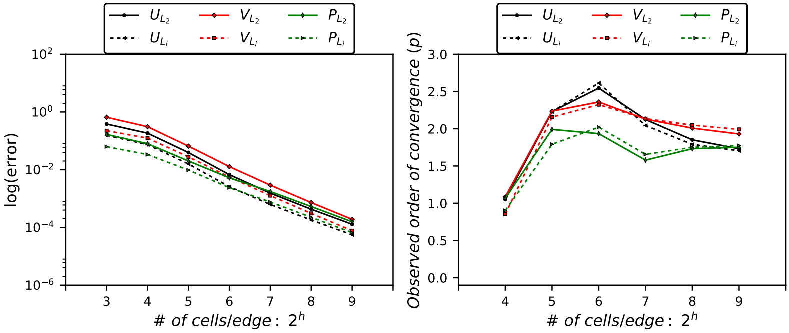


Figure 5.28: Convection dominated flow spatial refinement unidirectional triangular grid

5.5.9 Viscous Dominated Simulations

The purpose of this simulation is to numerically check the code's accuracy if the viscous term in the N-S equations is dominating the flow. Unidirectional triangular grid and $WBZ-\alpha\,(Bossak)$ time discretization scheme are used to perform the benchmark. VIS benchmark is defined in Table 5.6. Figure 5.29 shows the contribution of different terms of the governing equation to $\hat{\mathbf{s}}_f^{mom}$. Only mesh refinement simulation is performed for this benchmark.

Table 5.6: Benchmark: Viscous dominated flow

<table>
<tr><th colspan="3"><b>Material:</b> $\nu = 0.5$ $\rho = 0.3$</th></tr>
<tr><th>Test Case</th><th>Fields</th><th>Discretization</th></tr>
<tr><td>VIS</td><td>$\hat{u}_x = e^{-t}\sin(x)\cos(y)$
$\hat{u}_y = -e^{-t}\cos(x)\sin(y)$
$\hat{Pr} = \frac{3}{40}e^{-2t}(\cos(2x)+\cos(2y))$</td><td>Space (cells/direction):
$2^2 \xrightarrow{increment\ 1} 8$
$\Delta t = 0.001$</td></tr>
</table>

Figure 5.30 shows the performance of the code in dealing with diffusive problems. It can be concluded that the observed order of accuracy is nearly second-order, as shown in Figure 5.30. The code is verified, having the observed order of convergence matching the theoretical order of convergence.

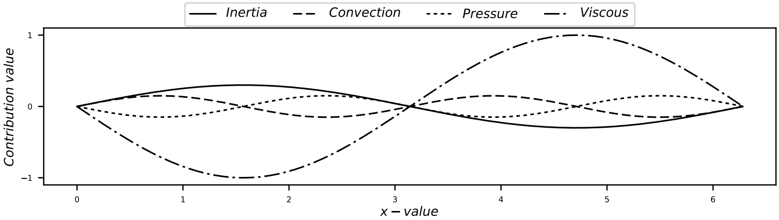


Figure 5.29: Viscous dominated functions at $X_2 = y = \pi$ and $t = 0.0$

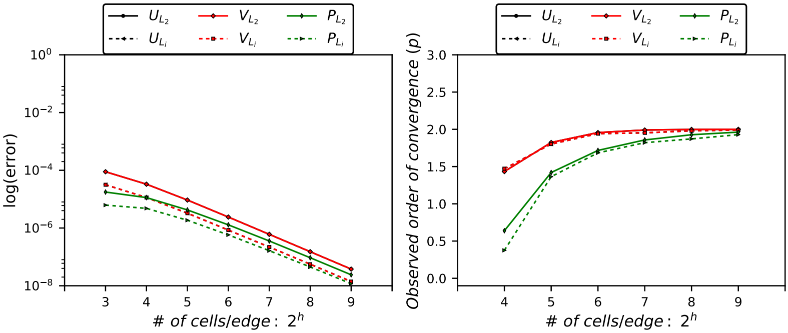


Figure 5.30: Viscous dominated flow spatial refinement unidirectional triangular grid

5.6 Summary

The presented benchmarks are pure mathematical exercises that do not evaluate the code performance in simulating the physics. The development of the benchmarks is code independent, which makes the benchmarks applicable to other codes. If a code successfully passes the TGV1, TGV2, and TGV3 benchmark, it is verified, and no further testing is required. Whereas, verifying the CFD application of KRATOS Multiphysics required more than the base benchmark. While performing the TGV2 simulation, the code performance is entirely unexpected. Therefore, more rigorous tests are required. Four test cases are developed to examine the code performance under different conditions by numerically exploiting the effect of each of the four terms in the N-S equations. From the inertia dominated benchmarks, it can be safely concluded that there is a time-space correlation in the discretized space. The time-space correlation originates from the calculation of the stabilization parameters in the problem formulation, which depends on the spatial grid size. In other words, only time-dependent solutions, such as IN1, cannot test the code performance thoroughly. This is not a disadvantage of the code, but it is part of the formulation used for the VMS element. All the benchmarks show an observed second-order accuracy. Therefore, the observed order of accuracy matches the theoretical order of accuracy. Finally, it can be safely said that the VMS monolithic solver is verified in the CFD application of the KRATOS Multiphysics for both ASGS and OSS formulations using either $WBZ-\alpha\ (Bossak)$ or $BDF2$ time discretization scheme and either triangular or quadrilateral grids.

All models are wrong, but
some models are useful.

George P. E. Box

CHAPTER 6

SOLUTION VERIFICATION METHODOLOGY APPLICATIONS

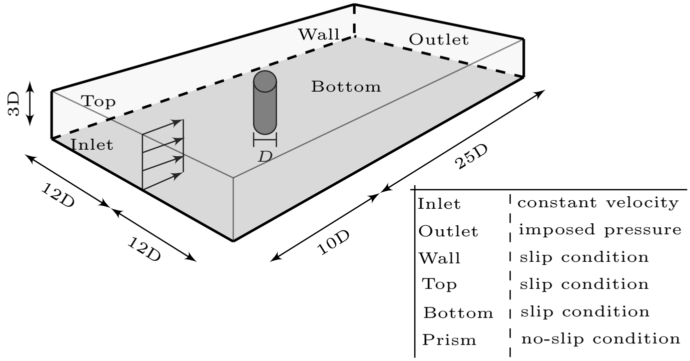
This chapter investigates the applicability of the solution verification approach introduced in Section 3.3. The methodology is applied on the $3-D$ flow around a circular cylinder at $Re = 3.90x10^3$, and the $3-D$ flow around a square prism at $Re = 3.90x10^3$. The monolithic variational multi-scale model implemented in KRATOS Multiphysics, introduced in Chapter 4, is used for the numerical simulation of the two flows. Furthermore, no wall function is used to estimate the boundary layer in the near-wall regions, but the VMS has some modeling for the sub-scales, as explained in Chapter 4. These two cases enable an extensive assessment of the mathematical model, flow dynamics, and solution verification methodology. This is a crucial step in understanding the applicability of the solution verification methodology to practical CWE applications where both smooth and sharp-edged bluff bodies are simulated with the existence of some experimental data for validation. Both test cases have the same computational set up to exploit the effect of flow characteristics and geometry on the performance of the newly developed solution verification methodology. Also, it proves that mesh design is not only dependent on Re, but also on the structure's geometry. These two cases are computed in order to evaluate the physical and numerical parameters related to the mathematical model, apply the solution verification methodology on well-studied test cases to test its applicability, study important flow features, and examine the performance of the simulation code with grids, not in the asymptotic range.

6.1 3-D Flow Around a Circular Cylinder

The flow around a circular cylinder has been extensively studied in the literature both experimentally and numerically (Beaudan et al. [135], Kravchenko et al. [136], Parnaudeau et al. [137], Franke [138], Ouvrard et al. [139], Wornom et al. [140], Bruno et al. [141], Lam et al. [142], D'Alessandro et al. [143], Kawata et al. [144], and Pereira et al. [145]). The selected test case is for $Re = 3900$ based on the undisturbed incoming velocity, U_{∞}, and cylinder diameter, D. In this order of Re, a transient shear layer flow regime is expected. Three shear layers characterize this flow regime; a boundary layer, a free shear layer, and a wake. As stated in Pereira [5], " In this range of Re, two boundary-layers detach from the cylinder's surface creating a recirculation region and two free shear-layers. Kelvin-Helmholtz (KH) rollers are then generated in these shear-layers ". The flow dynamics are governed by the flow in the recirculation zone. The frequencies f_{vs} and f_{KH} represent the instability frequencies called vortex-shedding frequency and Kelvin-Helmholtz frequency, respectively. These phenomena can be recognized throughout the spectral analysis of the stream-wise velocity signal in the free shear layer (Pereira [5]).

6.1.1 Computational Domain and Boundary Conditions

Figure 6.1 presents the geometrical dimensions of the computational domain and the applied boundary conditions. Figure 6.2 shows a plan view of the computational domain. The span-wise direction is defined such as $x_3/D = 3$. The inlet boundary is located $10D$ upstream of the cylinder center. The outlet boundary is located $25D$ downstream of the cylinder center. The side wall boundaries are $12D$ away from the cylinder center. Slip boundary condition is imposed at the side walls (at $x_2/D = \pm 12$). Also, slip boundary condition is imposed for top and bottom boundaries at $x_3/D = 0$ and $x_3/D = 3$. No-slip and impermeability boundary conditions are specified at the cylinder surface. An inlet boundary condition with fixed velocity is defined at the inlet $x_1/D = -10$. A zero-pressure gradient is defined at the outlet $x_1/D = 25$.



Inlet	constant velocity
Outlet	imposed pressure
Wall	slip condition
Top	slip condition
Bottom	slip condition
Prism	no-slip condition

Figure 6.1: Computational domain and boundary conditions: flow over a circular cylinder

The grid refinement studies are carried out with four grids having, approximately, $5.5M$, $4.4M$, $2.9M$ and $2M$ cells named h_1, h_2, h_3 and h_4, respectively. The grid's information is provided in Table 6.1. The domain is divided into four zones, as shown in Figure 6.2. Zone z_0 represents the cylinder surface and boundary layer. The grid density *(typical cell size h)* is

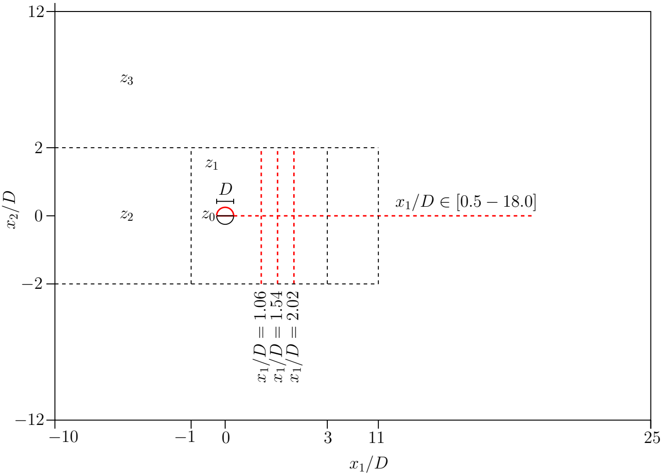


Figure 6.2: Plan view of the computational domain and measurement lines indicated in red

computed as the average of cell size estimate s_1 and cell size estimate s_2 (as described in Section 3.3.5.1). s_1 and s_2 are calculated as following:

$$s_1 = \sqrt[3]{\frac{Vol}{\#ofcells}} \qquad\qquad s_2 = \sum_{i=1}^{n_z} w_i h_i \tag{6.1}$$

Figure 6.3 and Figure 6.4 show the generated computational grids h_2 and h_4, respectively. It is important to note the zonal refinement concept application. The dimensionless time step $\Delta t_i U_\infty / D$ is 0.01 and is increased with the same value as the spatial discretization refinement ratio changes based on the mean values of cell size estimates s_1 and s_2, introduced in Section 3.3.5.1. Thus, the same refinement ratio is used in space and time. The calculation is run for 450 dimensionless time units. The influences of the spatial and temporal resolution, iterative convergence criteria, and simulation time to statistical convergence on the numerical accuracy of the predictions, are thoroughly studied in Pereira et al. [145]. The simulations are run on double precision and stringent iterative convergence criteria

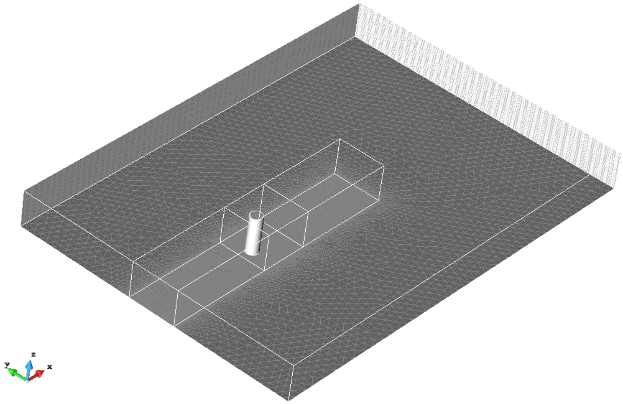


Figure 6.3: Computational grid h_2 for flow around a circular cylinder: the mesh is composed of 4.4 . 10^6 tetrahedral elements

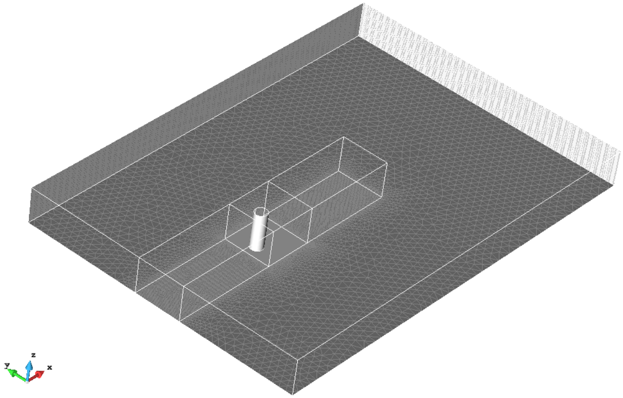


Figure 6.4: Computational grid h_4 for flow around a circular cylinder: the mesh is composed of 2.0 . 10^6 tetrahedral elements

Table 6.1: 3-D Cylinder Mesh Data: element size is normalized by D

Grid	# of cells		Zone element size		Cell size	Cell size
			$dx_1 = dx_2$	dx_3	s_1	s_2
h_1	5,501,331	z_0	0.008	0.0909	0.077086	0.054616
		z_1	0.04	0.0909		
		z_2	0.095	0.0909		
		z_3	0.6	0.0909		
h_2	4,418,865	z_0	0.009	0.0909	0.082927	0.059268
		z_1	0.05	0.0909		
		z_2	0.1	0.0909		
		z_3	0.6	0.0909		
h_3	2,914,857	z_0	0.01	0.0909	0.095264	0.065312
		z_1	0.06	0.0909		
		z_2	0.15	0.0909		
		z_3	0.6	0.0909		
h_4	2,014,353	z_0	0.02	0.0909	0.107751	0.089233
		z_1	0.07	0.0909		
		z_2	0.2	0.0909		
		z_3	0.6	0.0909		

to minimize the round-off and iterative convergence errors. The absolute and relative tolerance for velocity and pressure fields is set to 10^{-6} and 10^{-7}, respectively. Moreover, the tolerance for linear equation system iterative solver is set to 10^{-8} and 1000 maximum iterations. The flow is initialized with the constant free-stream velocity. The monolithic solver using ASGS and $WBZ - \alpha\ Bossak$ scheme is used for the numerical solution. The computational grid is designed to keep the $y^+ \leq 10$ for the coarsest mesh and the Courant number $CFL \leq 1.25$. The calculation of the y^+ and CFL is shown in appendix B.

6.1.2 Numerical Results and Discussion

A solution verification study, applying the guidelines introduced in Section 3.3, is presented in this section for the flow around a circular cylinder. Figure 6.2 shows the computational domain, and the red dotted lines indicate the measurement lines. These measurement lines are located at the mid-plane of the x_3 direction and are used to sample the velocity field. Both integral and local quantities are verified using the newly developed solution verification procedure[1]. First, drag (C_D) and lift (C_L) coefficients are computed for the cylinder. Second, the cylinder surface pressure coefficient (C_p) distribution is calculated along the red-dashed line indicated at the cylinder top half. Moreover, other flow quantities are measured, such as the Strouhal number St, the recirculation length L_r, and the minimum time-averaged stream-wise velocity $\overline{U_{min}}$. Also, the velocity profiles along the dashed red lines are measured. The velocity fields are measured at the center-line of the cylinder along the x_3 plane. All the flow quantities are time-averaged. Some of the measured quantities are validated against experimental data presented in Pereira et al. [17], Parnaudeau et al. [137], and Norberg [146].

Table 6.2 summarizes the flow quantities for discrete solutions on four grids $\left(h_i\right)$, the experimental data $\left(Exp.\right)$, and the predicted exact solution, $(Pre.)$ computed using the LSRMONTE discretization uncertainty estimator presented in Section 3.3.

As can be seen in Figure 6.5, the time-averaged drag coefficient, $\left(\overline{C_D}\right)$, is shown for the four different grid resolutions used. The LSRMONTE discretization uncertainty approach is used to estimate the numerical error at each grid. For the finest grid, $\left(h_1\right)$, the $\overline{C_D}$ has an exact error of 3.16 % compared to the experimental results. Also, it can be clearly seen that the experimental value is bounded by the discretization uncertainty estimated limits for each grid. By comparing the predicted exact solution $\left(\phi_{Pe}\right)$ to the experimental result, a 3.57 % error can be found. These marginal deviations and inclusion of the experimental values within the uncertainty bounds reflect an outstanding performance for the simulation code and the discretization uncertainty estimator (LSRMONTE). Finally, a second-order convergence is applied to get the best fitting curve to the simulation data to estimate the discretization uncertainty. A safety factor (SF) of 1.95 is used

[1] The verification methodology is summarized in appendix C

Table 6.2: The numerical solution with the respective discretization uncertainty estimate at different grid resolutions for the mean drag coefficient, $\overline{C_D}$, RMS lift coefficient, C_L^{RMS}, Strouhal number, St, recirculation length, $\overline{L_r}$, minimum stream-wise velocity magnitude, $\overline{U_{min}}$, and experimental data taken from Pereira et al. [17], Parnaudeau et al. [137], and Norberg [146]

	ϕ	$\overline{C_D}$	C_L^{RMS}	St	$\overline{L_r}$	$\overline{U_{min}}$
h_1	ϕ_{h_1}	0.949	0.046	0.209	1.663	-0.276
	U_{Num}	± 0.144	± 0.460	± 0.033	± 0.515	± 0.253
h_2	ϕ_{h_2}	0.957	0.037	0.214	1.646	-0.270
	U_{Num}	± 0.152	± 0.370	± 0.042	± 0.674	± 0.184
h_3	ϕ_{h_3}	0.979	0.034	0.216	1.714	-0.249
	U_{Num}	± 0.302	± 0.340	± 0.045	± 0.698	± 0.339
h_4	ϕ_{h_4}	0.966	0.039	0.224	1.634	-0.276
	U_{Num}	± 0.208	± 0.390	± 0.061	± 0.692	± 0.242
$Exp.$	ϕ_e	0.980	0.096	0.208	1.510	-0.340
	U_e	± 0.01	± 0.004	± 0.002	± 0.02	± 0.03
$Pre.$	ϕ_{Pe}	0.945	0.039	0.195	1.69	-0.265

to estimate the discretization uncertainty, which is completely different from the SF existing in the literature, as described in Section 3.3.5.3.

In Figure 6.6, the LSRMONTE is applied on the data for the root mean square (RMS) of the lift force coefficient C_L^{RMS}, St, $\overline{L_r}$, and $\overline{U_{min}}$. For the C_L^{RMS}, it can be observed that the LSRMONTE is extremely conservative, resulting in unreasonable uncertainty. This behavior can happen when the data points have no convergence trend or the same discrete solution for two sequentially refined grids. In other words, the convergence trend is difficult to capture because of the presence of outliers in the dataset or no improvement with refinement [2]. Thus, it is recommended to ignore the error bars' values and use the predicted exact solution or the fine grid solution values for comparison. The exact error of the fine grid solution is 15.2% compared to 59.4% for the predicted exact solution against the

[2] Further discussion on the performance, limitations, and possible improvement of the LSRMONTE can be found in Section 8.1.2

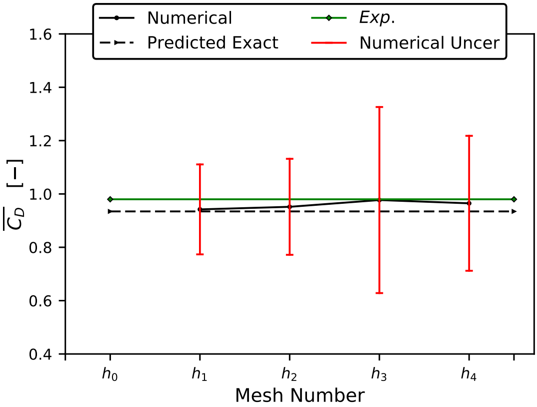


Figure 6.5: Discretization error estimation for the mean drag coefficient, $\overline{C_D}$, using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$

experimental results. The predicted exact solution is highly influenced by either the oscillatory data or the existence of an outlier in the four data points. A second-order convergence is assumed to provide the best curve fit to the simulation data. Finally, a safety factor of 2.0 is used for the uncertainty estimate.

Before getting into studying the simulation behavior over a turbulence quantity, such as St, point (31), located at the turbulence zone with coordinates $\left(X_1/D = 1.06, X_2/D = 1.076, X_3/D = 1.5\right)$ is used for the computation of the St number using spectral analysis. Then the turbulent flow structure is studied using an autocorrelation study for the velocity signal, where the flow integral length scale can be calculated. Figure 6.7 shows the velocity signals used for the spectral and autocorrelation analysis[3]. Figure 6.8 shows the autocorrelation functions for the velocity signal at the wake region behind the cylinder. It can be observed that the flow structure follows a decaying sinusoidal form. This implies that the flow structure is not random, as experienced in boundary layer flows in practical CWE applications. Furthermore, the integral length scale at the four different grids is different

[3] Appendix D and A show how to calculate flow quantities

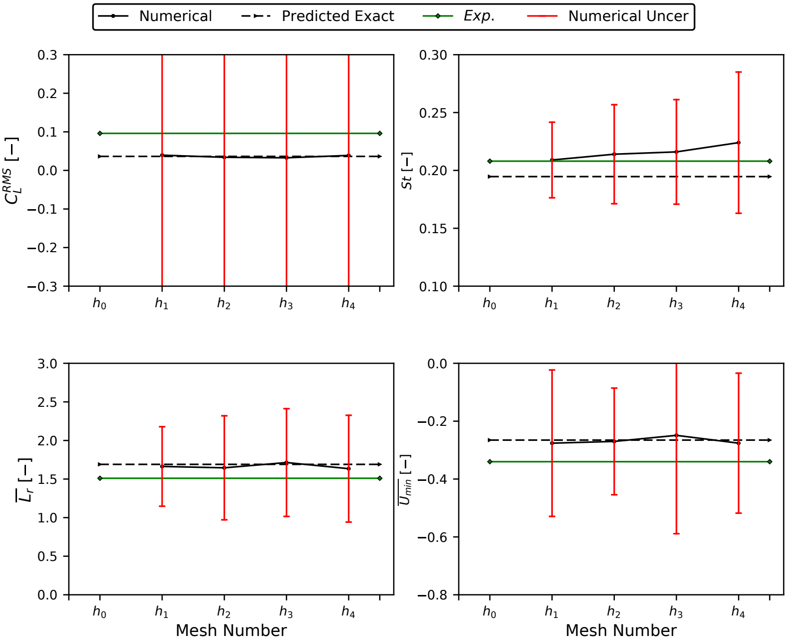


Figure 6.6: LSRMONTE Discretization error estimation for the RMS lift coefficient, C_L^{RMS}, Strouhal number, St, recirculation length, $\overline{L_r}$, minimum stream-wise velocity magnitude, $\overline{U_{min}}$: flow around a circular cylinder at $Re = 3900$

and is very close to the cylinder diameter (D), which shows an adequate grid resolution.

Figure 6.9 shows the power spectral density analysis for the stream-wise velocity signal at point 31. It can be seen that the spatial formation of the Kelvin-Helmholtz rollers is developed. This is shown by the frequency f_{KH} broadband shown in Figure 6.9. The Kelvin-Helmholtz rollers are decomposed into high-intensity turbulence, which leads to the formation of the vortex-shedding. This can be clearly observed by the sharp peak at frequency f_{vs}. Finally, spectral analysis's observation of these flow features proves a reasonable prediction for the flow statistics.

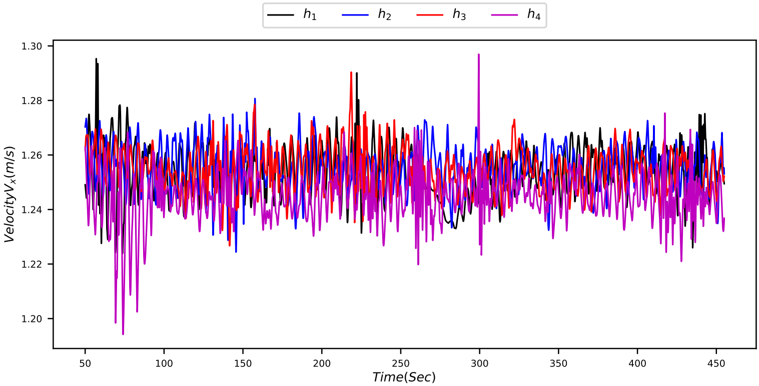


Figure 6.7: Velocity signal at point 31 for four different grid resolutions for flow over a circular cylinder at $Re = 3900$

The Strouhal number St can be computed using the vortex shedding frequency $\left(f_{vs}\right)$ using Equation (6.2):

$$St = \frac{f_{vs} D}{U_\infty} \tag{6.2}$$

By studying the St (Figure 6.6), it can be clearly seen that the simulation behavior and the LSRMONTE performance is outstanding. The St data has a 1.57 convergence rate for fitting a curve to the simulation data. This convergence rate does not apply by other uncertainty estimator approaches. Other approaches assume the convergence rate to be first-order, or second-order, or mixed order, which does not always result in the best curve fitting. Also, it can be observed that the error bars are not very conservative, with a safety factor of 1.97, yet still include the experimental values.

Observing the figures for $\overline{L_r}$ and $\overline{U_{min}}$ (Figure 6.6), it can be seen that the simulation code and the LSRMONTE technique have acceptable performance. The error bars in both cases bound the experimental value. One data point h_3 on the $\overline{U_{min}}$ figure has a very conservative error estimate, which is higher than the coarser mesh results h_4. This happens because the uncertainty estimator includes a parameter that accounts for the stan-

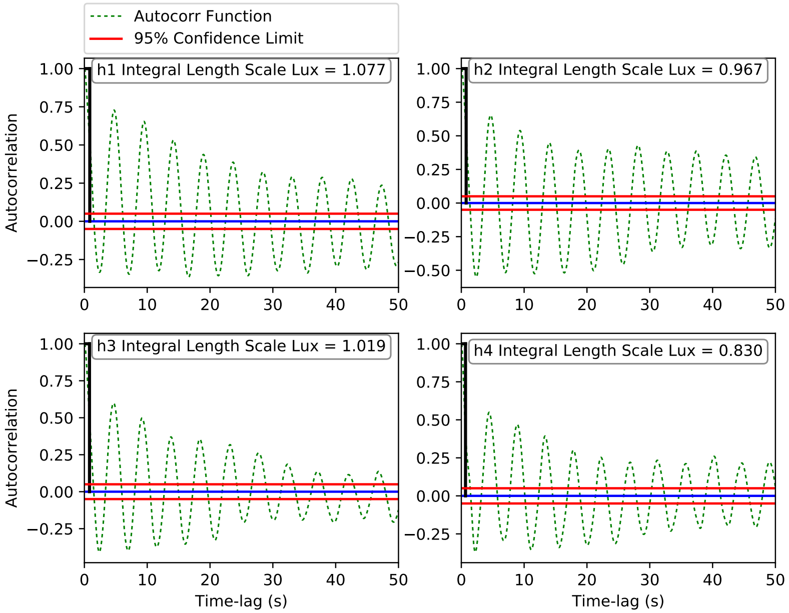


Figure 6.8: Autocorrelation analysis for the velocity signal at point 31 for four different grid resolutions for flow over a circular cylinder at $Re = 3900$

dard deviation between the simulation data point and the corresponding fitted data, which leads to higher uncertainty if a higher standard deviation exists.

Figure 6.10 compares the magnitudes of statistical uncertainty and numerical uncertainty. The statistical uncertainty is mainly computed using the bootstrapping technique introduced in Section 3.3.4. It is clearly seen that the statistical error is insignificant compared to the numerical error. This indicates a sufficient simulation time to converge the statistics in unsteady flow simulation. On the other hand, the LSRMONTE shows exceptional performance except at three points, where there is no improvement in the discrete solution with refinement, which leads to a very conservative uncertainty estimate. For all the other points, the LSRMONTE performance is

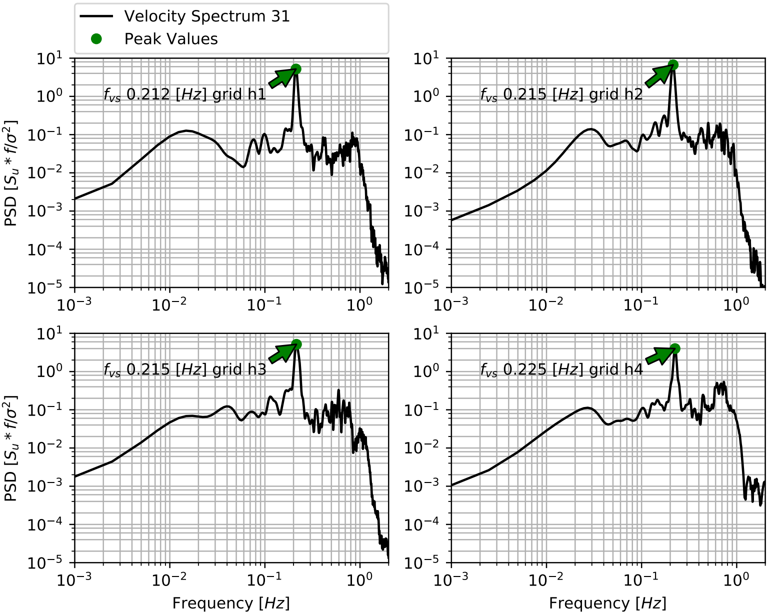


Figure 6.9: Power spectral density analysis for the velocity signal at point 31 for four different grid resolutions for flow over a circular cylinder at $Re = 3900$

outstanding and provides very reasonable results. By looking at Figure 6.11, the experimental results are plotted with the discrete solution with error bars. The fine grid solution h_1 and the experimental results are in a very good agreement with small deviations in some regions, especially the maximum suction region. Knowing that the assessment of local quantities is not an easy task; the error bars with the smaller safety factor are still bounding the experimental results within an acceptable range.

In Figure 6.12, the time-averaged stream-wise velocity component is measured along line X_1/D at the center-line of the cylinder at $\left(X_2/D = 0\right)$ and $\left(X_3/D = 1.5\right)$. It can be observed that the numerical uncertainty estimate is conservative. In this case, the conservativeness happens because strong flow mixing is experienced in the regions behind the circular cylinder. The

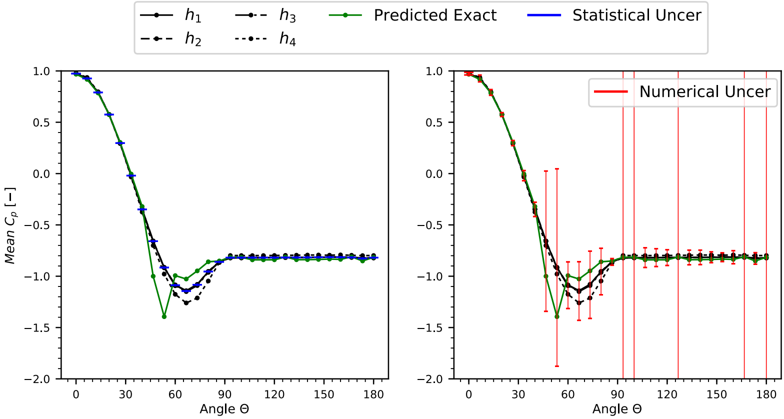


Figure 6.10: Left figure shows the time-averaged pressure coefficient distribution on the cylinder surface, $\overline{C_p}$, and statistical uncertainty. Right figure shows the discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1

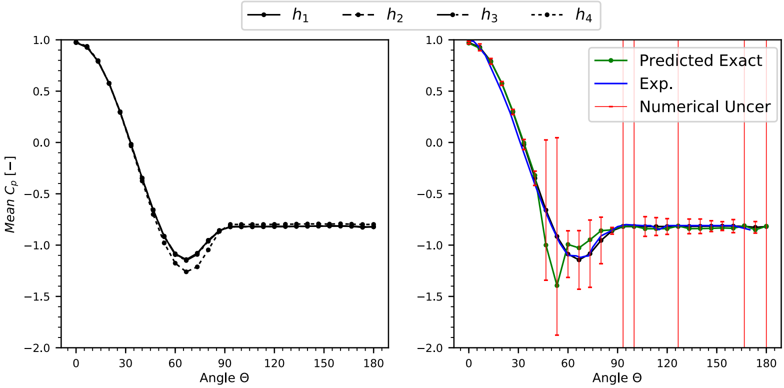


Figure 6.11: Left figure shows the time-averaged pressure coefficient distribution on the cylinder surface, $\overline{C_p}$. Right figure shows the discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1 and the experimental data taken from Norberg [146]

scales of the resolved vortices are different depending on the grid refinement. Thus, at some points, the uncertainty estimate is very conservative. Another reason for this problem would be the need for a higher refinement ratio to have a significant variation in the resolved scales.

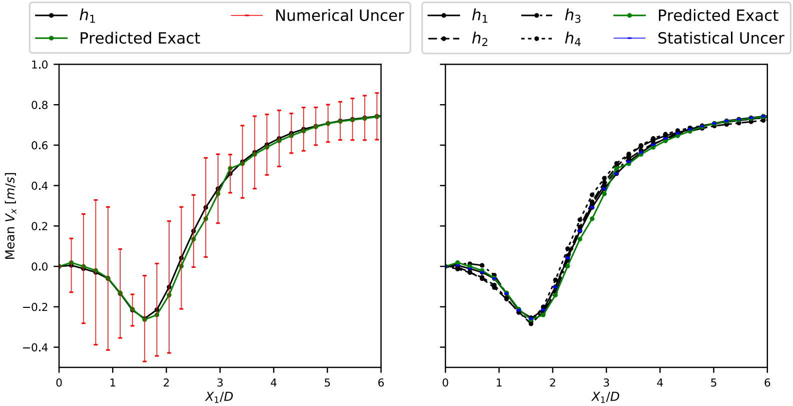


Figure 6.12: Left figure shows the mean stream-wise velocity discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1 at $\left(X_2/D = 0, X_3/D = 1.5\right)$. Right figure shows the discrete solution for the four grids and statistical uncertainty on the finest grid h_1

Moreover, Figure 6.13 and Figure 6.14 show the mean stream-wise velocity and transverse velocity components at $X_3/D = 1.5$ along X_2/D at three different locations on the vortex shading region lines $X_1/D = 1.06$, $X_1/D = 1.54$, and $X_1/D = 2.02$. Very high uncertainties are observed at some points in the region of high turbulence. Thus, it is recommended to use a higher refinement ratio between each mesh, or use a higher local refinement for the region behind the cylinder up to $X_1/D = 3$.

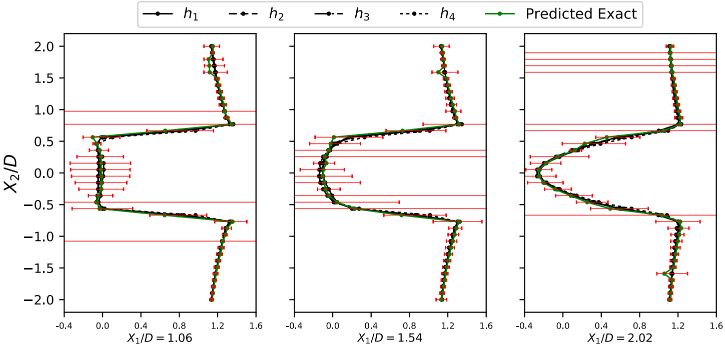


Figure 6.13: The mean stream-wise velocity discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1 at $(X_3/D = 1.5)$.

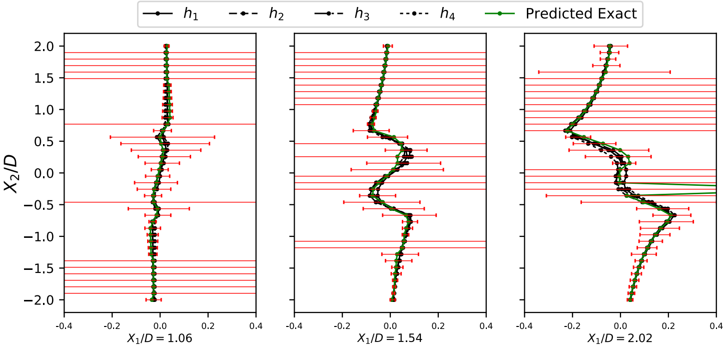


Figure 6.14: The mean transversal velocity discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1 at $(X_3/D = 1.5)$.

Finally, it can be safely stated that the numerical code and the LSRMONTE uncertainty discretization technique have an outstanding performance. Some of the regions need special treatment, such as local refinement, to improve the simulation results. The detection of regions that need more refinement can be identified by computing the discretization uncertainty using LSRMONTE for local quantities such as velocity. If the LSRMONTE results are very conservative to the extent of being unrealistic, then refinement is required in this area. The regions that need special treatment can be observed in Figure 6.11, Figure 6.12, and Figure 6.13, where the discretization uncertainty estimate is too conservative to be realistic[4].

[4] Further discussion on the performance, limitations, and possible improvement of the LSRMONTE can be found in Section 8.1.2

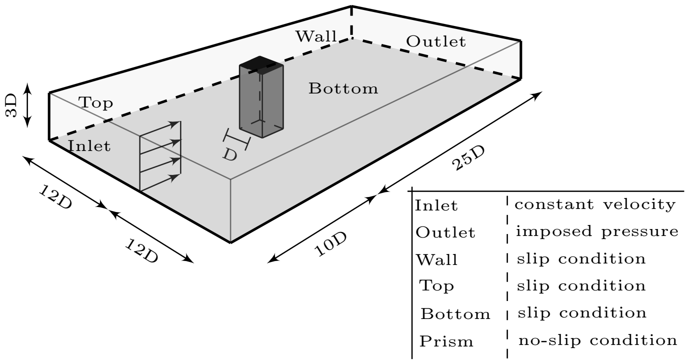
6.2 3-D Flow Around a Square Prism

The flow around a square prism has been extensively studied experimentally and numerically for a long time (Bruno et al. [141], Lam et al. [142], Bearman et al. [147], Lyn et al. [148], Nanda [149], Saha et al. [150], Khademinezhad et al. [151], Dutta et al. [152], Arslan et al. [153], and Gera et al. [154]). The flow past a square prism is highly unsteady compared to the flow around a circular cylinder. The sharp-edges flow separation triggers the complexity of the simulation. The selected test case is for a Reynolds number based on the undisturbed incoming velocity, U_∞, and side length, D, equal to 3900. In this range of Re, a transient shear layer flow regime is expected. This flow regime is characterized by three shear layers; a boundary layer, a free shear layer, and a wake. A more elaborate discussion on the flow properties can be found in Section 6.1.

6.2.1 Computational Domain and Boundary Conditions

Figure 6.15 presents the geometrical dimensions of the computational domain and the applied boundary conditions. Figure 6.16 shows a plan view of the computational domain. The span-wise direction is defined such as $x_3/D = 3$. The inlet boundary is located $10D$ upstream of the square center. The outlet boundary is located $25D$ downstream of the square center. The side wall boundaries are $12D$ away from the square center. A Slip boundary condition is imposed at the side walls (at $x_2/D = \pm 12$). Also, slip boundary condition is imposed for top and bottom boundaries at $x_3/D = 0$ and $x_3/D = 3$. No-slip and impermeability boundary conditions are specified at the prism surface. An inlet boundary condition with fixed velocity is defined at the inlet $x_1/D = -10$. A zero-pressure gradient is defined at the outlet $x_1/D = 25$.

The grid refinement study is carried out with four grids having approximately $5.5M$, $4.4M$, $2.9M$, and $2M$ cells. These grids are named h_1, h_2, h_3 and h_4, respectively. Figure 6.3 and Figure 6.4 show the generated computational grids h_2 and h_4, respectively. It is important to note the zonal refinement concept application. The grids' information are provided in Table 6.3. The domain is divided into four zones, see Figure 6.16. Zone z_0 represents the cylinder surface and boundary layer. The grid density *(typical cell size h)* is computed as the average of s_1 and s_2 (as described in Section 3.3.5.1).



Inlet	constant velocity
Outlet	imposed pressure
Wall	slip condition
Top	slip condition
Bottom	slip condition
Prism	no-slip condition

Figure 6.15: Computational domain and boundary conditions: flow over a square prism

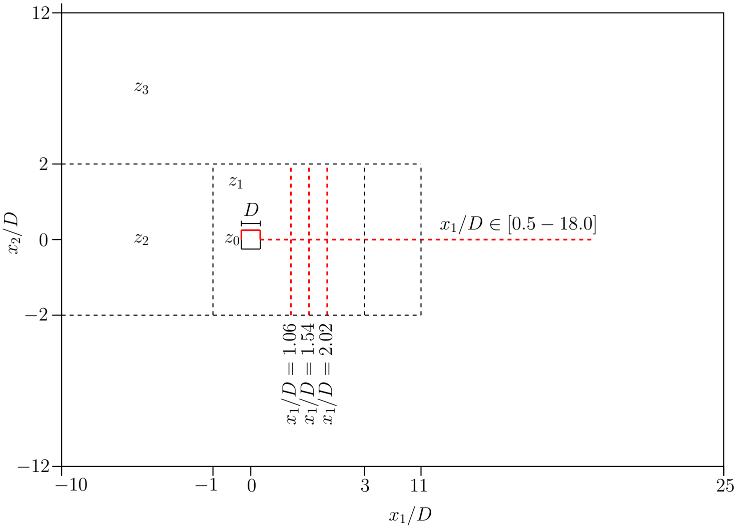


Figure 6.16: 3-D Square-Prism Simulation: Plan view of the computational domain and measurement lines indicated in red

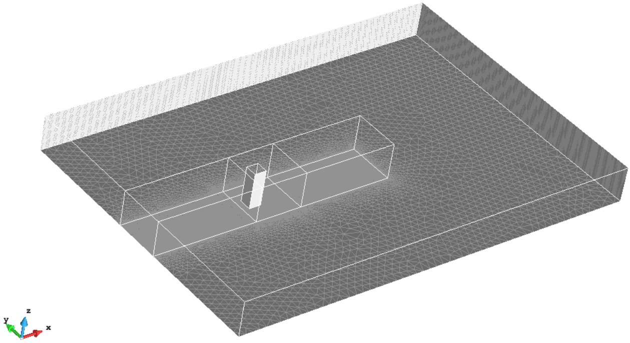


Figure 6.17: Computational grid h_2 for flow around a square prism: the mesh is composed of $4.4 \,.\, 10^6$ tetrahedral elements

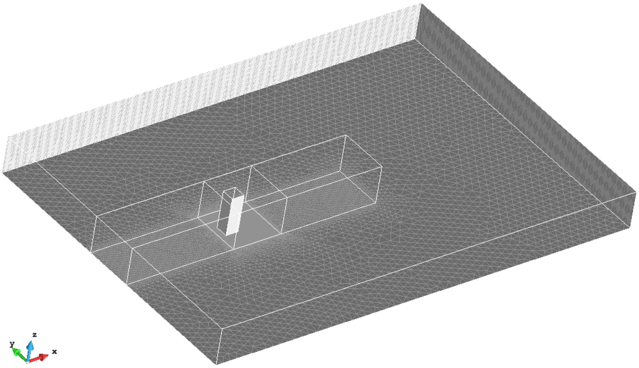


Figure 6.18: Computational grid h_4 for flow around a square prism: the mesh is composed of $2.0 \,.\, 10^6$ tetrahedral elements

Table 6.3: 3-D Square-Prism Mesh Data: element size is normalized by D

Grid	# of cells		Zone element size		Cell size	Cell size
			$dx_1 = dx_2$	dx_3	s_1	s_2
h_1	$5,512,112$	z_0	0.008	0.0909	0.077036	0.0546157
		z_1	0.04	0.0909		
		z_2	0.095	0.0909		
		z_3	0.6	0.0909		
h_2	$4,463,910$	z_0	0.009	0.0909	0.082647	0.059268
		z_1	0.05	0.0909		
		z_2	0.1	0.0909		
		z_3	0.6	0.0909		
h_3	$2,956,437$	z_0	0.01	0.0909	0.094815	0.065312
		z_1	0.06	0.0909		
		z_2	0.15	0.0909		
		z_3	0.6	0.0909		
h_4	$2,039,994$	z_0	0.02	0.0909	0.107297	0.089232
		z_1	0.07	0.0909		
		z_2	0.2	0.0909		
		z_3	0.6	0.0909		

The dimensionless time step $\Delta t_i U_\infty / D$ is 0.01 and increased with the same value as the spatial discretization refinement ratio changes. The spatial refinement ratio is defined based on the mean value of estimates s_1 and s_2, described in Section 3.3.5.1. Thus, the same refinement ratios are used in space and time. The calculations run for 470 dimensionless time units. The same computational setup is shown in Section 6.1.1 is used to simulate the flow around the square prism.

6.2.2 Numerical Results and Discussion

The measured quantities can be divided into integral and local quantities. Calculations for the statistical quantities, turbulence quantities and solution verification methodology are summarized in appendices A, B, and C, respectively.

6.2.2.1 Integral Quantities

In this section, the code's ability to calculate integral quantities is assessed using the newly developed solution verification methodology with uncertainty discretization estimation, LSRMONTE. The integral quantities measured in this case are the mean drag coefficient, $\overline{C_D}$, RMS lift coefficient, C_L^{RMS}, the Strouhal number, St, and the integral length scale, $L_{(ux)}$. Figure 6.19 shows the discretization error estimate of the drag force coefficient.

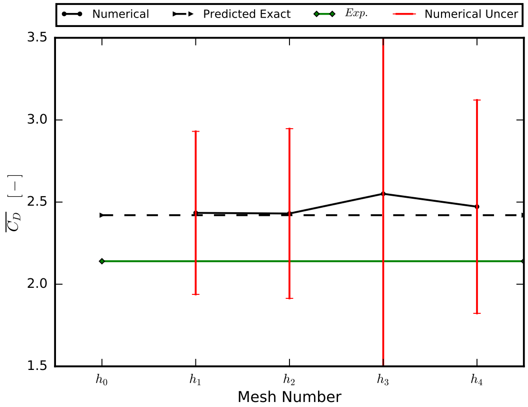


Figure 6.19: Discretization error estimation for the mean drag force coefficient, $\overline{C_D}$, using LSRMONTE approach for flow over a square prism at $Re = 3900$

It is observed that mesh (h_3) has the highest uncertainty due to a higher deviation from the estimated exact value. This behavior is due to having simulations not in the asymptotic range. Also, it can be seen that the numerical error is decreasing with refinement as expected. A 12.5 % exact error is computed between the numerically predicted exact value and the experimental value. The error is larger than the one computed for the flow around the circular cylinder in Section 6.1.2, because of the effect of sharp edges, which makes the flow more complicated on the sides and leeward side of the prism. However, the experimental result is bounded by the error estimate. An oscillatory convergence is noticed between meshes h_2, h_3, and h_4. On the other hand, monotonic convergence is observed between meshes h_1, h_2, and h_3. A second-order convergence order is used for the curve fitting.

In Figure 6.20, the LSRMONTE is applied on the data for the root mean square (RMS) of the lift force coefficient C_L^{RMS}, St, $\overline{L_r}$, and $\overline{U_{min}}$. For the St, it can be observed that the LSRMONTE is extremely conservative, which results in unreasonable uncertainty. This behavior can happen when the data points have no convergence trend, which requires higher refinement ratios[5]. Thus, it is recommended to ignore the values of the error bars and use the predicted exact solution or the fine grid solution values for comparison. A more conservative performance can be seen in estimating the discretization error in Figure 6.20 in assessing the C_L^{RMS}. An oscillatory convergence is observed between all meshes, and a second-order convergence is used for the curve fitting.

The complexity of the flow is highly reflected in the estimation of the numerical uncertainty, as explained above. Figure 6.21 shows the velocity signal at $X_1/D = 1.06$, $X_2/D = 1.282$, and $X_3/D = 1.5$. The auto-correlation is calculated and plotted in Figure 6.22. The integral length scale of the vortices in the vortex shedding zone is computed from the auto-correlation function. Furthermore, the auto-correlation plot shows a periodic wave on the back of the structure. The integral length scale ($L_{(ux)}$) of the flow turbulence around a square prism is ≈ 25% higher than that of the flow around a circular cylinder. Moreover, the Strouhal number (St) is computed from the spectral analysis shown in Figure 6.23. The numerical uncertainty for the St is estimated and shown in Figure 6.20. Since there is no significant improvement in the calculated St with refinement, it can be seen that the estimated uncertainty is too conservative[5]. The estimated uncertainty is unrealistic in this case. Table 6.4 summarizes all the findings

[5] Further discussion on the performance, limitations, and possible improvement of the LSRMONTE can be found in Section 8.1.2

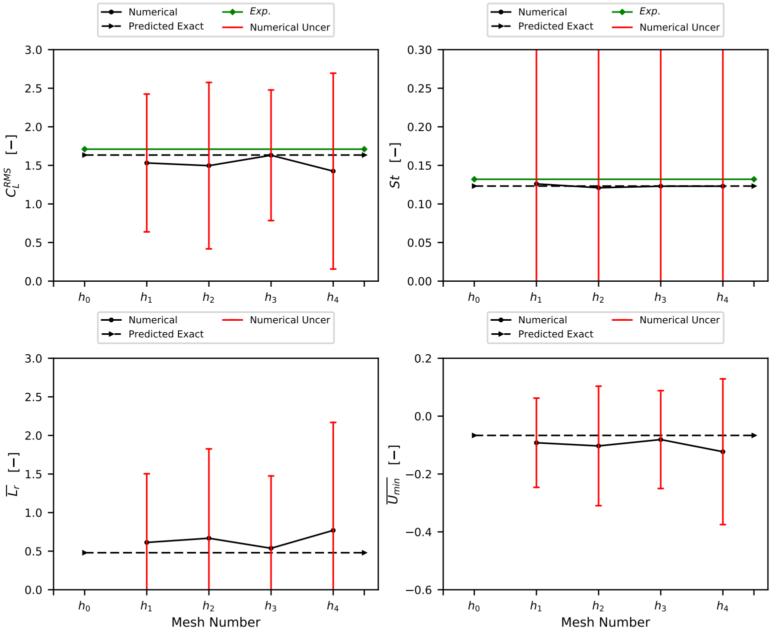


Figure 6.20: LSRMONTE Discretization error estimation for the RMS lift coefficient, C_L^{RMS}, Strouhal number, St, recirculation length, $\overline{L_r}$, minimum stream-wise velocity magnitude, $\overline{U_{min}}$: flow around a square prism at $Re = 3900$

in the measured quantities with estimated uncertainty for each grid level. The results are compared to some experimental values available in the literature (Bearman et al. [147], Lyn et al. [148], Nanda [149], and Saha et al. [150]). Unfortunately, no uncertainty is available for the experimental data for a fair comparison with the numerical results.

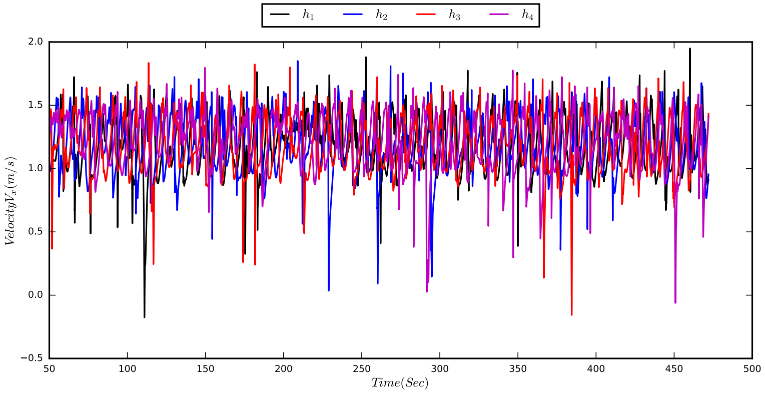


Figure 6.21: Velocity signal at point 33 for four different grid resolutions for flow over a square prism at $Re = 3900$

Table 6.4: The numerical solution with the respective discretization uncertainty estimate at different grid resolutions for the mean drag coefficient, $\overline{C_D}$, RMS lift coefficient, C_L^{RMS}, Strouhal number, St, recirculation length, $\overline{L_r}$, minimum stream-wise velocity magnitude, $\overline{U_{min}}$, and experimental data taken from Bearman et al. [147], Lyn et al. [148], Nanda [149], and Saha et al. [150]

	ϕ	$\overline{C_D}$	C_L^{RMS}	St	$\overline{L_r}$	$\overline{U_{min}}$
h_1	ϕ_{h_1}	2.434	1.531	0.126	0.613	-0.092
	U_{Num}	± 0.496	± 0.893	± 1.074	± 0.89	± 0.154
h_2	ϕ_{h_2}	2.430	1.496	0.121	0.668	-0.103
	U_{Num}	± 0.516	± 1.078	± 1.079	± 1.157	± 0.206
h_3	ϕ_{h_3}	2.551	1.631	0.123	0.538	-0.081
	U_{Num}	± 1.112	± 0.846	± 1.077	± 0.937	± 0.169
h_4	ϕ_{h_4}	2.472	1.426	0.123	0.770	-0.123
	U_{Num}	± 0.649	± 1.269	± 1.077	± 1.399	± 0.252
$Exp.$	ϕ_e	2.140	1.710	0.132	——	——
$Pre.$	ϕ_{Pe}	2.42	1.634	0.123	0.480	-0.067

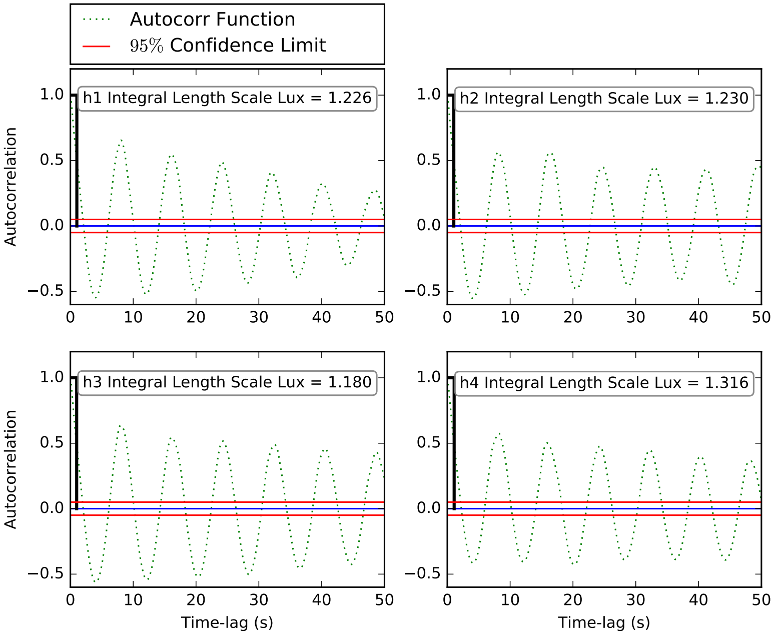


Figure 6.22: Auto-correlation analysis for the velocity signal at point 33 for four different grid resolutions for flow over a square prism at $Re = 3900$

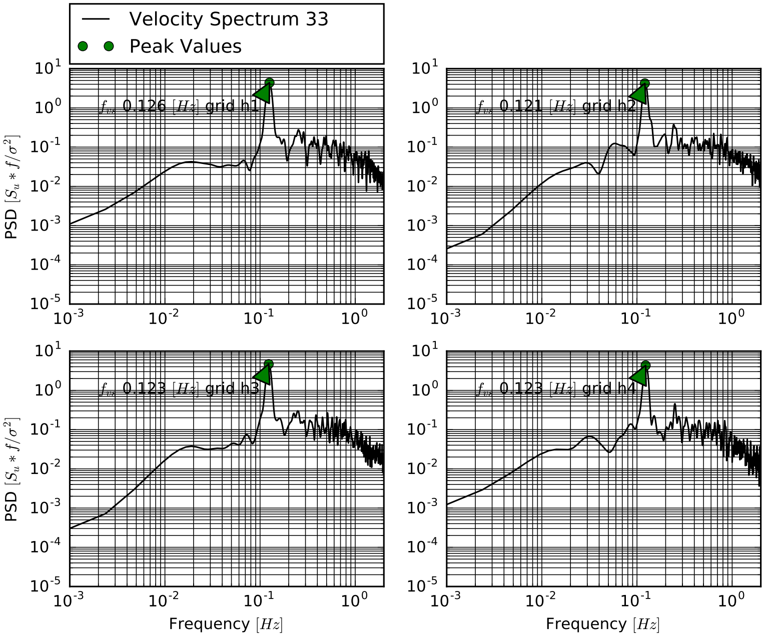


Figure 6.23: Power spectral density analysis for the velocity signal at point 31 for four different grid resolutions for flow over a circular cylinder at $Re = 3900$

6.2.2.2 Local Quantities

In this section, the code performance and numerical errors are evaluated for local quantities such as pressure and velocity at selected locations indicated by the red dashed lines in Figure 6.16. Local quantities are usually more sensitive to discretization than integral quantities. All the quantities are measured at the mid-plane in the x_3 direction. The following quantities are measured:

1. The recirculation length $\overline{L_r}$: It is measured from the back of the structure. Figure 6.20 shows the error estimation on the four grid levels. The estimate is too conservative on the coarsest grid with ≈ 80 % error. Then, the error is decreasing with refinement.

2. Minimum mean stream-wise velocity: It is defined at the center-line of the x_2 direction along the flow direction x_1. Figure 6.20 shows the discretization uncertainty with too conservative estimate[6] because the results are oscillating around zero.

3. Both stream-wise and transverse velocity along X_2 on lines $X_1/D = 1.06$, $X_1/D = 1.54$, and $X_1/D = 2.02$: The velocity is measured at the center-line of the domain at $X_3/D = 1.5$. This region contains high flow separation and turbulence. The estimated uncertainty is unrealistic[6] at some locations, as shown in Figure 6.24, at lines $X_1/D = 1.06$ and $X_1/D = 2.02$. For the three reference lines, a realistic estimate can be observed in the region bounded by $X_2/D = [-0.5, 0.5]$. Therefore, a higher refinement ratio is required to get better estimate.

   From Figure 6.25, it can be seen that the vertical movement of the flow vanishes as the flow develops and approaches the recirculation length. Flow symmetry can be clearly observed. Away from the edges of the square-prism, eddies are formed at $X_2/D \geq 0.5$ and $X_2/D \leq -0.5$.

   Finally, a higher refinement ratio is needed to resolve better the flow on the back of the structure.

[6] Further discussion on the performance, limitations and possible improvement of the LSRMONTE can be found in Section 8.1.2

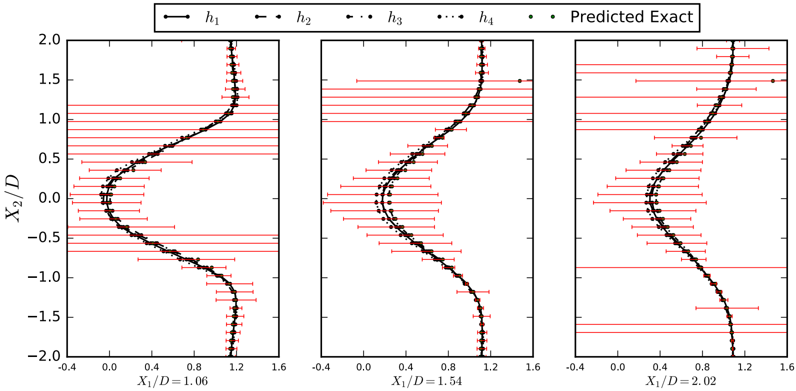


Figure 6.24: The mean stream-wise velocity discretization error estimation using LSRMONTE approach for flow over a square prism at $Re = 3900$ on the finest mesh h_1 at $(X_3/D = 1.5)$.

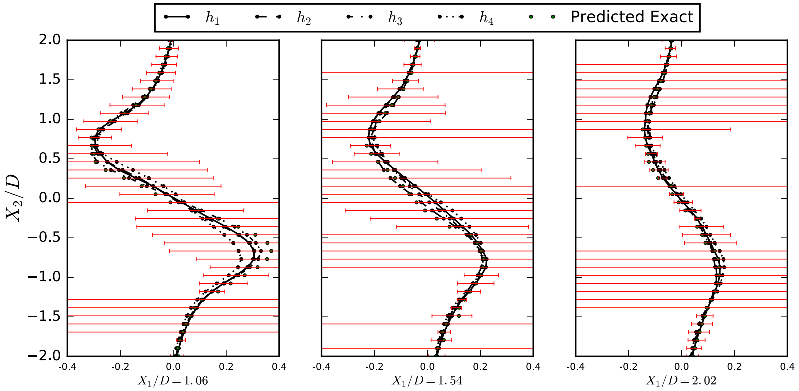


Figure 6.25: The mean transversal velocity discretization error estimation using LSRMONTE approach for flow over a square prism at $Re = 3900$ on the finest mesh h_1 at $(X_3/D = 1.5)$.

4. The stream-wise velocity at the back of the structure measured along the domain center-line: Figure 6.26 shows the mean stream-wise velocity. It can be seen that the numerical uncertainty is much higher than the statistical uncertainty. The results analysis indicates the need for a higher refinement ratio, although the results are in alignment with the experimental data. The higher refinement ratio will result in resolving smaller scales of turbulence.

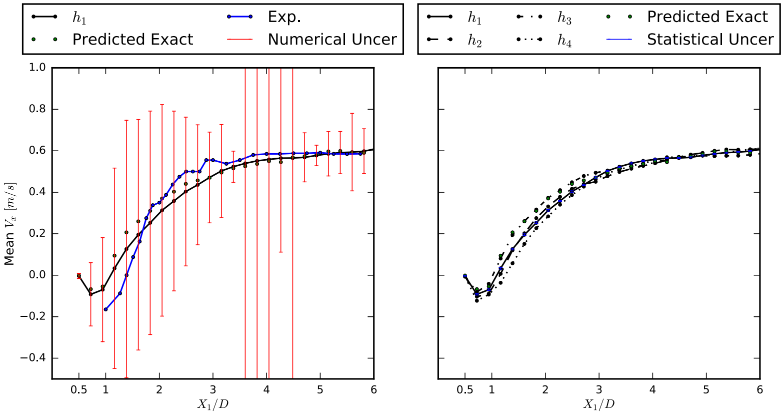


Figure 6.26: Left figure shows the mean stream-wise velocity discretization error estimation using LSRMONTE approach for flow over a square prism at $Re = 3900$ on the finest mesh h_1 at $\left(X_2/D = 0, X_3/D = 1.5\right)$. Right figure shows the discrete solution for the four grids and statistical uncertainty on the finest grid h_1

5. The mean pressure coefficient: It is measured at the upper half surface of the structure. Figure 6.27 and Figure 6.28 show the statistical error, numerical estimated error, and experimental data. Figure 6.27 compares the statistical to numerical uncertainty. The statistical uncertainty can be neglected (≤ 1 %) because the simulations are run for sufficient time to converge the statistics.

   Figure 6.28 compares the numerical results, including estimated discretization error, and the experimental data from Nanda [149]. The edge points are the most critical points where the discretization

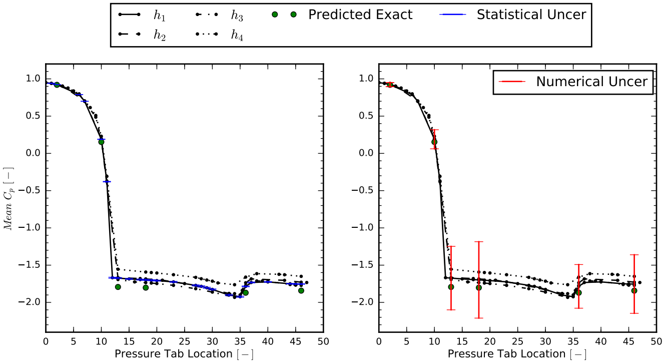


Figure 6.27: Left figure shows the time-averaged pressure coefficient distribution on the square surface, $\overline{C_p}$, and statistical uncertainty. Right figure shows the discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1

uncertainty is present. It can be seen that the numerical uncertainty is reliable and bounds the experimental data. Finally, it can be safely said that the numerical error estimation must be calculated to indicate simulation codes' performance. Also, the quality of the computational domain grid can be assessed through rigorous data sampling for local quantities and estimation of the associated uncertainties, which indicate the improvement of the solution with refinement and can be a guide for building a more reliable simulation.

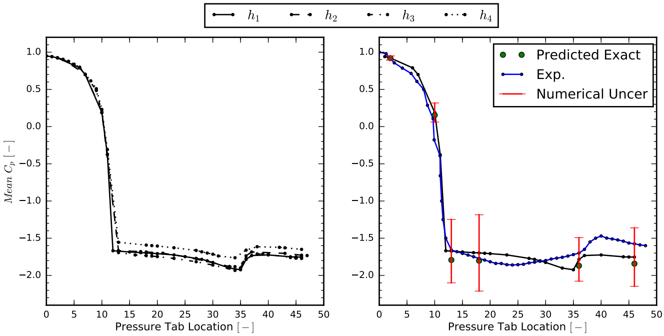


Figure 6.28: Left figure shows the time-averaged pressure coefficient distribution on the cylinder surface, $\overline{C_p}$. Right figure shows the discretization error estimation using LSRMONTE approach for flow over a circular cylinder at $Re = 3900$ on the finest mesh h_1 and the experimental data taken from Nanda [149]

Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.

George P. E. Box

CHAPTER 7

Atmospheric Boundary Layer Flow Around a Highrise Building: Solution Verification

This chapter presents the required workflow to predict wind loads on structures and estimate the numerical errors. Moreover, the application of the newly developed LSRMONTE approach for the estimation of numerical errors for the solution of the atmospheric boundary layer (ABL) flow around a highrise building at full-scale Reynolds number is presented. This chapter's main objective is to present how to evaluate wind loads on a tall and slender structure with confidence using an eddy-resolving simulation technique. ABL flow turbulence, together with bluff bodies aerodynamics such as vortex shedding and intermittent reattachment of shear layers, leads to along-wind and across-wind vibrations (Ricci et al. [155]). Meanwhile, flow turbulence imposes the main challenge for estimating discretization uncertainty. Therefore, the selection of the quantities of interest is significant in such a computation. Some publications on the simulation of flow around a highrise building can be found in Péntek et al. [156] and Thordal et al. [157].

7.1 Experimental Setup

This section introduces the experimental setup adopted to simulate a highrise building numerically. Experiments were performed at the atmospheric boundary layer wind tunnel facility of the Tokyo Polytechnic University (TPU). Pressure data are available and collected in a public database (*Tokyo Polytechnic University Aerodynamic Database* [158]). The database is organized according to building shape, building geometry, and flow characteristics. The focus of this work is an isolated highrise building with an equal breadth (B) to depth (D) ratio, and breadth to height (H) ratio, equals to 1 : 5 (Figure 7.1)

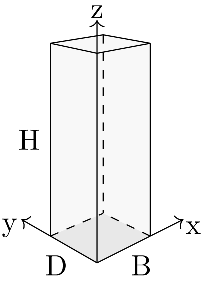


Figure 7.1: Highrise building geometry

The wind tunnel length scale is equal to 1/400, which leads to full scale building dimensions as $D = B = 40m$ and $H = 200m$. The wind tunnel has a $2.2m$ X $1.8m$ test section. This lead to a 1.2% blockage ratio. A terrain category IV with α exponent equals 0.25 is used for the reproduced wind profile, according to the Architectural Institute of Japan (AIJ) standards (AIJ [159]). The wind profile is defined as:

$$U_{(Z)} = U_{ref} \left(\frac{Z}{Z_{ref}} \right)^{\alpha} \tag{7.1}$$

where $U_{(Z)}$ is the mean wind speed at height Z, U_{ref} is the mean wind speed at reference height Z_{ref} and the power-law index ($\alpha = 0.25$). More discussion on the AIJ recommendations for wind loads on buildings can be found in AIJ [159] and Tamura et al. [160].

The experimentally produced wind has a mean wind velocity equal to $U_{ref} = 11.11\ m/s$ at the building reference height ($Z_{ref} = 160\ m$). The turbulence intensity is $I_{Z_{ref}} = 14\ \%$ at the building reference height. The turbulence intensity (I_Z) at height Z is defined as

$$I_Z = \frac{\sigma_{u_Z}}{U_Z} \tag{7.2}$$

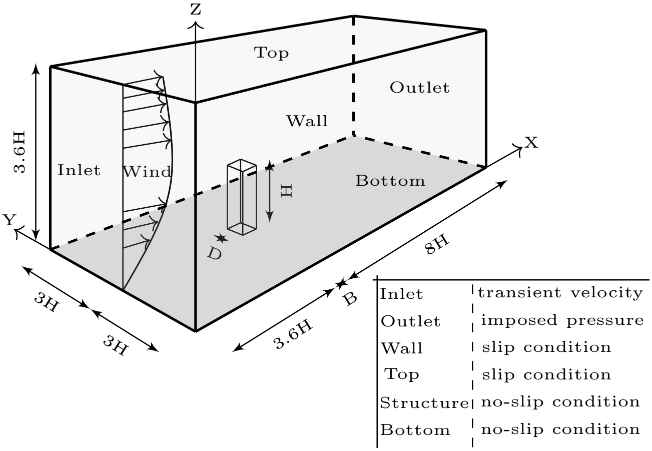
where σ_{U_Z} is the standard deviation of the fluctuating wind velocity u_Z. The building model was equipped with 500 pressure taps that acquired data at a sampling frequency of $1000 Hz$ for 32.8 seconds. A $1.52 * 10^8$ full-scale Reynolds number is used in the numerical simulation compared to a wind tunnel scale $Re = 3.8 * 10^5$ based on building height. Finally, a zero wind angle is used in the scope of this work. The wind flow is parallel to the local building $x-axis$.

7.2 Computational Setup

The VMS approach described in Chapter 4 is used for the flow simulation. Figure 7.2 shows a three-dimensional view of the used computational domain size. Also, the boundary conditions are indicated in Figure 7.2, and no wall-function is used in the near-wall region.

The domain size is designed based on recommendations from the literature (Ricci et al. [155] and Péntek et al. [156]). The most important aspect for the design of the across-wind section is the blockage ratio. The across-wind section is $6H$ wide and $3.6H$ high. The resulting blockage ratio is 0.93 %, compared to 1.2 % in the wind tunnel experiment. The building distance to the inlet boundary is $3.6H$. The outlet boundary is positioned at $8H$ distance from the back of the structure.

The air properties are kept at normal condition of an assumed air temperature $T_{air} = 20°C$, density $\rho_{air} = 1.2\ Kg/m^3$, and a kinematic viscosity of $\nu_{air} = 1.51e^{-5}\ m^2/s$. The inlet boundary has an imposed turbulent



Inlet	transient velocity
Outlet	imposed pressure
Wall	slip condition
Top	slip condition
Structure	no-slip condition
Bottom	no-slip condition

Figure 7.2: Computational domain and boundary conditions for ABL flow simulation around a highrise building. The H is defined as the building height.

velocity. A numerical wind generator is used to reproduce the ABL flow. The fluctuating component in the velocity field is generated by a wave superposition based method (Mann [161]). The wind field can be represented as a generalized Fourier-Stieltjes integral of its spectral components. The necessary factorization (i.e., 'square root') of the spectral tensor can be accomplished in closed form (Abodonya [16]). Further readings about the theory and usage of the numerical wind generator can be found in Abodonya [16], Mann [161], Alsofi [162], and Andre [163].

The fluid domain is divided into zones, as shown in Figure 7.3. Each zone has a uniform grid size. The grid is refined as it gets closer to the structure. Grid data is summarized in Table 7.1.

The cell size near the structure is $\approx 3.75e^{-3}$ meters on the finest grid (h_1). The grid resolution is defined based on recommendations of Ricci et al. [155] and Tominaga et al. [164]. The grid size near the structure is designed to result in a $y^+ \leq 10$ for the fine grid h_1 and $y^+ \leq 20$ for the coarse grid h_4. Moreover, the grid is coarsened as proceeding away from the structure as shown in Table 7.1. The simulation is run for total time $T_{sim} = 570\,Sec$. The

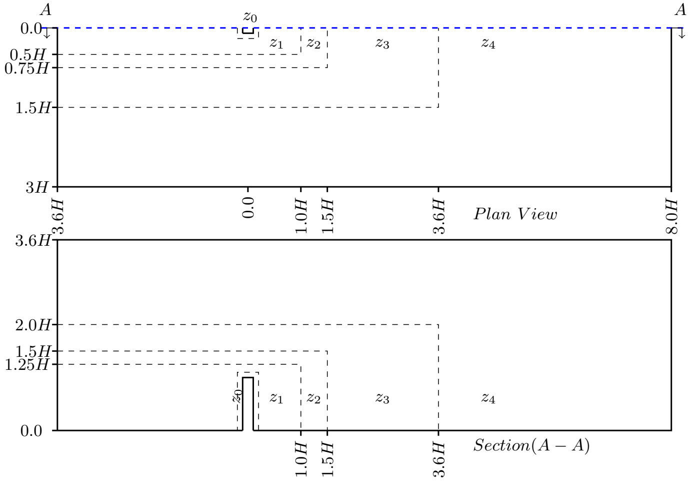


Figure 7.3: Computational grid design: top figure shown a plan view for half of the symmetric domain, while bottom figure shows cross section (A-A) indicated in the top figure

Table 7.1: Highrise Building Mesh Data: element size is normalized by H

Grid	# of cells	Zone element size $dx = dy = dz$				r
h_1	6,622,084	Str	$3.75e^{-3}$	z_2	0.04	1.00
		z_0	0.0125	z_3	0.09	
		z_1	0.025	z_4	0.25	
h_2	5,119,178	Str	$5.00e^{-3}$	z_2	0.0425	1.10
		z_0	0.01375	z_3	0.1	
		z_1	0.0275	z_4	0.25	
h_3	3,610,851	Str	$6.25e^{-3}$	z_2	0.045	1.23
		z_0	0.015	z_3	0.1125	
		z_1	0.030	z_4	0.25	
h_4	1,865,760	Str	$7.50e^{-3}$	z_2	0.0475	1.50
		z_0	0.01625	z_3	0.125	
		z_1	0.0325	z_4	0.25	

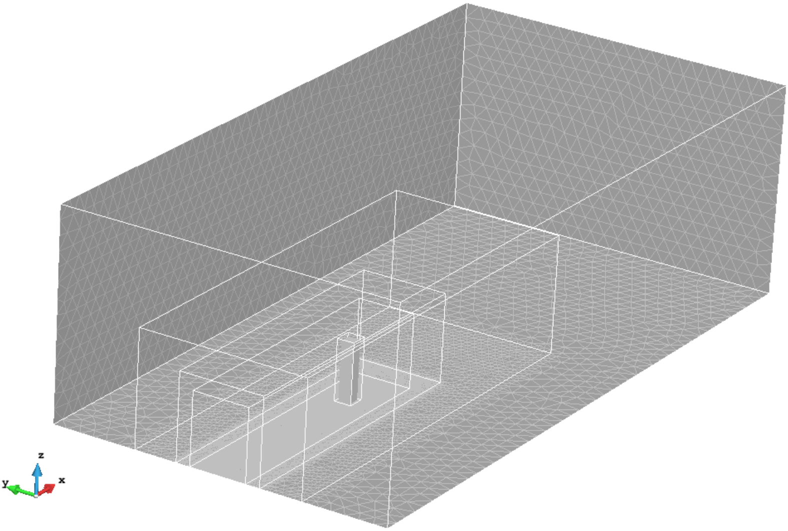


Figure 7.4: Computational grid h_4 for wind flow around a highrise building: the mesh is composed of 1.86 . 10^6 tetrahedral elements

simulation time is $\approx 10\ minutes$ as recommended by the architectural institute of japan standards, where the aeroelastic instability or vortex-induced vibration occurs within this period (AIJ [159]). Furthermore, ten minutes wind is sufficient to cover most of the flow frequencies in the micrometeorology region in the wind autospectrum (Abodonya [16]). In the simulation design, the Courant Friedrichs-Lewy (CFL) number is set to $CFL \leq 1.0$. The time step is defined from the CFL number as:

$$\Delta t = \frac{CFL * dx_{min}}{U_{Z_{ref}}} \tag{7.3}$$

This leads to a time step size $\Delta t \approx 0.07\ s$ for the finest grid. In order to obtain a higher temporal resolution, the time step is defined for the finest mesh (h_1) as $\Delta t = 0.01\ s$. Furthermore, the time step is coarsened with the same rate as the spatial resolution.

7.3 Simulation Quality Assessment

In this section, the newly proposed framework is applied to the simulation of the ABL flow around a highrise building. Four main steps are required to judge the quality of a wind simulation:

1. Code verification: The code verification framework was introduced in Chapter 3. Moreover, the code's functionalities, used to simulate the problem under investigation, are verified as shown in the proposed test scenarios in Chapter 5. Thus, once the needed functions are verified, it is safe to estimate the wind loads and quantify the numerical errors.

2. Design of the test case: The design of the test case under investigation is introduced in Section 7.2.

3. Selection of the quantities of interest (QOI): In simulating natural wind flow around a bluff body, it is imperative to pick quantities that represent mean values and fluctuating effects. Meanwhile, both local and integral quantities are to be considered. The measured QOI are presented in Section 7.3.1

4. Data analysis and numerical error estimation: The newly developed procedures introduced in Chapter 3 is used for solution verification and summarized in appendix C. Moreover, the methodology was tested for general fluid flow around bluff bodies in Chapter 6 to test its performance. Thus, the new methodology is used in this case for uncertainty evaluation for the QOI, as presented in Section 7.4 and Section 7.5.

7.3.1 Measurement Quantities Selection

The selection of the QOI in simulating natural wind flow is split into two cases:

1. Empty channel[1] simulation: It is performed to make sure that the numerically generated wind field is in a good agreement with the experimental results. The following data is acquired in simulating the empty channel:

   - Monitoring the stream-wise velocity $(U_{Z_{ref}})$ at the domain center at reference height (Z_{ref}). The points are located at $X \in [-3.6H, 0.0], Y = 0.0, Z = Z_{ref}$. Then, the following information is extracted from the signal:

     a) Mean $U_{Z_{ref}}$
     b) Integral length scale (L_{U_x})
     c) Turbulent kinetic energy (TKE)
     d) At some selected points, the spectral analysis is performed to check the frequency loss due to mapping of the generated wind field to the fluid domain inlet.

   - Vertical wind profile is measured along $Z \in [0.0, H]$, $X = 0.0$, $Y = 0.0$. Then, the mean wind velocity and turbulence intensity are calculated. The simulated flow characteristics are compared to the experimental flow. The numerical wind is regenerated; if the simulated flow parameters do not match the experimental flow parameters, it mainly means wind speed at reference height and turbulence intensity.

2. Performing the actual simulation: The structure response to wind load is simulated, and the following quantities are measured:

   - Monitoring the stream-wise velocity (U_{Z_ref}) at the domain center at reference height (Z_{ref}). The points are located at $X \in [-3.6H, -0.5D], Y = 0.0, Z = Z_{ref}$. Then, the following information is extracted from the signal and compared:

     a) Mean $U_{Z_{ref}}$

[1] Empty channel simulation means simulation using exactly the same setup without the presence of the structure

b) Integral length scale (L_{U_x})
c) Turbulent kinetic energy (TKE)

- Resultant force coefficients: the force coefficient is defined as:

$$C_F = \frac{F}{0.5\rho_{air} U_{Z_{ref}}^2 DH} \tag{7.4}$$

- Pressure coefficients: the pressure coefficient is defined as:

$$C_P = \frac{P - P_{inf}}{0.5\rho_{air} U_{Z_{ref}}^2} \tag{7.5}$$

  The pressure is measured at 500 taps located as shown in Figure 7.5.

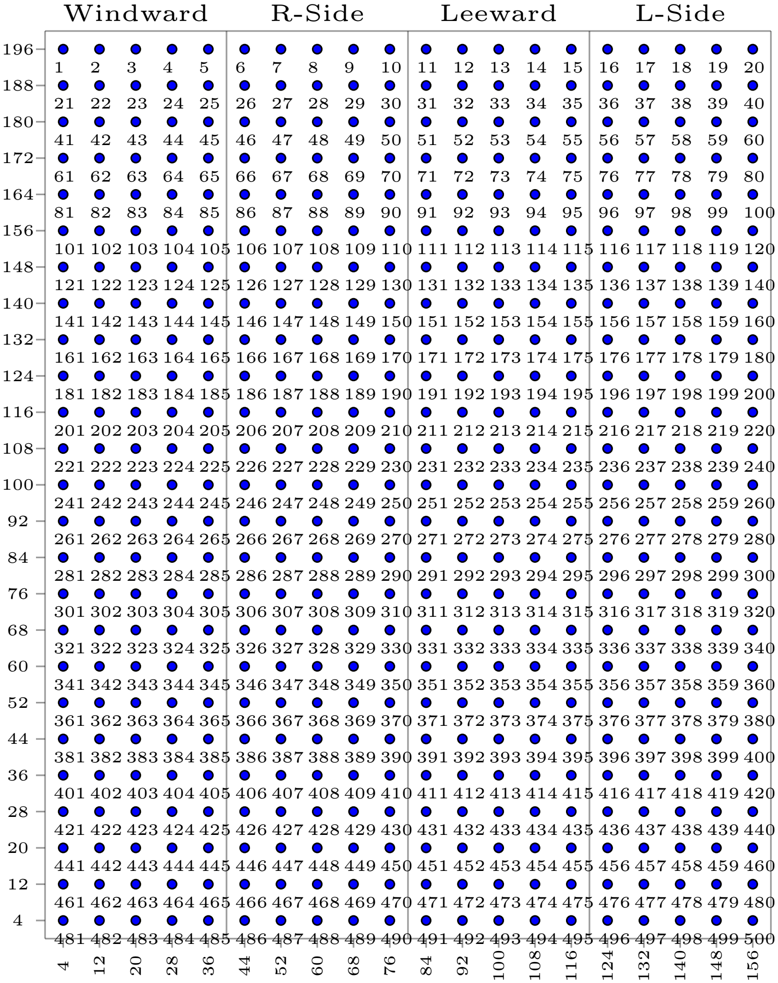


Figure 7.5: Location of pressure taps along the highrise building facade from all sides. The taps are equally spaced in both directions and all distances are in meters

7.4 Results Empty Channel

In this section, the mapping of the numerically generated wind field to the fluid domain is checked, and the effect of mesh refinement is studied. Linear mapping is used to impose the wind field on the fluid domain inlet, as shown in.

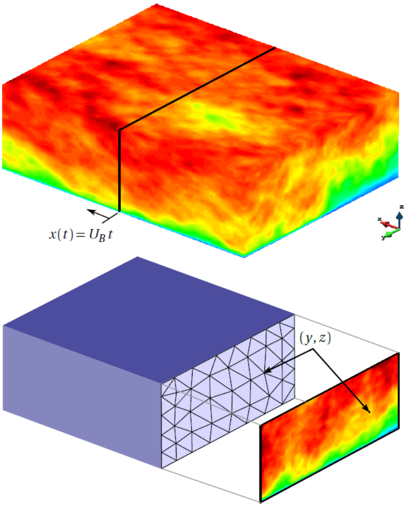


Figure 7.6: The mapping between numerically generated wind flow and computational domain having a transient inlet boundary condition (Andre [163])

The grid size has a significant effect on the preservation of the wind characteristics. Therefore, the empty channel simulation is performed first to check the quality of the simulated wind at the building location. This is very crucial in simulating ABL flows.

The stream-wise velocity is measured, and the mean wind profile is calculated. The numerically simulated wind profile is compared to the theoretical log-profile and experimental profile. The experimental profile can be found in *Tokyo Polytechnic University Aerodynamic Database* [158]. Figure 7.7 left plot shows the mean wind profile. The mean statistical uncertainty in the simulated wind field is shown to be minimal, which indicates a sufficient simulation time to converge the statistics. Meanwhile, the numerical uncertainty due to discretization is shown. It can be clearly seen that the simulated mean wind velocity is in a good agreement with the experimental wind velocity. Also, the numerical uncertainty estimation is reasonable and bounds the experimental data. The near-wall region, from the ground to $50 meters$ of the building height, is a bit deviating from the experimental values, due to the use of coarse meshes, not resolving the boundary layer and not using a wall-function. Therefore, it is very crucial to perform parametric studies and estimate the modeling uncertainties emanated from the selection of the boundary conditions and wall-functions. The evaluation of the modeling uncertainties is not in the scope of this work.

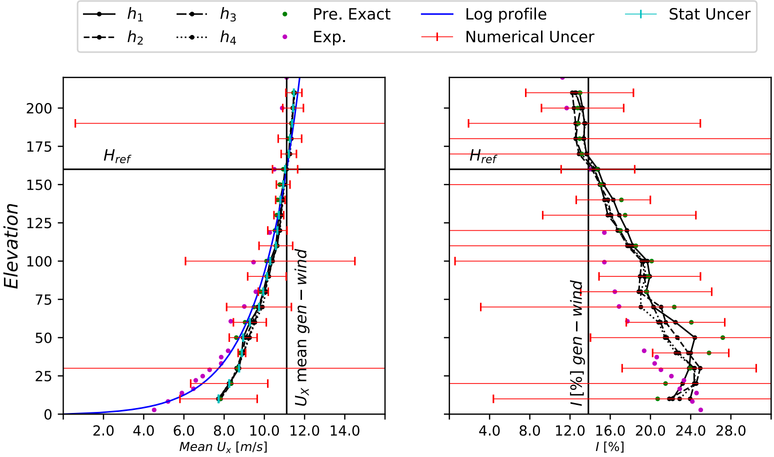


Figure 7.7: Vertical stream-wise velocity wind profile along line at $Z \in [0.0, H]$, $X = 0.0$, $Y = 0.0$. On the left is the mean velocity. On the right is the turbulence intensity.

Figure 7.7 right shows the turbulence intensity. It is crucial to notice that the numerical uncertainty is very high at some points, because of the random nature of the flow[2]. Turbulence intensity is a susceptible measure

[2] Further discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

that is profoundly affected by the grid size. The most important point is the Z_{ref}, where the simulated wind turbulence should match the experimental turbulence. In ideal cases, both numerical and experimental profiles should be identical, but it can not be achieved because the numerical wind generator fits turbulence at one point, the reference height. With the assessment of the numerical uncertainty of the turbulence intensity profile, it can be seen that the numerical uncertainty encloses the experimental data. The simulation over-estimates the wind turbulence at the bottom of the fluid domain because the boundary layer is not resolved at the bottom wall. Therefore, the effect of not resolving the boundary layer is highly investigated in the measurement of the QOI, such as C_p, C_D, and C_{F_Y}, discussed in Section 7.5.

Furthermore, the stream-wise velocity is measured at different locations from the inlet domain boundary to the location where the structure will be positioned. The points are located at $X \in [-3.6H, 0.0]$, $Y = 0.0$, $Z = Z_{ref}$, are equally spaced.

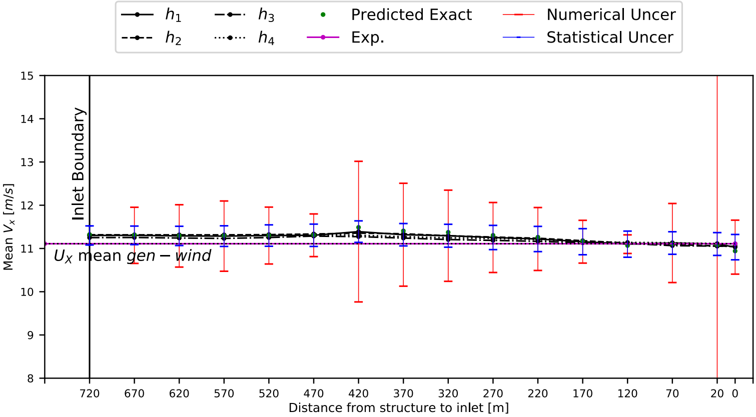


Figure 7.8: Mean stream-wise velocity (U_{Z_ref}) from inlet to structure. The points are located at $X \in [-3.6H, 0.0]$, $Y = 0.0$, $Z = Z_{ref}$.

Figure 7.8 shows the mean stream-wise velocity. It can be observed that the mean velocity is conserved from the inlet to the structure and in a good agreement with the experimental mean velocity. Both statistical and numerical uncertainty are measured. It can be noticed that the mean statistical quantity is converging very well with less than 5 % statistical uncertainty. Also, the numerical uncertainty is very reasonable and bounds the experimental value. The numerical uncertainty for most of the points is less than 10 %.

For a better judgment on the conservation of turbulence properties, both turbulent kinetic energy (TKE) and integral length scale ((L_{U_x})) are calculated from the velocity signal at each point. The effect of the inlet boundary condition can be clearly seen in Figure 7.9 at location $X = -3.6H$. The TKE is unrealistic due to the effect of the boundary condition. Figure 7.9 shows the TKE, where the numerical simulation over-predicts the TKE. The mean statistical uncertainty for TKE is negligible, which implies a converging statistic. Moreover, the numerical uncertainty is very reasonable for most of the points[3].

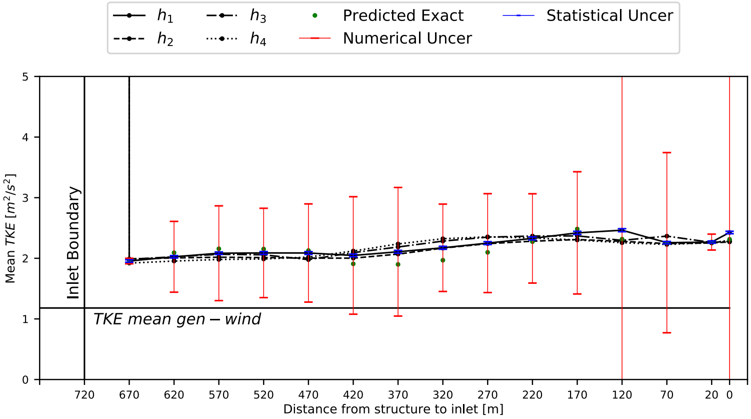


Figure 7.9: Mean turbulent kinetic energy from inlet to structure. The points are located at $X \in [-3.6H, 0.0]$, $Y = 0.0$, $Z = Z_{ref}$.

Figure 7.10 shows the (L_{U_x}), where the numerical simulation under-predicts the (L_{U_x}) in the regions very close to the inlet. Then, the flow's (L_{U_x})

[3] Discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

is well simulated at the region close to the structure. The (L_{U_x}) simulation is not dependent on the grid size only, but also the domain size; the domain should be big enough to accommodate large vortex size. Therefore, the distance from the inlet boundary to the structure location should be long enough to allow for the redevelopment of the flow vortices. The numerical error is estimated for the (L_{U_x}). It can be observed in Figure 7.10 that the estimation is conservative at some points where data does not have a clear convergence behavior, and large variations occur with refinement.

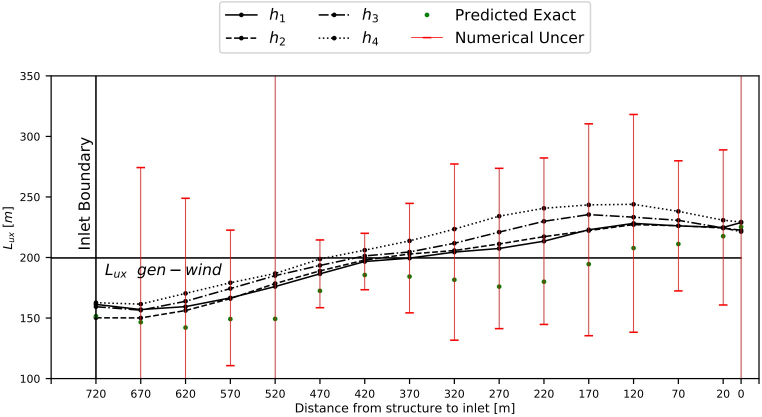


Figure 7.10: Mean integral length scale from inlet to structure. The points are located at $X \in [-3.6H, 0.0]$, $Y = 0.0, Z = Z_{ref}$.

Finally, it can be safely said that the numerically generated wind complies satisfactorily with the target profile and wind characteristics and can be used to predict the wind load on the structure under investigation.

7.5 Results Highrise

After checking the quality of the simulated atmospheric boundary layer wind field, the highrise building is simulated. The velocity field is recorded from the inlet to the structure to monitor the turbulence characteristics of the incoming wind field. Also, it is important to compare to the results obtained from empty-channel simulations. Figure 7.11 represents the mean stream-wise wind velocity.

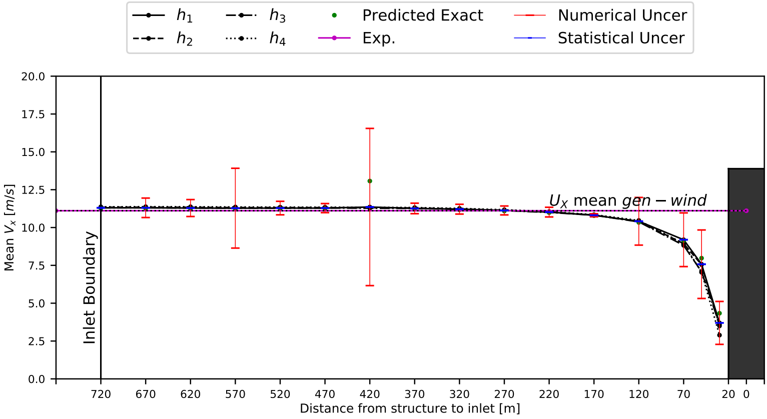


Figure 7.11: Highrise building simulation: The mean stream-wise velocity (U_{Z_ref}) is measured for points located at $X \in [-3.6H, -0.75D]$, $Y = 0.0$, and $Z = Z_{ref}$. The points are arranged from the inlet to the structure.

It can be seen that the mean wind speed is preserved and approaching zero as it gets closer to the structure. The mean statistical uncertainty is negligible ≤ 1 %, which indicates a sufficient simulation run time. Moreover, the numerical uncertainty is reasonably estimated. The performance of the LSRMONTE approach is outstanding for this QOI except for the two points located at 570 and 420. The unrealistic numerical uncertainty at these two points is due to undefined solution convergence over the four sequentially refined grids. Although LSRMONTE is not sensitive to discrete solutions in the asymptotic range, there are still some limitations on the capability of the newly proposed approach, as explained in detail in Section 8.1.2.

This unrealistic estimation can be avoided by using a biased version of the LSRMONTE. The biased LSRMONTE filters the discrete solutions for fitness to be used for the LSRMONTE uncertainty estimation.

Figure 7.12 shows the numerical and statistical uncertainty for the turbulent kinetic energy for the approaching flow at $U_{Z_{ref}}$. The effect of the boundary conditions on the simulation of the wind field can be remarked by the unrealistic error estimated at the first three points[4]. Otherwise, the numerical uncertainty is well predicted for TKE at most of the remaining points. The error in the estimation of TKE at station 170 is over-predicted because data has an anomalous behavior. Compared to the numerical uncertainty, the mean statistical uncertainty is negligible.

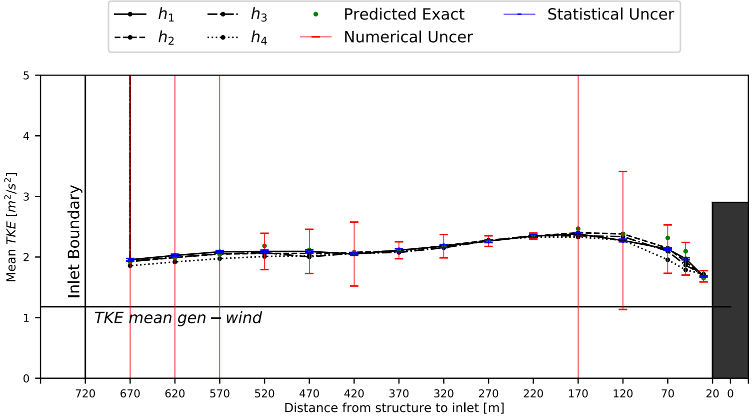


Figure 7.12: Highrise building simulation: The mean turbulent kinetic energy is measured for points located at $X \in [-3.6H, -0.75D]$, $Y = 0.0$, and $Z = Z_{ref}$. The points are arranged from the inlet to the structure

Figure 7.13 presents the L_{U_x}, where the simulated flow has an integral length scale smaller than that of the numerically generated data near the inlet boundary. Therefore, it is vital to have enough distance to the structure for the flow to develop. Based on observations from Figure 7.10 and

[4] Further discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

Figure 7.13, it can be safely said that a distance from 2.5H to 3.5H is sufficient for the redevelopment of the flow from the inlet boundary. Afterward, the flow integral length scale is higher than the numerically generated data on coarse meshes. It can be observed that the finer the mesh, the better the L_{U_x} estimation. Moreover, the numerical uncertainty is reasonably estimated for almost all the points[4], which proves the ability of the newly developed solution verification technique (LSRMONTE) to estimate errors in complex problems.

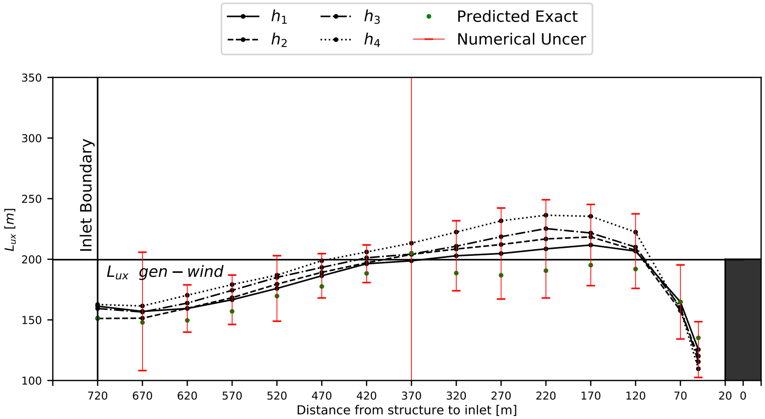


Figure 7.13: Highrise building simulation: The mean integral length scale (L_{U_x}) is measured for points located at $X \in [-3.6H, -0.75D]$, $Y = 0.0$, and $Z = Z_{ref}$. The points are arranged from the inlet to the structure

Overall, the performance of the empty channel simulations in representing the wind field is very close to that of the flow with the structure. The simulations' behavior is different in the regions away from the building and very close to the inlet boundary.

7.5.1 Force Coefficients

Simulation performance and numerical uncertainty are evaluated for the flow integral quantities. The statistical uncertainty is kept minimum for flow under investigation by running the simulation for a sufficient time and sampling frequency. Table 7.2 shows the statistical uncertainty computed for different simulations run time. Statistical uncertainty is computed with a 95 % confidence using the BCA-MBB technique introduced in Section 3.3.4.1. It can be seen that statistical uncertainty is considered negligible for the simulation under investigation when running for more than 500 $secs$.

Total Simulation Time	Statistical uncertainty for $\overline{C_D}$ [%]			
	h_1	h_2	h_3	h_4
50	23.8	25.4	23.1	23.2
100	15.5	16.4	15.8	15.1
200	6.9	6.7	7.0	6.9
300	5.4	5.5	5.0	5.0
400	4.0	4.1	3.9	3.9
500	3.2	3.0	3.4	3.1

Table 7.2: The statistical convergence uncertainty for different grid resolutions for the mean drag coefficient, $\overline{C_D}$.

Figure 7.14 (a) shows the mean drag force coefficient, $\overline{C_D}$, and the estimated numerical uncertainty. It can be observed that the numerical error is reasonably estimated and bounds the experimental solution for all the grids. The $\overline{C_D}$ has a 14.5% estimated error compared to a 7.9% exact error with the experimental results. A monotonic convergence is observed in the estimation of the numerical error for C_Y^{RMS} and C_L^{RMS} represented in Figure 7.14 (d) and (e), respectively. An oscillatory convergence is exercised for $\overline{C_D}$, C_D' and $\overline{C_Y}$ presented in Figure 7.14 (a), (b) and (c) respectively. Moreover, the LSRMONTE approach assumed a second-order convergence for all the measured QOIs. It is observed that the estimated numerical error is higher than the exact error for most of the QOIs, except for the C_D'. For the C_D', the error estimation on the grid h_4 only is higher than the exact error. Moreover, the error estimation for $\overline{C_Y}$ is too conservative, since $\overline{C_Y}$ should be zero, and the simulation results are oscillating around zero. Finally, all the results are summarized in Table 7.3.

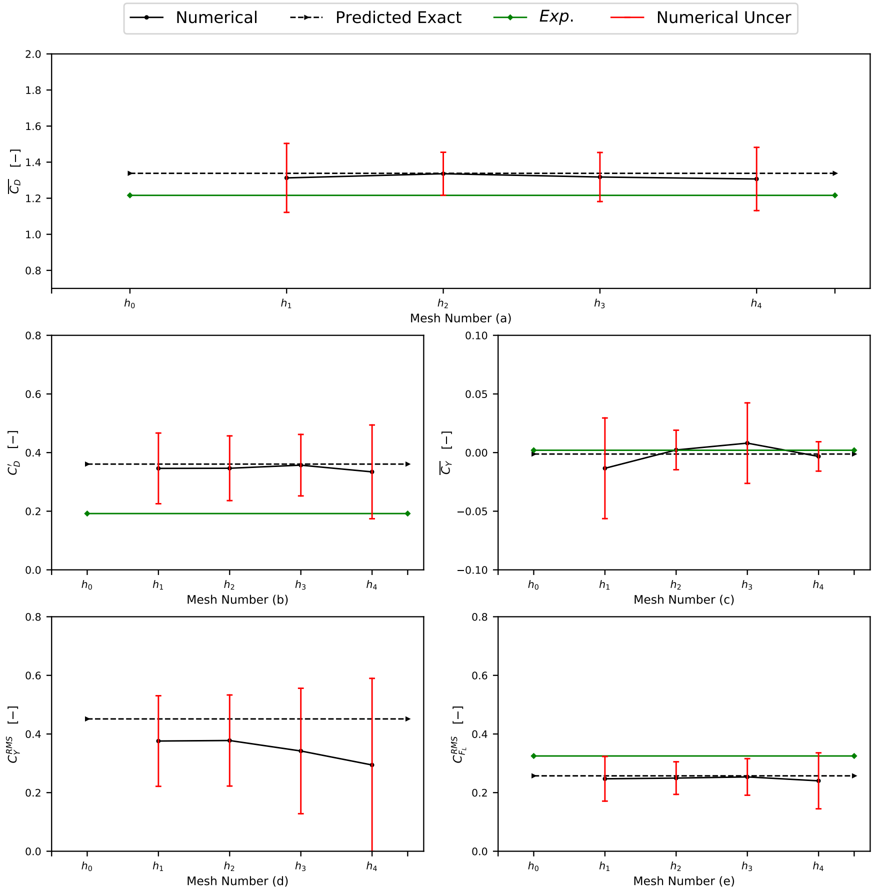


Figure 7.14: The numerical uncertainty for the mean drag force coefficient, $\overline{C_D}$, drag force coefficient standard deviation, C'_D, mean cross-wind force coefficient, $\overline{C_Y}$, RMS cross-wind force coefficient, C_Y^{RMS}, RMS lift coefficient, C_L^{RMS}.

Table 7.3: The numerical solution with the respective discretization uncertainty estimate at different grid resolutions for the mean drag coefficient, $\overline{C_D}$, drag coefficient standard deviation, C_D', mean cross-wind force coefficient, $\overline{C_Y}$, RMS cross-wind force coefficient, C_Y^{RMS}, RMS lift coefficient, C_L^{RMS}, and experimental data taken from *Tokyo Polytechnic University Aerodynamic Database* [158]

	ϕ	$\overline{C_D}$	C_D'	$\overline{C_Y}$	C_Y^{RMS}	C_L^{RMS}
h_1	ϕ_{h_1}	1.313	0.346	-0.013	0.376	0.247
	U_{Num}	± 0.191	± 0.120	± 0.043	± 0.154	± 0.076
h_2	ϕ_{h_2}	1.336	0.347	0.002	0.378	0.250
	U_{Num}	± 0.119	± 0.111	± 0.017	± 0.155	± 0.056
h_3	ϕ_{h_3}	1.318	0.357	0.008	0.342	0.254
	U_{Num}	± 0.136	± 0.105	± 0.034	± 0.214	± 0.064
h_4	ϕ_{h_4}	1.307	0.334	-0.003	0.294	0.240
	U_{Num}	± 0.175	± 0.160	± 0.012	± 0.295	± 0.096
$Exp.$	ϕ_e	1.216	0.192	0.002	—	0.325
$Pre.$	ϕ_{Pe}	1.338	0.361	-0.001	0.451	0.257
$h1$	$\epsilon_{exc.}$	7.90%	80.2%	55.0%	—	24.0%
$h1$	$\epsilon_{est.}$	14.5%	34.7%	100.0%	40.9%	30.7%

7.5.2 Pressure Coefficients

Pressure coefficients (C_P) are the most sensitive parameter to measure for the evaluation of simulation quality. The LSRMONTE approach is used for the estimation of the numerical error for the mean pressure coefficient ($\overline{C_P}$). Figure 7.15 shows the locations of the lines S1, S2, and S3 that are used to thoroughly investigate the (C_P) distribution on the structure.

Figure 7.16 shows the flow features around the simulated highrise building. The domain is colored with the instantaneous velocity field at $T = 209\ seconds$. Moreover, the structure is colored with the magnitude of the pressure coefficients C_P. It can be clearly seen that positive $C_P - values$ are observed in the windward face and suction on the right-side of the structure. More discussion is presented in subsequent paragraphs using point-wise comparisons.

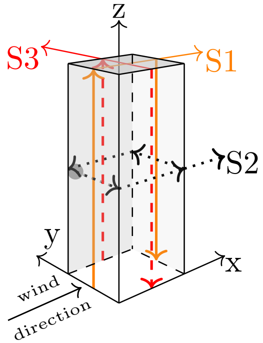


Figure 7.15: Highrise building geometry showing the locations of the (C_P) measurement lines

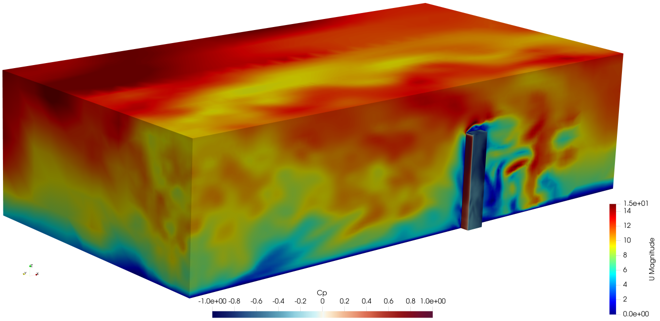


Figure 7.16: CWE simulation domain for flow around a Highrise building: the domain is colored by instantaneous stream-wise velocity and the structures is colored by (C_P)

Figure 7.17 shows the $(\overline{C_P})$ measured at the center-line of the windward side of the building. It can be observed that the numerical simulation under-estimates $(\overline{C_P})$ on the building's top region, $120 < H < 200$. On the other hand, $(\overline{C_P})$ is over-estimated closer to the building's bottom, where strong turbulence occurs in the near-wall region. Also, the estimated numerical uncertainty is too conservative at some of the regions where flow vortices are formed[5]. Having a declining (C_P) as the building's bottom is approached can be explained through the formation of the horseshoe vortex (Ricci et al. [155] and Randerson [165]).

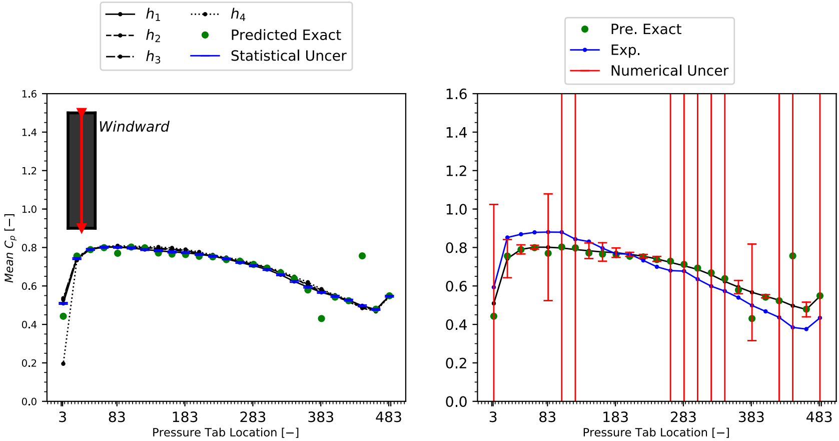


Figure 7.17: Mean pressure coefficients $(\overline{C_P})$ measured along S1 line at the windward building side. Left figure shows the estimated statistical uncertainty for grid h_1. Right figure shows the predicted discretization uncertainty for grid h_1 and the results are compared to experimental data.

Having higher pressures at the building's top and bottom, near the ground, is caused by the vertical component of the velocity gradient that characterizes the approaching boundary layer (Ricci et al. [155]). Ricci et al. [155] states that "This pressure gradient drives the flow downwards close to the windward surface of the high-rise building and, as it approaches the ground

[5] Further discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

surface, deviates the flow upwind. This is considered the mechanism responsible for the instability of the incoming boundary layer (Randerson [165]), which also controls the position of the horseshoe vortex core."

Furthermore, Figure 7.18 shows the pressure distribution along the centerline at the leeward side of the building. The $(\overline{C_P})$ is in a very good agreement with the experimental values. Meanwhile, the numerical uncertainty is reasonably estimated and bounds the experimental data. Only towards the bottom of the building, near the ground, the numerical error estimation is too conservative. In Figure 7.17 and Figure 7.18, the statistical uncertainty is negligible, because the simulation runs for a sufficient period of time.

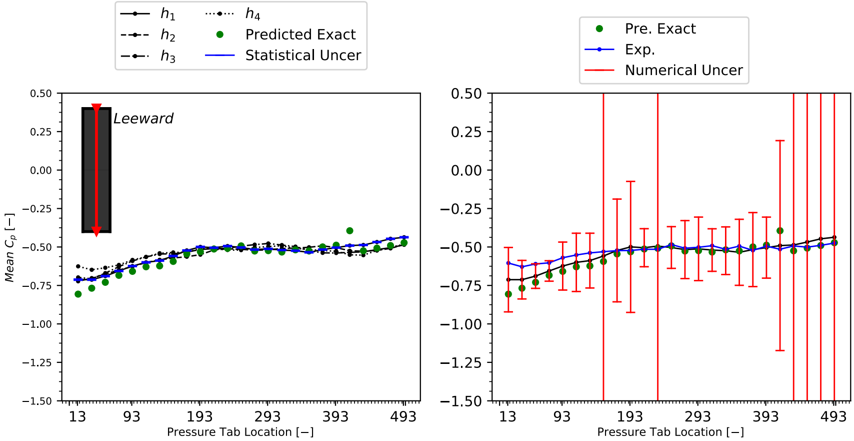


Figure 7.18: Mean pressure coefficients $(\overline{C_P})$ measured along S1 line at the leeward building side.

Figure 7.19 shows the $(\overline{C_P})$ measured at the mid-height along the four sides of the high-rise building. The performance of the LSRMONTE technique in estimating the numerical error is outstanding. Also, it is a bit conservative near the edges where flow detachment occur[6]. Moreover, the statistical uncertainty is negligible, and the numerical results are in alignment with the experimental data.

[6] Further discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

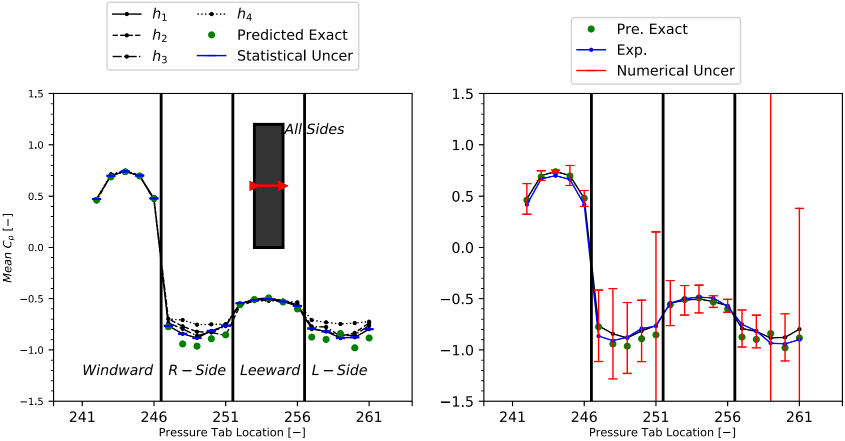


Figure 7.19: Mean pressure coefficients $(\overline{C_P})$ measured along S2 line. Left figure shows the estimated statistical uncertainty for grid h_1. Right figure shows the predicted discretization uncertainty for grid h_1 and the results are compared to experimental data.

Figure 7.20 and Figure 7.21 show the $(\overline{C_P})$ over the center-line on the lateral sides of the building. It is very interesting to observe that the numerical error estimation reflects the flow complexity. It can be seen that the estimated numerical error is too conservative at a number of points, especially those located near the building's top[7]. This path, in particular, is located in a very challenging zone for numerical simulation. Flow separation and reattachment are expected at different locations along the lateral sides. Furthermore, the highest suction is recorded at the lateral sides of the building. Both Figure 7.20 and Figure 7.21 show very small statistical uncertainty.

Figure 7.22 and Figure 7.23 compare the numerical $(\overline{C_P})$ distribution on all surfaces of the building to the experimental data. The numerical $\overline{C_P} - values$ are in a good agreement with experimental values. The numerical solution over-estimates $(\overline{C_P})$ at some locations on the windward surface (Figure 7.22). In contrast, the numerical solution under-estimates

[7] Further discussion on having too conservative numerical uncertainty can be found in Section 8.1.2

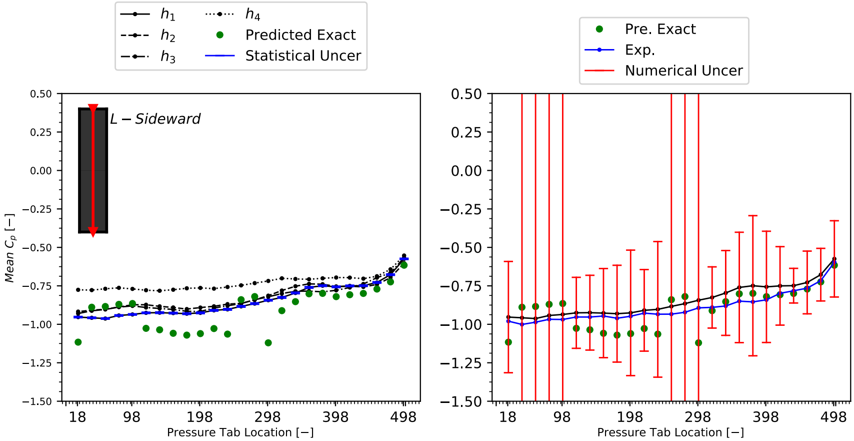


Figure 7.20: Mean pressure coefficients $(\overline{C_P})$ measured along S3 line at the left building side. Left figure shows the estimated statistical uncertainty for grid h_1. Right figure shows the predicted discretization uncertainty for grid h_1 and the results are compared to experimental data.

the $(\overline{C_P})$ at some locations on the leeward surface (Figure 7.22). These locations are approximately the top and bottom 25% of the building's height. Furthermore, the numerical simulation under-estimates the $(\overline{C_P})$ at some locations on both lateral surfaces (Figure 7.23). The lateral surfaces experience the most complex flow for a numerical simulation.

Moreover, the estimated numerical error is presented in Figure 7.24. The numerical error is estimated using the LSRMONTE technique. The error map shows a need for refinement at the top and bottom 25% of the building's height. The results shown in Figure 7.24 are for original proposed LSRMONTE with 99.7% confidence interval (CI) for uncertainty estimation. The uncertainty estimator can be tuned to be less conservative, using 68.3% CI, but still representative as shown in Figure 7.25. For a general tuning of the uncertainty estimator, a large number of simulations is required and it might be different from one simulation case study to another. Thus, it is highly recommended to start the evaluation with the original LSRMONTE. Therefore, it can be safely stated that the LSRMONTE is performing well.

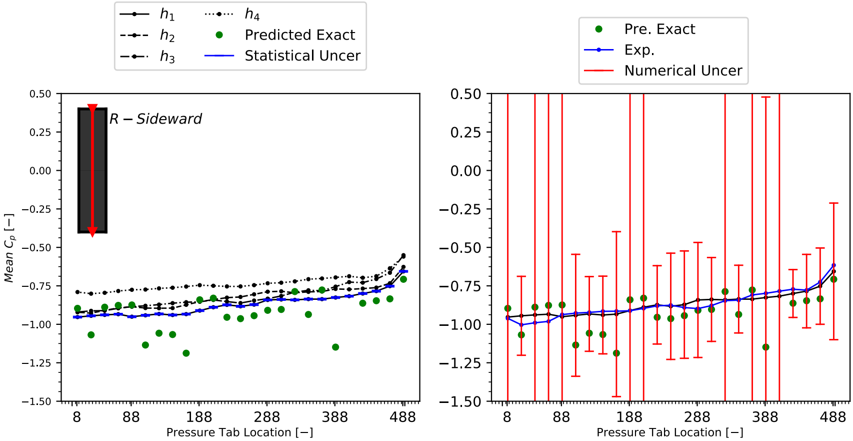


Figure 7.21: Mean pressure coefficients ($\overline{C_P}$) measured along S3 line at the right building side. Left figure shows the estimated statistical uncertainty for grid h_1. Right figure shows the predicted discretization uncertainty for grid h_1 and the results are compared to experimental data.

This conclusion is also supported by the exact error computed with experimental data. The exact error is presented in Figure 7.26. It can be seen that approx. 92% of the points have en exact error ≤ 20%. In general, the results show that refining the top and bottom 25% of the building's height improves the simulation quality. Also, the current grid size (h_1) is adequate for the objectives of this research.

Finally, the estimation of the numerical uncertainty for all the QOI shows an acceptable simulation quality. It can be observed on all the figures that some of the points are having a too conservative numerical uncertainty because the performance of the LSRMONTE is highly dependent on the data behavior as discussed in Section 8.1.2.

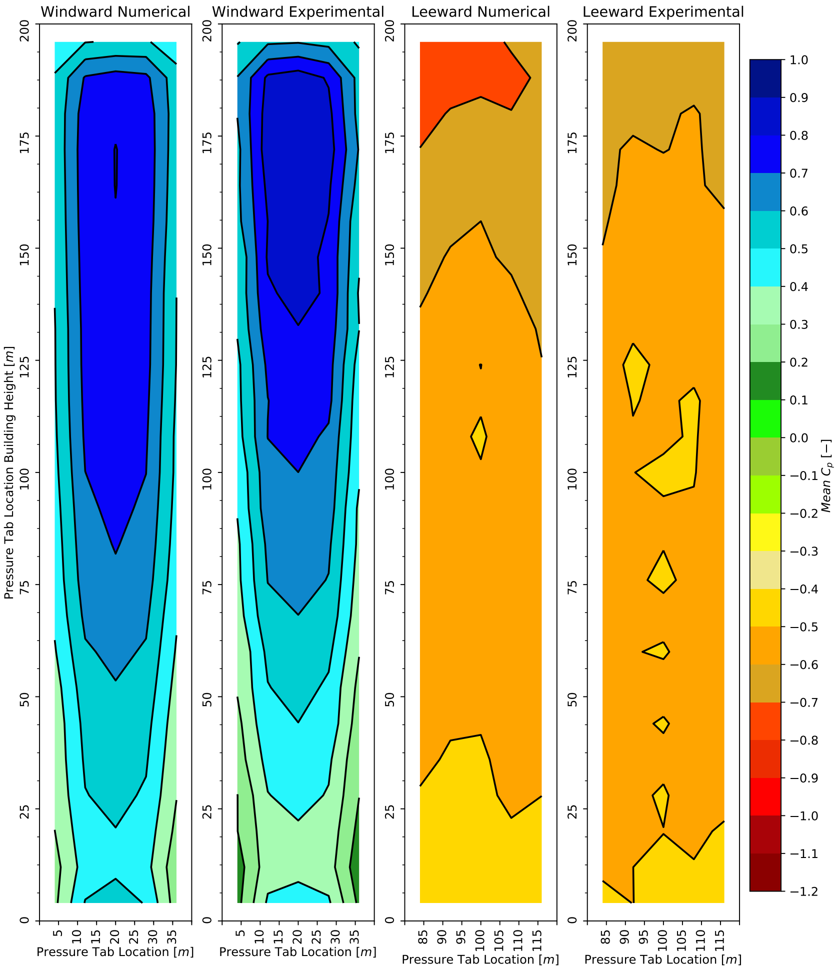


Figure 7.22: Mean pressure coefficients $(\overline{C_P})$ distribution on the windward and leeward faces of the high-rise building: Numerical and Experimental. The y-axis represents the building height (m) and x-axis represents side distance (m). The color map shows the $(\overline{C_P})$

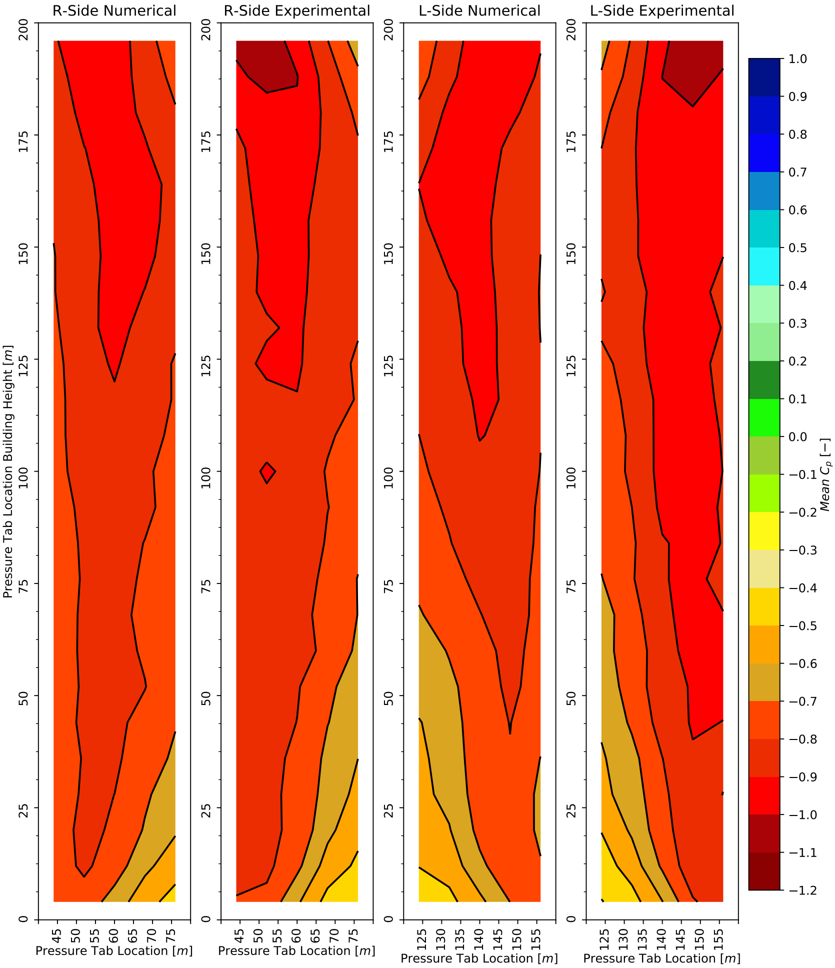


Figure 7.23: Mean pressure coefficients $(\overline{C_P})$ distribution on the right and left faces of the high-rise building: : Numerical and Experimental. The y-axis represents the building height (m) and x-axis represents side distance (m). The color map shows the $(\overline{C_P})$

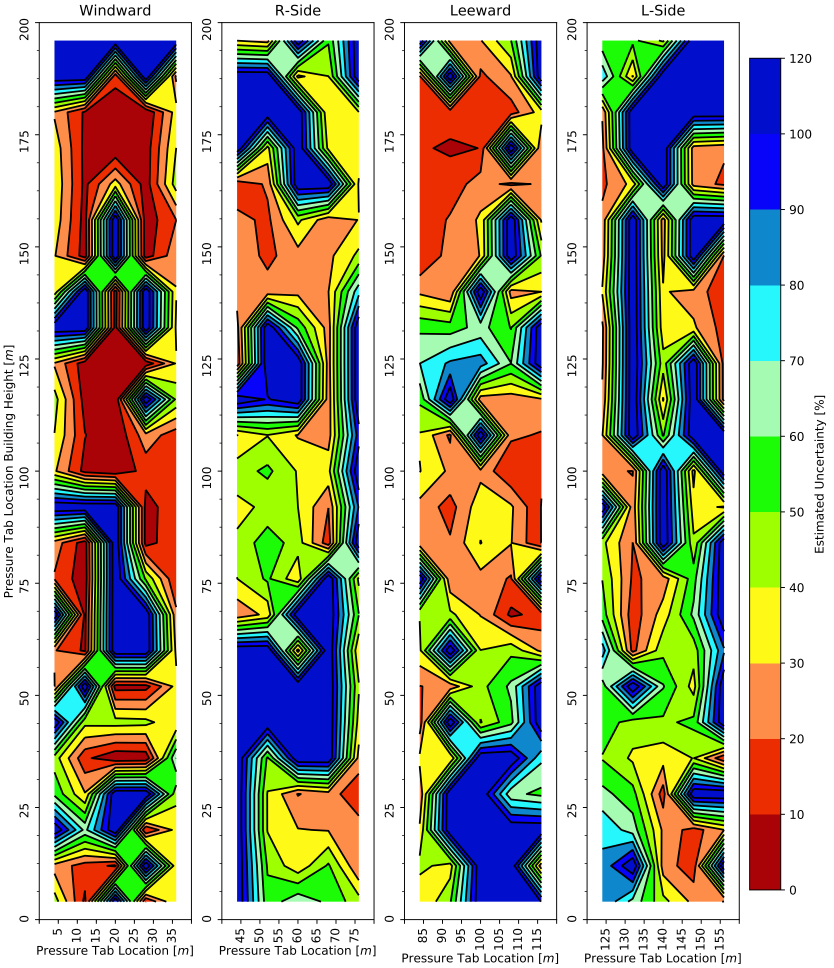


Figure 7.24: Estimated numerical uncertainty of the mean pressure coefficients ($\overline{C_P}$) distribution on all faces of the high-rise building: the color bar indicates the estimated uncertainty using LSRMONTE with 99.7% CI uncertainty estimator

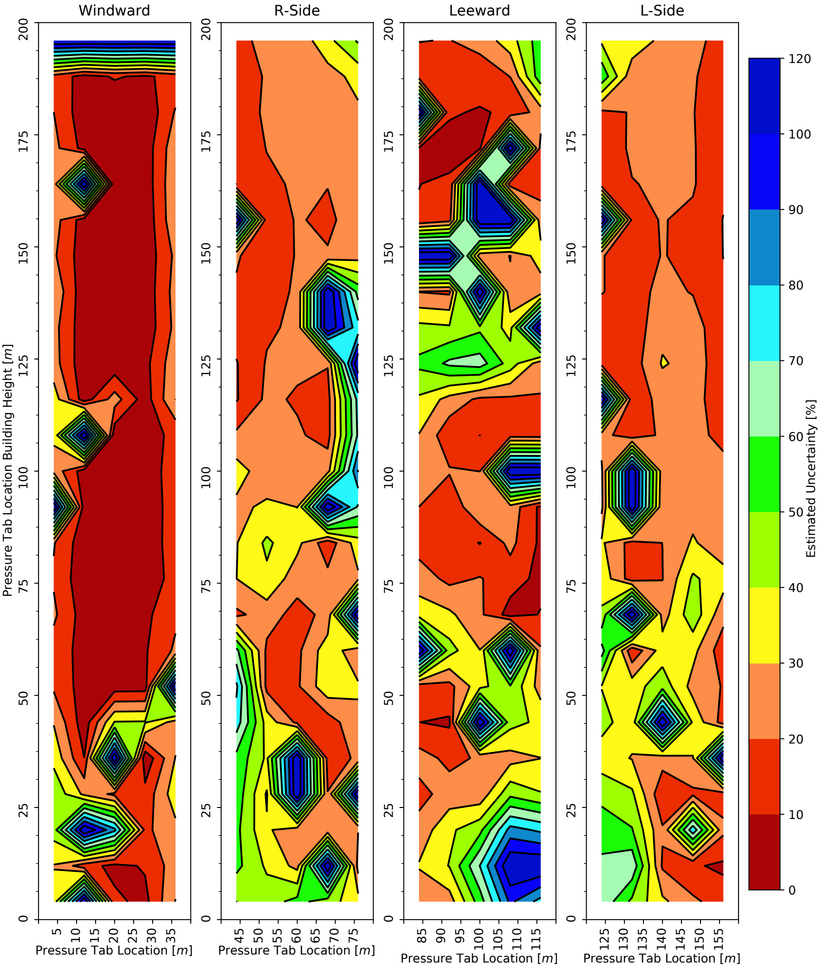


Figure 7.25: Estimated numerical uncertainty of the mean pressure coefficients ($\overline{C_P}$) distribution on all faces of the high-rise building: the color bar indicates the estimated uncertainty using LSRMONTE with 68.3% CI uncertainty estimator

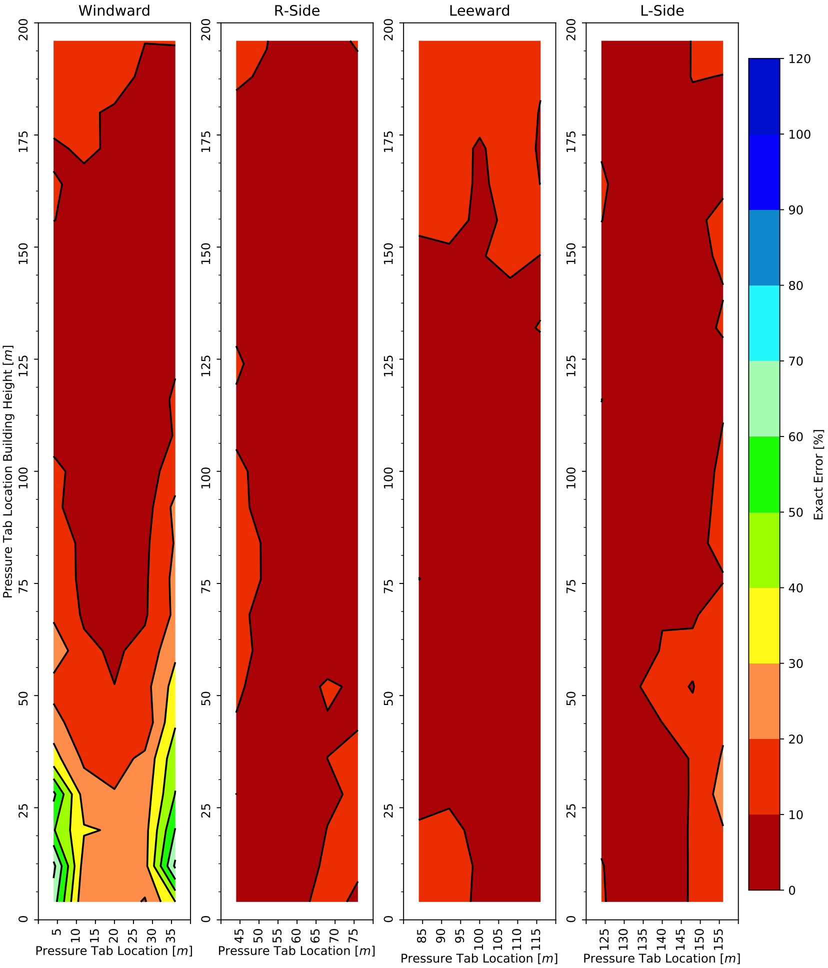


Figure 7.26: Error of the mean pressure coefficients $(\overline{C_P})$ compared to experimental data on all faces of the high-rise building: the color bar indicates the calculated error from experimental results

There is no reason anyone would want a computer in their home.

Ken Olsen, founder of Digital Equipment Corporation, 1977

CHAPTER 8

Discussion, Conclusion and Outlook

8.1 Discussion on the Verification Methodology Performance

In this section, the performance of the newly proposed verification methodology's different components is discussed after assessing the performance in Chapter 5, Chapter 6 and Chapter 7.

8.1.1 Code Verification

Code verification is the first building block of the developed methodology. The developed framework and benchmarks precisely evaluate the abilities and limits of the CFD simulation code under investigation. The hierarchical order of the developed benchmarks has been used successfully to detect bugs and reveal possible improvements of the software, which influence the order of accuracy.

8.1.2 Solution Verification

The solution verification procedures are mainly developed to assess the mapping effects in simulating *atmospheric boundary layer (ABL)* flows, quantify the statistical convergence uncertainty, and estimate discretization uncertainty.

- Assess mapping effects: the assessment is performed by computing and monitoring changes of flow parameters such as mean wind speed, turbulent kinetic energy, and integral length scale at different stations located along the upstream flow. The monitoring approach is mainly proposed to assess numerical ABL wind flow on an empty domain to ensure the mean wind speed's conformity. Moreover, Moreover, the integral length scale assessment shows the breakup in the flow structures near the inlet and the flow development as it travels in the upstream region. The unrealistic value of the turbulent kinetic energy at the inlet boundary indicates the mapping effect. Thus, it is imperative to have a sufficient upstream distance to minimize the artificial effects of mapping and allow for flow redevelopment.

- Statistical convergence error: the importance of quantifying the statistical convergence error with uncertainty evolves from having unsteady turbulent flow. The biased-corrected accelerated moving block bootstrap (BCA-MBB) is a powerful tool for computing statistics with uncertainty, which saves the computational power required to repeat the simulations many times to obtain meaningful statistics. Moreover, quantifying statistical convergence can help answer the question: Is the simulation time sufficient for converging statistics? Is the sampling frequency sufficient? The performance of the BCA-MBB is not guaranteed when computing extreme value statistics with uncertainty.

- Discretization error estimation: In this work, the discretization error estimation consists of both error and uncertainty estimation. Thus, it is also called discretization uncertainty estimation (LSRMONTE). LSRMONTE is developed with particular emphasis on dealing with non-monotonically converging data and defining the error estimator's uncertainty based on the uncertainty in the parameters defining the error term. Figure 8.1 shows a schematic representation of possible data behavior for discrete solutions with mesh refinement. The performance of all the available discretization uncertainty estimators is the same for monotonically converging data in Figure 8.1a. The LSRMONTE showed a less conservative estimate compared to the state of the art methods, but still reliable especially when having anomalous data behavior as shown in Figure 8.1b, 8.1c, 8.1e, and 8.1f.

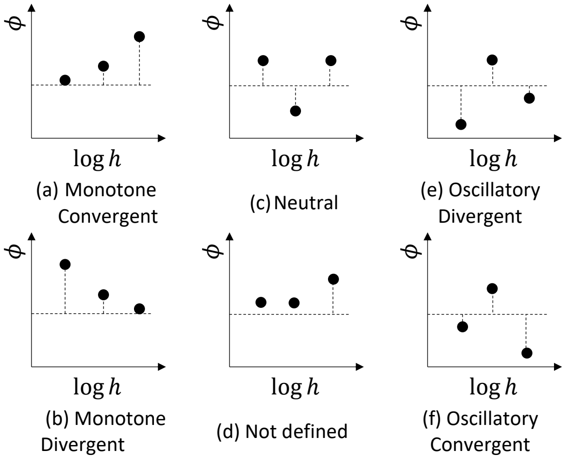


Figure 8.1: Schematic representation of possible behavior for computed solutions under mesh refinement. The y-axis represents the numerical solution (ϕ). The x-axis represents the log of the typical grid size (h). The dashed lines are intended to guide the eye.

The behavior of the LSRMONTE is extremely conservative to be realistic when data behavior is not defined, as shown in Figure 8.1d. Not-defined data behavior is the main reason for unrealistic estimation using LSRMONTE. A data behavior is not-defined when two sequentially refined grids produce the same result, which implies the need for a higher refinement ratio. To study the not-defined data behavior, let us investigate the mean pressure coefficient ($\overline{C_P}$) at point 258 located along line S2 in the highrise building simulation presented in Section 7.5.2. Figure 8.2 right shows the unrealistic uncertainty estimation, using LSRMONTE, when having a discrete solution is approximately the same for grids h_2 and h_1. The unrealistic estimate results from a numerical instability when performing curve fitting for the flat part of the cure formed by solutions on grids h_2 and h_1. To overcome this behavior, a biased version of the LSRMONTE is defined where only three data points are used to estimate the error and the discrete solution of grid h_1 is ignored. The newly defined version is called biased because it ignores part of the data. Figure 8.2 left shows the performance of the LSRMONTE biased solution. In the biased solution, the performance of the LSRMONTE has an outstanding improvement.

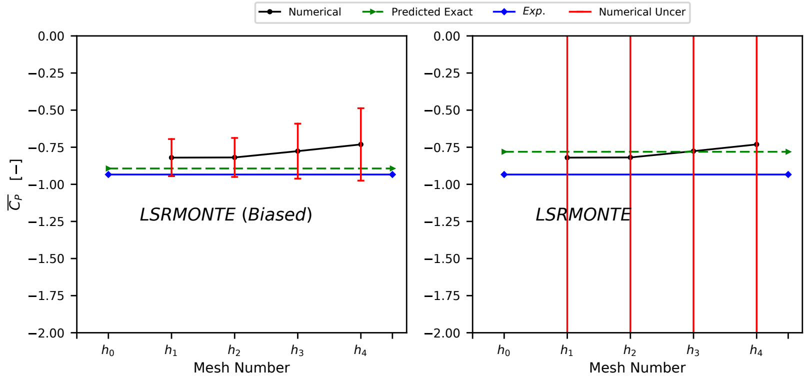


Figure 8.2: Discretization uncertainty estimation, LSRMONTE, for Mean pressure coefficient ($\overline{C_P}$) point 258 located along line S2 in the highrise building simulation presented in Section 7.5.2 for four sequentially refined grids

To study the performance of the LSRMONTE over the state of the art procedure (LSR1), see Section 2.2.1.5, the interquartile range (IQR) for the error is calculated for the 500 pressure tabs located on the highrise building simulated in Chapter 7. Figure 8.3 shows the IQR box plot. It can be seen that the median of the error using LSR1 is the highest. Also, the dispersion of the data is very high, which implies the high conservativeness of the estimator, knowing that most of the data are non-monotonically converging. Moreover, LSRMONTE_BIASED is providing better results than LSRMONTE because the data with unrealistic uncertainty is treated. It can be seen that LSRMONTE has a much better median and very low dispersion compared to LSR1, and LSRMONTE_BIASED has a bit better median and lower dispersion than LSRMONTE.

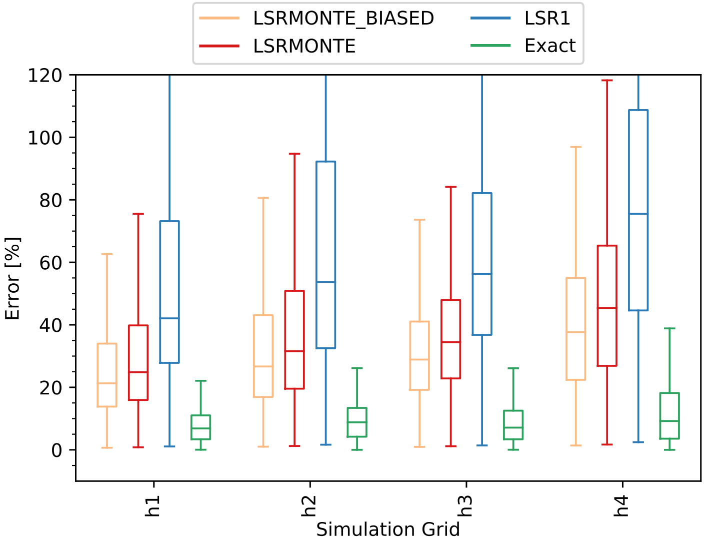


Figure 8.3: Interquartile range (IQR) box plot for the error calculated at 500 pressure tabs located on the highrise building simulated in Chapter 7

Overall, it can be concluded that the LSRMONTE has an outstanding performance with the limitation of dealing with data having an undefined convergence.

8.2 Conclusions

In this thesis, an integrated, flexible, and modular framework was developed for quality assessment of *Computational Wind Engineering* (CWE) simulations. The new development was applied to a finite element-based fluid solver with an emphasis on eddy-resolving models. The particular application was the simulation of *Atmospheric Boundary Layer* (ABL) flow around a high-rise building. In this context, the assessment of the numerical results from simulating natural wind flow using eddy-resolving models was the main methodical focus.

The developed framework consists of two main steps: code verification and solution verification. After selecting the computational models to be used, code verification was performed to ensure the correctness of the code implementation. Moreover, the application of interest was to be simulated, and *Quantities of Interest* (QOI) were to be selected for the intended application. Solution verification activities were performed on the QOI on different grid levels to estimate the numerical uncertainty. The complete framework has been introduced in Chapter 3.

The developed stringent code verification procedure for assessing simulation code with a hierarchical manner was presented in this thesis. The increasing complexity of the proposed benchmarks, Section 3.2, has been successfully used to detect code deficiencies, and propose possible improvements to the code, which influence the code's order of convergence (Chapter 5). The order of accuracy tests using the *Method of Manufactured Solutions* (MMS) ensured their ability to form a rigorous and strict tool for software consistency testing. The developed benchmarks were not meant to test the code's physical efficiency but the numerical code implementation. The development of the test cases is software-independent. The assessed CWE software Kratos Multi-Physics showed second-order convergence in space and time. Due to the linearization of the $WBZ{-}\alpha(Bossak)$ time integration scheme implemented in the code, the formal temporal order of convergence dropped to a lower level than the scheme could theoretically achieve. The convergence of the $WBZ-\alpha(Bossak)$ scheme was conditionally dependent on the selection of the time step size. Theoretically speaking, the error is decreasing with smaller time step size, which is not always achieved in the $WBZ{-}\alpha(Bossak)$ time discretization scheme implemented in Kratos Multi-Physics. However, it is always right when using the $BDF2$ time discretization scheme on the same code. Therefore, it can be safely concluded that the tested functionalities in KRATOS Multiphysics have a second-order convergence.

The developed solution verification methodology consists of a sequence of steps that has been introduced in Section 3.3. The error estimation has two components: statistical uncertainty and numerical uncertainty. The statistical uncertainty has been introduced in Section 3.3.4. The numerical uncertainty estimation technique, introduced in Section 3.3.5, puts a great emphasis on the definition of the parameters involved in the error evaluation using *Richardson Extrapolation* based on power series expansion. The development has focused mainly on the definition of the grid size (Section 3.3.5.1), error estimation for non-monotonically converging data, and the discretization uncertainty estimation (Section 3.3.5.3). The *Least-Square Root with Monte Carlo Sampling* (LSRMONTE) has been introduced for the discretization error estimation with a special focus on simulations, not in the asymptotic range and non-monotonically converging data.

The solution verification methodology has been applied to general fluid flow around a 3-D circular cylinder and 3-D square prism (Chapter 6). The performance of the solution verification approach was useful in providing insights into the simulation quality in general. More precisely, mesh quality was tested, and from the estimated numerical uncertainty, the regions where mesh improvement is needed were detected. The uncertainty estimator provides unrealistic values at points where improvement is required, especially when dealing with local quantities or integral quantities that are highly influenced by flow turbulence. Furthermore, if the simulation runs for sufficiently enough time, statistical uncertainty should be negligible. The same conclusions can be drawn from the simulation of the ABL flow around a high-rise building (Chapter 7).

It was safely concluded that simulation quality must be checked in the CWE community, even if there remains a great deal of work to be conducted in this research area. The performance of the newly proposed quality assessment guidelines is outstanding. The performance was checked when the numerically estimated uncertainty was compared to experimental results.

Finally, there is still a long way to go until simulation quality techniques are mature enough to be used in daily engineering practice. This thesis presents the first milestone for a complete reliability assessment method and overcoming challenges of CWE simulations.

8.3 Outlook

Based on the performed simulations and analysis of the results in Chapter 6 and Chapter 7, more studies are needed to build confidence in CWE, such that:

- Input and modeling parameters uncertainties must be quantified to have a complete framework for verification and validation in CWE.
- It is crucial to study the effect of the integral length scale of the numerically generated wind on the structure response. In most of the literature, the ABL flow is presented by the mean wind velocity profile and turbulence intensity, which are not sufficient to describe the natural wind flow.
- Numerical simulation codes must be rigorously tested before usage in practice, and code limitations must be clearly defined. In other words, verification and validation activities must be performed repeatedly to assure quality.
- Well-studied validation data are needed for CWE simulations.
- Representation of natural wind flows in both physical and numerical wind tunnel is not accurate. In the physical wind tunnel, there is a maximum limit for the size of the approaching vortices. In the numerical wind tunnel, small scales of turbulence are not resolved but modeled. Consequently, a more detailed study is needed to compare the limitations and advantages of both physical and numerical wind tunnels.
- The domain size in all directions has a direct effect on the quality of the results. Thus, detailed instructions are needed for the design of the domain size. The dimensions of the domain should be related to the size of the largest vortex (integral length scale).
- The identification of mesh quality is highly dependent on the quantity of interest. For the same grid, solution quality can be very good for an integral quantity, such as the mean drag force coefficient, and can be very bad for another quantity, such as the root mean square for the cross-wind force coefficient.

- The total sampling time and sampling frequency are crucial in identifying the statistical convergence and quality of the results.
- The results obtained from the LSRMONTE can be used for the design of the finer grid with higher quality, where the weaknesses in the coarse grids are identified.

Finally, more work is required in developing benchmarks for validating CWE simulations, uncertainty quantification for input and modeling parameters, and optimization of the domain size. Moreover, real-time measures for the convergence of the simulation quantities of interest are needed to save computational time and apply verification methodology for all numerically simulated structures.

Appendices

APPENDIX A

CALCULATION OF STATISTICS

The definitions of the statistics for a general variable (x) time series $\left[x_n\right]_{n=0}^{N-1}$, with length N, as presented in this thesis are given below.

A.1 Mean

The mean of the variable is computed as

$$\overline{x} = \sum_{n=0}^{N-1} \frac{x_n}{N}. \tag{A.1}$$

A.2 Standard Deviation

The variable standard deviation is computed as

$$\sigma_x = \sqrt{\sum_{n=0}^{N-1} \frac{|x'_n|^2}{N}}. \tag{A.2}$$

A.3 Root-Mean Square

The RMS of the variable is computed as

$$x^{RMS} = \sqrt{\sum_{n=0}^{N-1} \frac{x_n^2}{N}}. \tag{A.3}$$

A.4 Pearson Autocorrelation

The autocorrelation at the time delay $\tau = m\Delta t$ is computed as

$$\rho_x^m = \frac{1}{\sigma_x^2 N} \sum_{n=0}^{N-1} x_n' x_{m+n \ mod \ N}'. \tag{A.4}$$

A.5 Spectral Density

The discrete Fourier transform of the velocity fluctuation is computed as

$$\hat{x}_m' = \frac{\sqrt{2T}}{N} \sum_{n=0}^{N-1} x_n' \exp\left(-\frac{2\pi imn}{N}\right), \quad m = 0 \ldots N-1. \tag{A.5}$$

The spectral density at the frequency $f = m/T$ is given by

$$S_x^m = \overline{|\hat{x}_m'|^2} \tag{A.6}$$

and is estimated by dividing the total time series into sub-intervals (usually 8 or 16), computing Equation (A.5) on each sub-interval and averaging the results (Andre [163]).

Appendix B

Calculation of Flow Parameters

The definitions of the turbulence parameters for a time series of wind velocity as presented in this thesis are given below.

B.1 Turbulent Kinetic Energy (TKE)

The TKE for a 3-D velocity field $\boldsymbol{u}_{(x,y,z,t)}$, having $\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u}'$ is computed as

$$TKE = \frac{1}{2}\overline{\boldsymbol{u}'^2}. \tag{B.1}$$

B.2 Turbulent Kinetic Energy Dissipation Rate (TKEDR)

The TKEDR is defined as the gradient of the TKE and is computed as

$$TKEDR = -\frac{d(TKE)}{d(t)}. \tag{B.2}$$

The gradient is computed using a central difference scheme.

B.3 Integral Time and Length Scales

The integral time scale (Ts) and integral length scale (Ls) are defined for the longitudinal turbulent wind speed (u) as

$$Ts = \int_{\tau=0}^{\tau_{zero}} \rho_{(\tau)} d(\tau), \quad \tau_{zero} \text{ is the time lag } (\tau) \text{ for } \rho_{(\tau)} = 0 \tag{B.3}$$

The autocorrelation ($\rho_{(\tau)}$) can be evaluated as shown in Section A.4. Then, the integral length scale is computed as

$$Ls = \overline{u}.Ts. \tag{B.4}$$

B.4 Courant–Friedrichs–Lewy Number (CFL)

The CFL is computed from the cell size dx and time step size Δt as

$$CFL = \frac{dx}{\Delta t} \tag{B.5}$$

B.5 Estimation of y^+

The y^+ can be estimated using the density ρ, absolute viscosity μ, cell size near wall (wall distance) dx and friction velocity u_* as *Yplus wall distance estimation* [166]:

$$y^+ = \frac{dx\,\rho\,u_*}{\mu} \tag{B.6}$$

the u_* is evaluated as

$$u_* = \sqrt{\frac{\tau_w}{\rho}} \tag{B.7}$$

where τ_w is the wall shear stress computed as

$$\tau_w = C_f\,\frac{1}{2}\rho U_\infty^2 \tag{B.8}$$

where U_∞ is the free-stream velocity and C_f is the skin friction and is computed using the Schlichting skin-friction correlation, which is valid for $Re < 10^9$ as

$$C_f = [2 log_{10}(Re) - 0.65]^{-2.3} \tag{B.9}$$

and Re is calculated as

$$Re = \frac{\rho\ U_\infty\ L_b}{\mu} \tag{B.10}$$

where L_b is a characteristic length for the boundary.

APPENDIX C

Complete Verification Methodology

Specify the Application of Interest

ABL wind load assessment on structures using CWE simulations. Formulate the flow problem: by defining

- the objective of the computation
- the structure geometry
- the characteristics of the approaching wind
- the dimensionality of the model
- the state of the flow (steady or unsteady)
- the domain size
- the flow regime (laminar or turbulent)

Define Simulation Prerequisites

Simulation software selection and identification of the:

- Numerical method (problem formulation)
- Spatial discretization scheme
- Temporal discretization scheme
- Theoretical order of convergence in space and time
- Available boundary conditions
- Solution strategies for linear equation system

Check Correctness of Code Implementation

At this stage code verification activities, which are introduced in Section 3.2 and applied in Chapter 5, are performed as:

1. Having either exact or manufactured solutions (ϕ), use symbolic math programming to compute source terms $\hat{\boldsymbol{s}}_f^{mom}$ and $\hat{\boldsymbol{s}}_f^{mass}$ as:

$$\hat{\boldsymbol{s}}_f^{mom} = \underbrace{\frac{\partial \hat{\boldsymbol{u}}}{\partial t}}_{\text{Inertia}} + \underbrace{\nabla . (\hat{\boldsymbol{u}}\hat{\boldsymbol{u}})}_{\text{Convection}} - \underbrace{\nabla . (2\nu\hat{\boldsymbol{D}})}_{\text{Viscous}} + \underbrace{\nabla \hat{P}r}_{\text{Pressure}} \tag{C.1}$$

$$\hat{\boldsymbol{s}}_f^{mass} = \nabla . \hat{\boldsymbol{u}} \tag{C.2}$$

2. Define boundary conditions $\phi^\gamma = \hat{\phi}^\gamma = \phi_{(x=x^\gamma,t)}$ and initial condition $\phi_{(x,t=0)} = \hat{\phi}_{(x,t=0)}$

3. Here the Taylor-Green Vortex (TGV) is the base benchmark, because $\hat{\boldsymbol{s}}_f^{mom} = 0$ and $\hat{\boldsymbol{s}}_f^{mass} = 0$ and can be used to test black-box codes. The TGV is defined as

$$\hat{u}_x = e^{-2\nu t}\sin(x)\cos(y) \tag{C.3}$$

$$\hat{u}_y = -e^{-2\nu t}\cos(x)\sin(y) \tag{C.4}$$

$$\hat{P}r = \frac{\rho}{4}e^{-4\nu t}(\cos(2x) + \sin(2y)) \tag{C.5}$$

where $\nu = 0.2$ and $\rho = 1.0$

4. Define domain size, spatial and temporal resolution, and simulation time
5. Generate computational grid such as structured mesh, unidirectional structured mesh, alternate diagonal structured mesh and unstructured mesh
6. Define sampling points in space and sampling intervals in time to be consistence in error calculation
7. Perform the simulation
8. Evaluate the error for N sampling points using E_2 and E_i norms as

$$E_2 = ||\phi - \phi||^2 = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\phi_n - \hat{\phi}_n)^2} \tag{C.6}$$

$$E_i = max|\phi_n - \hat{\phi_n}|, \quad n \in [1, N] \tag{C.7}$$

9. Repeat simulation with systematic refinement for the following cases:
   - Refine in space using very small time step
   - Refine in time using very fine grid size
   - Refine in space and time simultaneously

   The following conditions must be followed:

- The refinement ratio either in space or in time must be equal to the theoretical order of convergence
- Simulation should be repeated for at least four sequential refinement steps to rigorously investigate the convergence trend

10. Calculate the observed order of convergence as

$$p = \frac{\log\left(\frac{E\left(h_{coarse}\right)}{E\left(h_{fine}\right)}\right)}{\log(r)} \tag{C.8}$$

11. If the observed matches the theoretical order of convergence for all the refinement cases shown in step 9, it can be safely said that the code under investigation is verified. Otherwise, more tests are required to debug for coding mistakes

12. These benchmarks are developed for rigorous code testing and it is only replacing the TGV exact solution defined in step 3. The benchmarks are manufactured solutions in which $\hat{\mathbf{s}}_f^{mom} \neq 0$ and $\hat{\mathbf{s}}_f^{mass} = 0$. All the previous steps are repeated for the following benchmarks:

    a) **IN**: the field terms are defined as

    $$\hat{u}_x = \sin(t) \tag{C.9}$$

    $$\hat{u}_y = -\sin(t) \tag{C.10}$$

    $$\hat{Pr} = \sin(t) \tag{C.11}$$

    where $\nu = 0.1$ and $\rho = 1.0$

    b) **PRES**: the field terms are defined as

    $$\hat{u}_x = e^{-2\nu t}\sin(x)\cos(y) \tag{C.12}$$

    $$\hat{u}_y = -e^{-2\nu t}\cos(x)\sin(y) \tag{C.13}$$

$$\hat{Pr} = \rho e^{-4\nu t}(\cos(2x) + \sin(2y)) \tag{C.14}$$

where $\nu = 0.1$ and $\rho = 1.0$

c) **CON**: the field terms are defined as

$$\hat{u}_x = e^{-2\nu t}\sin(x)\cos(y) \tag{C.15}$$

$$\hat{u}_y = -e^{-2\nu t}\cos(x)\sin(y) \tag{C.16}$$

$$\hat{Pr} = \frac{\rho}{40}e^{-4\nu t}(\cos(2x) + \sin(2y)) \tag{C.17}$$

where $\nu = 0.05$ and $\rho = 1.5$

d) **VIS**: the field terms are defined as

$$\hat{u}_x = e^{-2\nu t}\sin(x)\cos(y) \tag{C.18}$$

$$\hat{u}_y = -e^{-2\nu t}\cos(x)\sin(y) \tag{C.19}$$

$$\hat{Pr} = \frac{\rho}{4}e^{-4\nu t}(\cos(2x) + \sin(2y)) \tag{C.20}$$

where $\nu = 0.5$ and $\rho = 0.3$

13. For any of the benchmarks: if $p = p_f \rightarrow$ stop code is verified for this benchmark, else
    - check if the benchmark is correctly implemented
    - if the benchmark is correctly implemented, find code bugs and rerun the simulation, then continue with other benchmarks introduced in step 12
    - if not implemented correctly, detect and correct benchmark implementation, then repeat testing (steps 4 - 11)
14. Document results

Design and Execution of Numerical Simulations

Condition: the software must successfully pass the code verification benchmarks to be eligible for simulating the problem under investigation.

- Model the geometry and flow domain
- Establish the boundary and initial conditions
- Generate the grid
- Establish the simulation strategy
- Establish (output) sampling for the quantities of interest
- Perform the simulation for at least four sequentially refined grids in space and time simultaneously.

Solution Verification to Quantify Numerical Error

The theoretical background is introduced in Section 3.3 and applied in Chapter 6 and Chapter 7.

- Model the geometry and flow domain
- Establish the boundary and initial conditions
- Generate the grid
- Establish the simulation strategy
- Establish (output) sampling for the quantities of interest
- Perform the simulation for at least four sequentially refined grids in space and time simultaneously.
- Quantify numerical errors:
  1. *Input:*
     a) quantities of interest $\boldsymbol{\Phi}_i$ and typical grid size h_i, where $i \in \{1, 2, ..., N\}$ and N is the number of the sequentially coarsened grid ($i = 1$ denotes the finest grid and $i = N$ denotes the coarsest grid)

   b) $\boldsymbol{\Phi}_i$ is a time series

2. *Assumptions:*
   a) Iterative and convergence errors are kept minimum to be ignored
   b) $N \geq 4$

3. *Evaluation of statistics with uncertainty*
   The theoretical background is introduced in Section 3.3.4.
   - Compute statistics according to equations in appendices A and B
   - For any $\boldsymbol{\Phi}_i$, use the BCA-MBB Non-parametric bootstrapping to compute statistics with uncertainty:
     The BCA-MBB non-parametric bootstrap is presented in Efron et al. [167] as
     a) Let's call $\boldsymbol{\Phi}_i$ time series $\boldsymbol{x}$. The non-parametric inference begins with an observed data set

        $$\boldsymbol{x} = (x_1, x_2, ..., x_n), \tag{C.21}$$

        where the x_i are independent and identically distributed (iid) observations from an unknown probability distribution F on a space χ,

        $$x_i \overset{ind}{\sim} F, \quad i = 1, 2, ..., n. \tag{C.22}$$

        The space χ can be anything with no parametric form assumed for F
     b) A real-valued statistic $\hat{\theta}$ has been computed by applying some estimating algorithm $t(\cdot)$ to $\boldsymbol{x}$,

        $$\hat{\theta} = t(\boldsymbol{x}); \tag{C.23}$$

        and the objective is to assign a confidence interval to $\hat{\theta}$
     c) A non-parametric bootstrap sample $\boldsymbol{x}^*$,

        $$\boldsymbol{x}^* = (x_1^*, x_2^*, ..., x_n^*) \tag{C.24}$$

        is composed of n random draws, with replacement using MBB shown in Section 3.3.4.1, from the set of original data $\{x_1, x_2, ..., x_n\}$

d) Bootstrap replication is produced

$$\hat{\theta}^* = t(\boldsymbol{x}^*), \quad of\,\hat{\theta} \tag{C.25}$$

which produces B such replications

$$\hat{\theta}_i^* = t(\boldsymbol{x}_i^*), \quad i = 1, 2, ..., B, \tag{C.26}$$

e) The vector of bootstrap replications, say

$$\boldsymbol{t}^* = (\hat{\theta}_1^*, \hat{\theta}_2^*, ..., \hat{\theta}_B^*), \tag{C.27}$$

is used to provide estimate for $\hat{G}$ (the cdf of the bootstrap distribution of $\hat{\theta}$) and $\hat{z}_0$,

$$\hat{z}_0 = \Phi^{-1}\hat{G}(\hat{\theta}) \tag{C.28}$$

where Φ is the standard normal cdf

f) estimate the acceleration a using the jackknife differences d_i,

$$\hat{a} = \frac{1}{6}\frac{\sum_{i=1}^{n} d_i^3}{\left(\sum_{i=1}^{n} d_i^2\right)^{3/2}}, \quad d_i = \hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} \tag{C.29}$$

g) The standard method $level - \alpha$ endpoint

$$\hat{\theta}_{stand}(\alpha) = \hat{\theta} \pm z^{(\alpha)}\hat{\sigma} \tag{C.30}$$

where $\hat{\theta}$ is computed using BCA as

$$\hat{\theta}_{BCA}(\alpha) = G^{-1}\Phi\left(z_0 + \frac{z_0 + z^{(\alpha)}}{1 - a(z_0 + z^{(\alpha)})}\right) \tag{C.31}$$

and $\hat{\sigma}$ is computed using the empirical standard deviation of the $\hat{\theta}_i^*$ values,

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{B}\left(\hat{\theta}_i^* - \hat{\theta}_\cdot^*\right)^2}{(B-1)}}, \quad \hat{\theta}_\cdot^* = \frac{\sum \hat{\theta}_i^*}{B} \tag{C.32}$$

4. *Evaluation of discretization uncertainty*

The theoretical background is introduced in Section 3.3.5.

Condition: the statistical uncertainty calculated in step 3 should be small enough to continue with the evaluation of the discretization uncertainty.

- Having the the statistical value of $\boldsymbol{\Phi}_i$ called ϕ_i
- Estimate the typical grid size h_i using the conventional definition h_i^{avg} and using the weighting average definition h_i^{wavg} (Section 3.3.5.1) as

$$h_i^{avg} = \begin{cases} \sqrt[3]{\frac{Vol}{\#of cells}} & \text{in } 3-D \\ \sqrt[2]{\frac{Area}{\#of cells}} & \text{in } 2-D \\ \frac{Length}{\#of segments} & \text{in } 1-D \end{cases} \tag{C.33}$$

and,

$$h_i^{wavg} = \sum_{i=1}^{n_z} w_i h_{z_i} \tag{C.34}$$

where h_{z_i} is the cell size corresponding to zone z_i, n_z is the number of zones and w_i is defined as:

$$w_i = \frac{\frac{1}{h_{z_i}}}{\sum_{i=1}^{n_z} \frac{1}{h_{z_i}}} \tag{C.35}$$

ensuring that

$$\sum_{i=1}^{n_z} w_i = 1 \tag{C.36}$$

h_i can be any value between h_i^{avg} and h_i^{wavg}, it can be described by a normal distribution with mean ($\overline{h_i} = \mu = \frac{h_i^{avg}+h_i^{wavg}}{2}$) and standard deviation ($\sigma = \left|\frac{h_i^{avg}-h_i^{wavg}}{2}\right|$).
- The discretization uncertainty procedures are described in algorithm C.1

Algorithm C.1: LSRMONTE Algorithm

```
Input   :Vector $\phi_i$ $\{i = 1, 2, ..., n_g$ and $n_g$ number of grids$\}$
         Vector $\overline{h_i}$ called $h_i$ here for ease of notations
         Vector $h_i^{avg}$ and $h_i^{wavg}$
Output:Vector $\phi_i$
         Vector $\phi_0$ estimated exact solution
         Vector $U_\phi(\phi_i)$ uncertainty for variable $\phi_i$
Require: define error functions:
  $\delta_1 = \phi_i - \phi_0 = \alpha h_i$
  $\delta_2 = \phi_i - \phi_0 = \alpha h_i^2$
  $\delta_{12} = \phi_i - \phi_0 = \alpha_1 h_i + \alpha_2 h_i^2$
  $\delta_{p_j} = \phi_i - \phi_0 = \alpha h_i^{p_j}$
Require: define least-squares functions:
  $S_{RE}\left(\phi_0, \alpha, p\right) = \sqrt{\sum_{i=1}^{n_g}\left(\phi_i - \left(\phi_0 + \alpha h_i^p\right)\right)^2}$,
  $S_1\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g}\left(\phi_i - \left(\phi_0 + \alpha h_i\right)\right)^2}$,
  $S_2\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g}\left(\phi_i - \left(\phi_0 + \alpha h_i^2\right)\right)^2}$,
  $S_{12}\left(\phi_0, \alpha_1, \alpha_2\right) = \sqrt{\sum_{i=1}^{n_g}\left(\phi_i - \left(\phi_0 + \alpha_1 h_i^1 + \alpha_2 h_i^2\right)\right)^2}$,
  $S_{p_j}\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g}\left(\phi_i - \left(\phi_0 + \alpha h_i^{p_j}\right)\right)^2}$

Require: compute weights for least-squares functions:
  $w_i = \frac{\frac{1}{h_i}}{\sum_{i=1}^{n_g}\frac{1}{h_i}}$,
Ensure: $\sum_{i=1}^{n_g} w_i = 1$
Require: define least-squares functions:
  $S_{RE}^w\left(\phi_0, \alpha, p\right) = \sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i - \left(\phi_0 + \alpha h_i^p\right)\right)^2}$,
  $S_1^w\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i - \left(\phi_0 + \alpha h_i\right)\right)^2}$,
  $S_2^w\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i - \left(\phi_0 + \alpha h_i^2\right)\right)^2}$,
  $S_{12}^w\left(\phi_0, \alpha_1, \alpha_2\right) = \sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i - \left(\phi_0 + \alpha_1 h_i^1 + \alpha_2 h_i^2\right)\right)^2}$,
  $S_{p_j}^w\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g} w_i\left(\phi_i - \left(\phi_0 + \alpha h_i^{p_j}\right)\right)^2}$
```

```
**Error Estimation**
initialize $F_s \leftarrow 0.0$
solve $S_{RE}$ and $S_{RE}^{w}$ for $\left(\phi_0, \alpha, p\right)$ and get standard deviation $\sigma_{fit}$ for each fit
**if** $p_{S_{RE}} > 0$ and/or $p_{S_{RE}^{w}} > 0$ **then**
    Monotonic convergence
    **if** $p_{S_{RE}} > 0$ or $p_{S_{RE}^{w}} > 0$ **then**
        keep fit with $p > 0$ and its respective $\sigma_{fit}$
    **else**
        Keep fit with minimum $\sigma_{fit}$
    **end if**
    **if** $0.5 \leq p \leq 2.0$ **then**
        $\epsilon_{\phi} \leftarrow \alpha h_i^p$
        $\sigma \leftarrow \sigma_{fit}$
        $F_s \leftarrow 1.25$
    **else if** $p > 2.0$ **then**
        solve $S_1$, $S_2$, $S_1^w$, $S_2^w$
        **if** $\sigma_{S_1} < \{\sigma_{S_2} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i$, $\sigma \leftarrow \sigma_{S_1}$
        **else if** $\sigma_{S_2} < \{\sigma_{S_1} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i^2$, $\sigma \leftarrow \sigma_{S_2}$
        **else if** $\sigma_{S_1^w} < \{\sigma_{S_1} \wedge \sigma_{S_2} \wedge \sigma_{S_2^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i$, $\sigma \leftarrow \sigma_{S_1^w}$
        **else**
            $\epsilon_{\phi} \leftarrow \alpha h_i^2$, $\sigma \leftarrow \sigma_{S_2^w}$
        **end if**
    **else**
        solve $S_1$, $S_2$, $S_1^w$, $S_2^w$, $S_{12}$, $S_{12}^w$,
        **if** $\sigma_{S_1} < \{\sigma_{S_2} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w} \wedge \sigma_{S_{12}} \wedge \sigma_{S_{12}^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i$, $\sigma \leftarrow \sigma_{S_1}$
        **else if** $\sigma_{S_2} < \{\sigma_{S_1} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w} \wedge \sigma_{S_{12}} \wedge \sigma_{S_{12}^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i^2$, $\sigma \leftarrow \sigma_{S_2}$
        **else if** $\sigma_{S_1^w} < \{\sigma_{S_1} \wedge \sigma_{S_2} \wedge \sigma_{S_2^w} \wedge \sigma_{S_{12}} \wedge \sigma_{S_{12}^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i$, $\sigma \leftarrow \sigma_{S_1^w}$
        **else if** $\sigma_{S_2^w} < \{\sigma_{S_1} \wedge \sigma_{S_2} \wedge \sigma_{S_1^w} \wedge \sigma_{S_{12}} \wedge \sigma_{S_{12}^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha h_i^2$, $\sigma \leftarrow \sigma_{S_2^w}$
        **else if** $\sigma_{S_{12}} < \{\sigma_{S_1} \wedge \sigma_{S_2} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w} \wedge \sigma_{S_{12}^w}\}$ **then**
            $\epsilon_{\phi} \leftarrow \alpha_1 h_i + \alpha_2 h_i^2$, $\sigma \leftarrow \sigma_{S_{12}}$
        else if $\sigma_{S_{12}^w} < \{\sigma_{S_1} \wedge \sigma_{S_2} \wedge \sigma_{S_1^w} \wedge \sigma_{S_2^w} \wedge \sigma_{S_{12}}\}$ then
            $\epsilon_\phi \leftarrow \alpha_1 h_i + \alpha_2 h_i^2$, $\sigma \leftarrow \sigma_{S_{12}^w}$
        end if
    end if
else if $p_{S_{RE}} < 0 \wedge p_{S_{RE}^w} < 0$ then
    Anomalous Data Behavior
    $p$ is not defined
    $p$ is assumed $p \in [0.5, 2]$
    $p \leftarrow 0.5$
    for $p \leq 2$ do
        $p_j \leftarrow p$
        solve $S_{p_j}$ and $S_{p_j}^w$
        if $p_j == 0.5 \wedge \sigma_{S_{p_j}} < \sigma_{S_{p_j}^w}$ then
            $\epsilon_\phi \leftarrow \alpha h_i^{p_j}$, $\sigma \leftarrow \sigma_{S_{p_j}}$
        else if $p_j == 0.5 \wedge \sigma_{S_{p_j}^w} < \sigma_{S_{p_j}}$ then
            $\epsilon_\phi \leftarrow \alpha h_i^{p_j}$, $\sigma \leftarrow \sigma_{S_{p_j}^w}$
        else if $p_j ! = 0.5 \wedge \sigma < \sigma_{S_{p_j}}$ then
            $\epsilon_\phi \leftarrow \alpha h_i^{p_j}$, $\sigma \leftarrow \sigma_{S_{p_j}}$
        else if $p_j ! = 0.5 \wedge \sigma < \sigma_{S_{p_j}^w}$ then
            $\epsilon_\phi \leftarrow \alpha h_i^{p_j}$, $\sigma \leftarrow \sigma_{S_{p_j}^w}$
        end if
        $p \leftarrow p + 0.1$
    end for
    solve $S_{12}$, $S_{12}^w$
    if $\sigma_{S_{12}} < \sigma$ then
        $\epsilon_\phi \leftarrow \alpha_1 h_i + \alpha_2 h_i^2$, $\sigma \leftarrow \sigma_{S_{12}}$
    else if $\sigma_{S_{12}^w} < \sigma$ then
        $\epsilon_\phi \leftarrow \alpha_1 h_i + \alpha_2 h_i^2$, $\sigma \leftarrow \sigma_{S_{12}^w}$
    end if
end if
```

```
Uncertainty Estimation $U_\phi(\phi_i)$
Require: calculate data range parameter $\Delta\Phi$:

    $\Delta\Phi \leftarrow \frac{(\phi_i)_{max} - (\phi_i)_{min}}{n_g - 1}$    (C.37)

if $F_s == 0.0$ then
    calculate $F_{sh}$ as in Equation (3.37)
    $F_s \leftarrow 1.25 + F_{sh}$
end if
if $\sigma < \Delta\Phi$ then
    $U_\phi(\phi_i) \leftarrow F_s \epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}| \quad if\ \sigma < \Delta\Phi,$
else
    $U_\phi(\phi_i) \leftarrow F_s \frac{\sigma}{\Delta\phi}(\epsilon_\phi(\phi_i) + \sigma + |\phi_i - \phi_{fit}|) \quad if\ \sigma \geq \Delta\Phi.$
end if
```

Appendix D

Least-Squares Solution for Power Series Expansions

The solution of the least-squares for the power series expansions shown in Equation (3.22) to Equation (3.29) is presented and this solution is valid for weighted $(w_i = nw_i = 1.0)$ and non-weighted $(nw_i = n_g w_i)$ approaches (Eça et al. [12]). n_g represents the number of grids.

D.1 Single Term Expansion with Three Unknowns

This solution is valid for single term expansions with three unknowns $\left(\phi_0, \alpha, p\right)$ and id obtained from the minimum of the function:

$$S_{RE}^{w}\left(\phi_0, \alpha, p\right) = \sqrt{\sum_{i=1}^{n_g} w_i \left(\phi_i - \left(\phi_0 + \alpha h_i^p\right)\right)^2}, \tag{D.1}$$

which is solved by getting:

$$\frac{\partial S_{RE}^{w}}{\partial \phi_0} = 0, \quad \frac{\partial S_{RE}^{w}}{\partial \alpha} = 0, \quad \frac{\partial S_{RE}^{w}}{\partial p} = 0 \tag{D.2}$$

This leads to a system of non-linear equations as

$$\phi_0 = \sum_{i=1}^{n_g} w_i \phi_i - \alpha \sum_{i=1}^{n_g} w_i h_i^p, \tag{D.3}$$

$$\alpha = \frac{\sum_{i=1}^{n_g} w_i \phi_i h_i^p - \left(\sum_{i=1}^{n_g} w_i \phi_i\right)\left(\sum_{i=1}^{n_g} w_i h_i^p\right)}{\sum_{i=1}^{n_g} w_i \phi_i h_i^{2p} - \left(\sum_{i=1}^{n_g} w_i \phi_i\right)\left(\sum_{i=1}^{n_g} w_i h_i^p\right)} \tag{D.4}$$

$$\sum_{i=1}^{n_g} w_i \phi_i h_i^p \log(h_i) - \phi_0 \sum_{i=1}^{n_g} w_i h_i^p \log(h_i) - \alpha \sum_{i=1}^{n_g} w_i h_i^{2p} \log(h_i) = 0 \tag{D.5}$$

which has a standard deviation given by

$$\sigma_{S_{RE}^{w}} = \sqrt{\frac{\sum_{i=1}^{n_g} n w_i \left(\phi_i - \left(\phi_0 + \alpha h_i^p\right)\right)^2}{\left(n_g - 3\right)}} \tag{D.6}$$

D.2 Single Term Expansion with $p_j - order$ Term

The minimum of the function is computed to get $\phi_0 \; and \; \alpha$

$$S_{p_j}^{w}\left(\phi_0, \alpha\right) = \sqrt{\sum_{i=1}^{n_g} w_i \left(\phi_i - \left(\phi_0 + \alpha h_i^{p_j}\right)\right)^2} \tag{D.7}$$

which is obtained from

$$\frac{\partial S^w_{p_j}}{\partial \phi_0} = 0, \quad \frac{\partial S^w_{p_j}}{\partial \alpha} = 0 \tag{D.8}$$

This leads to a system of linear equations

$$\begin{bmatrix} 1 & \sum_{i=1}^{n_g} w_i h_i^{p_j} \\ \sum_{i=1}^{n_g} w_i h_i^{p_j} & \sum_{i=1}^{n_g} w_i h_i^{2p_j} \end{bmatrix} \cdot \begin{bmatrix} \phi_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_g} w_i \phi_i \\ \sum_{i=1}^{n_g} w_i \phi_i h_i^{p_j} \end{bmatrix} \tag{D.9}$$

that has a standard deviation given by

$$\sigma_{S^w_{p_j}} = \sqrt{\frac{\sum_{i=1}^{n_g} n w_i \left(\phi_i - \left(\phi_0 + \alpha h_i^{p_j}\right)\right)^2}{\left(n_g - 2\right)}} \tag{D.10}$$

D.3 Two-Term Expansion with First and Second-Order Terms

$\phi_0, \alpha_1\ and\ \alpha_2$ are obtained from the minimum of the function

$$S^w_{12}\left(\phi_0, \alpha_1, \alpha_2\right) = \sqrt{\sum_{i=1}^{n_g} w_i \left(\phi_i - \left(\phi_0 + \alpha_1 h_i^1 + \alpha_2 h_i^2\right)\right)^2}, \tag{D.11}$$

which is solved by

$$\frac{\partial S^w_{12}}{\partial \phi_0} = 0, \quad \frac{\partial S^w_{12}}{\partial \alpha_1} = 0, \quad \frac{\partial S^w_{12}}{\partial \alpha_2} = 0 \tag{D.12}$$

This leads to a system of linear equations

$$\begin{bmatrix} 1 & \sum_{i=1}^{n_g} w_i h_i & \sum_{i=1}^{n_g} w_i h_i^2 \\ \sum_{i=1}^{n_g} w_i h_i & \sum_{i=1}^{n_g} w_i h_i^2 & \sum_{i=1}^{n_g} w_i h_i^3 \\ \sum_{i=1}^{n_g} w_i h_i^2 & \sum_{i=1}^{n_g} w_i h_i^3 & \sum_{i=1}^{n_g} w_i h_i^4 \end{bmatrix} \cdot \begin{bmatrix} \phi_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_g} w_i \phi_i \\ \sum_{i=1}^{n_g} w_i \phi_i h_i \\ \sum_{i=1}^{n_g} w_i \phi_i h_i^2 \end{bmatrix} \tag{D.13}$$

that has a standard deviation given by

$$\sigma_{S_{p_j}^w} = \sqrt{\frac{\sum_{i=1}^{n_g} n\, w_i \left(\phi_i - \left(\phi_0 + \alpha_1 h_i + \alpha_2 h_i^2\right)\right)^2}{\left(n_g - 3\right)}} \tag{D.14}$$

APPENDIX E

Simulation Details: Flow Over a Flat Plate

The data for the estimation of the discretization uncertainty was obtained from the workshop on estimation of discretization errors based on grid refinement studies (*Workshop on Estimation of Discretization Errors Based on Grid Refinement Studies* [168]). A typical grid is shown in Figure E.1.

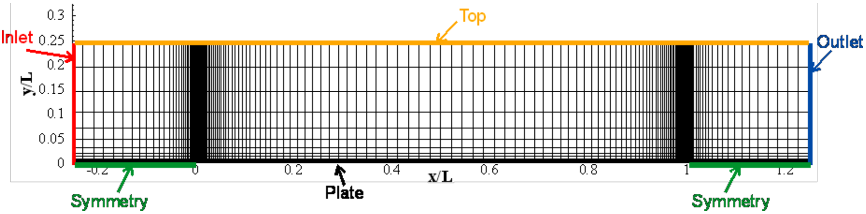


Figure E.1: Grid design for flow over a flat plate at $Re = 10^7$ and $Re = 10^9$

The domain is a rectangle of length $1.5L$ and width $0.25L$, where L is the plate length. As shown in Figure E.1, the inlet is located $0.25L$ from the leading edge of the plate and the outlet is located $0.25L$ from the trailing edge of the plate. The mesh densities are shown in Table E.1.

r_i	N_x	N_y		N_{cells}	
		$Re = 10^7$	$Re = 10^9$	$Re = 10^7$	$Re = 10^9$
1.000	1537	193	321	294912	491520
1.231	1249	157	261	194688	324480
1.455	1057	133	221	139392	232320
1.600	961	121	201	115200	192000
2.000	769	97	161	73728	122880
2.462	625	79	131	48672	81120
2.909	529	67	111	34848	58080
3.200	481	61	101	28800	48000
4.000	385	49	81	18432	30720
4.923	313	40	66	12168	20280
5.818	265	34	56	8712	14520
6.400	241	31	51	7200	12000
8.000	193	25	41	4608	7680

Table E.1: Mesh densities for flow over a flat plate

The boundary conditions for stream-wise velocity V_x, vertical velocity V_y, pressure P, turbulent kinetic energy k, turbulent frequency ω and undamped eddy-viscosity $\tilde{\nu}$[1] are defined as shown in Table E.2

[1] y_{c2} is the distance of the near-wall center to the plate

Boundary	Inlet	Outlet	Symmetry	Top	Plate
V_x	V_∞	$\frac{\partial V_x}{\partial x} = 0$	$\frac{\partial V_x}{\partial y} = 0$	$\frac{\partial V_x}{\partial y} = 0$	$V_x = 0$
V_y	$V_y = 0$	$\frac{\partial V_y}{\partial x} = 0$	$V_y = 0$	$\frac{\partial V_y}{\partial y} = 0$	$V_y = 0$
P	$P_{interior}$	$\frac{\partial P}{\partial x} = 0$	$\frac{\partial P}{\partial y} = 0$	$P = 0$	$\frac{\partial P}{\partial y} = 0$
k	$1.5x10^{-4} V_\infty^2$	$\frac{\partial k}{\partial x} = 0$	$\frac{\partial k}{\partial y} = 0$	$\frac{\partial k}{\partial y} = 0$	$k = 0$
ω	$10^9 \frac{k}{V_\infty L}$	$\frac{\partial \omega}{\partial x} = 0$	$\frac{\partial \omega}{\partial y} = 0$	$\frac{\partial \omega}{\partial y} = 0$	$\omega_{c2} = \frac{80 V_\infty L}{y_{c2}^2}$
$\tilde{\nu}$	$10^{-9} V_\infty L$	$\frac{\partial \tilde{\nu}}{\partial x} = 0$	$\frac{\partial \tilde{\nu}}{\partial y} = 0$	$\frac{\partial \tilde{\nu}}{\partial y} = 0$	$\tilde{\nu} = 0$

Table E.2: Boundary conditions for flow over a flat plate simulation

LIST OF FIGURES

List of Tables

BIBLIOGRAPHY

[1] https://www.archdaily.com/881090/the-10-different-ways-to-measure-a-skyscrapers-height/59df5756b22e38292f00095e-the-10-different-ways-to-measure-a-skyscrapers-height-image. Accessed: 2018-12-30.

[2] http://www.birdair.com/projects/khalifa-international-stadium. Accessed: 2018-12-30.

[3] B. Blocken. "50 years of Computational Wind Engineering: Past, present and future." In: *Journal of Wind Engineering and Industrial Aerodynamics* 129 (June 2014), pp. 69–102. DOI: 10.1016/j.jweia.2014.03.008.

[4] W. L. Oberkampf and T. G. Trucano. "Verification and Validation in Computational Fluid Dynamics 1." In: *Progress in Aerospace Sciences* 38.March 2002 (2002), pp. 210–270. DOI: 10.1016/S0376-0421(02)00005-2.

[5] F. M. S. Pereira. "Towards Predictive Scale-Resolving Simulations of Turbulent External Flows." PhD Thesis. University of Lisbona, 2018, p. 274.

[6] ASME V&V10 Committee. "Guide for verification and validation in computational solid mechanics." In: 2006.

[7] P. Gousseau, B. Blocken, and G. van Heijst. "Quality assessment of Large-Eddy Simulation of wind flow around a high-rise building: Validation and solution verification." In: *Computers & Fluids* 79 (June 2013), pp. 120–133. DOI: 10.1016/j.compfluid.2013.03.006.

[8] W. L. Oberkampf and C. J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511760396.

[9] R. J. Fisch. “Code Verification of Partitioned FSI Environments for Lightweight Structures.” PhD Thesis. Technischen Universität München, 2014. ISBN: 9783943683271.

[10] T. S. Phillips and C. J. Roy. “A New Extrapolation-Based Uncertainty Estimator for Computational Fluid Dynamics.” In: *Journal of Verification, Validation and Uncertainty Quantification* 1.4 (Jan. 2017), p. 041006. DOI: 10.1115/1.4035666.

[11] P. J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa, 1998. ISBN: 9780913478080.

[12] L. Eça and M. Hoekstra. “A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies.” In: *Journal of Computational Physics* 262 (2014), pp. 104–130. DOI: 10.1016/j.jcp.2014.01.006.

[13] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, and P. E. Raad. “Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications.” In: *Journal of Fluids Engineering* 130.7 (2008), p. 078001. DOI: 10.1115/1.2960953.

[14] J. Franke. “A review of verification and validation in relation to CWE.” In: 2010.

[15] S. Murakami. “Overview of turbulence models applied in CWE–1997.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 74-76 (Apr. 1998), pp. 1–24. DOI: 10.1016/S0167-6105(98)00004-X.

[16] A. Abodonya. “Validation of Wind-Structure Interaction Simulations for Pneumatic Inflatable Structures.” Master Thesis. Technical University of Munich, 2014, p. 164.

[17] F. S. Pereira, G. Vaz, L. Eça, and S. S. Girimaji. “Simulation of the flow around a circular cylinder at Re=3900 with Partially-Averaged Navier-Stokes equations.” In: *International Journal of Heat and Fluid Flow* 69.February (2018), pp. 234–246. DOI: 10.1016/j.ijheatfluidflow.2017.11.001.

[18] Guilhem Lacaze and Joseph C. Oefelein. “Development of Quality Assessment Techniques for Large Eddy Simulation of Propulsion and Power Systems in Complex Geometries.” In: *Sandia Report* March (2015).

[19] S. B. Pope. “A Perspective on Turbulence Modeling.” In: *Modeling Complex Turbulent Flows*. Ed. by M. D. Salas, J. N. Hefner, and L. Sakell. Dordrecht: Springer Netherlands, 1999, pp. 53–67. ISBN: 978-94-011-4724-8. DOI: `10.1007/978-94-011-4724-8_5`.

[20] L. Eca, M. Hoekstra, and G. Vaz. “Verification of Solutions in Unsteady Flows.” In: *ASME 2015 Verification and Validation Symposium*. Las Vegas, Nevada, 2015.

[21] *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. AIAA G.: American Institute of Aeronautics and Astronautics. American Institute of Aeronautics and Astronautics, 1998. ISBN: 9781563472855. DOI: `10.2514/4.472855.001`. eprint: `https://arc.aiaa.org/doi/pdf/10.2514/4.472855.001`.

[22] T. Stathopoulos. “The numerical wind tunnel for industrial aerodynamics: Real or virtual in the new millennium?” In: *Wind and Structures* 5 (Apr. 2002), pp. 193–208. DOI: `10.12989/was.2002.5.2_3_4.193`.

[23] L. Eça. “Solution Verification of Unsteady Flow Calculations.” In: vol. September. 2015, p. 2.

[24] L. Eça, G. Vaz, M. Hoekstra, S. Doebling, R. Singleton, G. Srinivasan, G. Weirs, T. S. Phillips, and C. J. Roy. *Assessment of Discretization Uncertainty Estimators Based on Grid Refinement Studies*. Minneapolis, 2018.

[25] NASA. “Standard for models and simulations.” In: *National Aeronautics and Space Administration (NASA-STD),* (2009).

[26] ASME V&V20 Committee. “Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer.” In: *The American Society of Mechanical Engineers (ASME)* (2009).

[27] ANS. *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry: ANSI/ANS-10.4*. American Nuclear Society, 1985.

[28] K. Salari and P. Knupp. *Code Verification by the Method of Manufactured Solutions*. Tech. rep. Sandia National Laboratories, 2000, p. 124.

[29] S. P. Veluri. "Code Verification and Numerical Accuracy Assessment for Finite Volume CFD Codes." PhD. Virginia Polytechnic Institute and State University, 2010, p. 126.

[30] R. G. Marchand. "The method of manufactured solutions for the verification of Computational Electromagnetic codes." PhD Thesis. Stellenbosch University, 2013, p. 115.

[31] P. Knupp and K. Salari. *Verification of Computer Codes in Computational Science and Engineering*. CHAPMAN & HALL/CRC, 2002.

[32] S. Steinberg and P. J. Roache. "Symbolic manipulation and computational fluid dynamics." In: *Journal of Computational Physics* 57.2 (1985), pp. 251–284. DOI: `10.1016/0021-9991(85)90045-2`.

[33] Š. Brglez. "Code verification for governing equations with arbitrary functions using adjusted method of manufactured solutions." In: *Engineering with Computers* 30.4 (2013), pp. 669–678. DOI: `10.1007/s00366-013-0315-0`.

[34] C. J. Roy, C. C. Nelson, T. M. Smith, and C. C. Ober. "Verification of Euler / Navier – Stokes Codes using the Method of Manufactured Solutions." In: *International Journal for Numerical Methods in Fluids* 620 (2004), pp. 599–620.

[35] L. Eça and M. Hoesktra. "Code Verification of Unsteady Flow Solvers with the Method of the Manufactured Solutions." In: *Proceedings of the seventeenth International Offshore and Polar Engineering Conference* June 2008 (2007), pp. 1–19.

[36] E. Luís, G. Vaz, and M. Hoekstra. "Code Verification of Refresco With a Statistically Periodic." In: *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering* June (2014), pp. 1–11. DOI: `10.1115/OMAE2014-23258`.

[37] S. Étienne, A. Garon, and D. Pelletier. "Some manufactured solutions for verification of fluid-structure interaction codes." In: *Computers and Structures* 106-107 (2012), pp. 56–67. DOI: 10.1016/j.compstruc.2012.04.006.

[38] K. R. Yu, S. Étienne, A. Hay, and D. Pelletier. "Code verification for unsteady 3-D fluid–solid interaction problems." In: *Theoretical and Computational Fluid Dynamics* 29.5-6 (2015), pp. 455–471. DOI: 10.1007/s00162-015-0367-4.

[39] A. Choudhary, C. J. Roy, E. A. Luke, and S. P. Veluri. "Code verification of boundary conditions for compressible and incompressible computational fluid dynamics codes." In: *Computers and Fluids* 126 (2016), pp. 153–169. DOI: 10.1016/j.compfluid.2015.12.003.

[40] P. J. Roache. "Code Verification by the Method of Manufactured Solutions." In: *Journal of Fluids Engineering* 124.1 (2002), p. 4. DOI: 10.1115/1.1436090. arXiv: 1602.06747.

[41] A. Choudhary, C. J. Roy, J. F. Dietiker, M. Shahnam, R. Garg, and J. Musser. "Code verification for multiphase flows using the method of manufactured solutions." In: *International Journal of Multiphase Flow* 80 (2016), pp. 150–163. DOI: 10.1016/j.ijmultiphaseflow.2015.12.006.

[42] S. P. Veluri, C. J. Roy, and E. A. Luke. "Comprehensive code verification techniques for finite volume CFD codes." In: *Computers and Fluids* 70 (2012), pp. 59–72. DOI: 10.1016/j.compfluid.2012.04.028.

[43] L. Eça and M. Hoekstra. "Verification of turbulence models with a manufactured solution." In: *European Conference on Computational Fluid Dynamics* (2006), pp. 1–14.

[44] S. Étienne, D. Tremblay, and D. Pelletier. "Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems." In: *36th AIAA Fluid Dynamics Conference and Exhibit* June (2006), pp. 1–11. DOI: 10.2514/6.2006-3218.

[45] L. Eça and M. Hoekstra. "Code Verification of unsteady flow solvers with method of manufactured solutions." In: *International Journal of Offshore and Polar Engineering* 18.2 (2008), pp. 120–126.

[46] L. Eça and M. Hoekstra. “Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions.” In: *Computers and Fluids* 38.8 (2009), pp. 1580–1591. DOI: 10.1016/j.compfluid.2009.01.003.

[47] C. Denis and S. Montan. “Numerical Verification of Industrial Numerical Codes.” In: *ESAIM: Proceedings and Surveys* 35.March (2012), pp. 107–113.

[48] J. DeBonis. “Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods.” In: *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* February (2013). DOI: 10.2514/6.2013-382.

[49] T. S. Phillips. “Extrapolation-based Discretization Error and Uncertainty Estimation in Computational Fluid Dynamics.” Master of Science Thesis. Virginia Polytechnic Institute and State University, 2012, p. 79.

[50] M. Ainsworth and J. T. Oden. “A posteriori error estimation in finite element analysis.” In: 37.96 (2011).

[51] T. S. Phillips and C. J. Roy. “Richardson Extrapolation-Based Discretization Uncertainty Estimation for Computational Fluid Dynamics.” In: *Journal of Fluids Engineering* 136.12 (2014), p. 121401. DOI: 10.1115/1.4027353.

[52] S. E. Gant. *Quality and Reliability Issues with Large-Eddy Simulation.* Tech. rep. 0. Derbyshire: Health and Safety Laboratory, 2009, p. 121.

[53] W. Rider, W. Witkowski, J. R. Kamm, and T. Wildey. “Robust verification analysis.” In: *Journal of Computational Physics* 307 (2016), pp. 146–163. DOI: 10.1016/j.jcp.2015.11.054.

[54] P. J. Roache. “Criticisms of the “Correction Factor” Verification Method.” In: *ASME. J. Fluids Eng* 125(4) (2003), pp. 732–733. DOI: 10.1115/1.1588693.

[55] L. F. Richardson. "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam." In: *Philosophical Transactions of the Royal Society of London* 210 (1911), pp. 307–357.

[56] C. J. Roy, A. Raju, and M. M. Hopkins. "Estimation of Discretization Errors Using the Method of Nearby Problems." In: *AIAA Journal* 45.6 (2007), pp. 1232–1243. DOI: 10.2514/1.24282.

[57] C. J. Roy, T. S. Phillips, M. J. Kurzen, A. J. Sinclair, and A. Raju. *Discretization Error Estimation and Exact Solution Generation using the Method of Nearby Problems*. Tech. rep. Albuquerque, New Mexico and Livermore, California: Sandia National Laboratories, 2011, pp. 1–80.

[58] T. Xing. "A general framework for verification and validation of large eddy simulations." In: *Journal of Hydrodynamics* 27.2 (2015), pp. 163–175. DOI: 10.1016/S1001-6058(15)60469-3.

[59] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, and K. Overholt. "Sixth edition fire dynamics simulator technical reference guide volume 1 : mathematical model." In: *NIST Special Publication 1018* 1 (2015). DOI: 10.6028/NIST.SP.1018-1.

[60] A. P. Smardz. "Validation of Fire Dynamics Simulator (FDS) for forced and natural convection flows." Master of Science Thesis. University of Ulster, 2006.

[61] Y. Addad, S. Benhamadouche, and D. Laurence. "The negatively buoyant wall-jet: LES results." In: *International Journal of Heat and Fluid Flow* 25.5 (2004), pp. 795–808. DOI: 10.1016/j.ijheatfluidflow.2004.05.008.

[62] J. S. Baggett, J. Jiménez, and A. G. Kravchenko. *Resolution requirements in large-eddy simulations of shear flows*. Tech. rep. Chapman 1979. 1997, pp. 51–66.

[63] J. Mahaffy, B. Chung, F. Dubois, F. Ducros, E. Graffard, M. Heitsch, and M. Henriksson. *Best Practice Guidelines for the use of CFD in Nuclear Reactor Safety Applications*. Tech. rep. Paris, France: Nuclear Energy Agency, 2007, p. 166. DOI: Nea/csni/r(2007)5.

[64] S. E. Gant. “Reliability issues of LES-related approaches in an industrial context.” In: *Flow, Turbulence and Combustion* 84.2 (2010), pp. 325–335. DOI: 10.1007/s10494-009-9237-8.

[65] S. B. Pope. *Turbulent Flows*. Vol. 1. Cambridge University Press, 2000, p. 771. ISBN: 9780511840531. DOI: 10.1088/0957-0233/12/11/705. arXiv: arXiv:1011.1669v3.

[66] I. Celik, M. Klein, M. Freitag, and J. Janicka. “Assessment measures for URANS/DES/LES: an overview with applications.” In: *Journal of Turbulence* 7 (2006), N48. DOI: 10.1080/14685240600794379.

[67] I. Celik, Z. Cehreli, and I. Yavuz. “Index of Resolution Quality for Large Eddy Simulations.” In: *ASME. J. Fluids Eng* 127(5) (2005), pp. 949–958. DOI: 10.1115/1.1990201.

[68] M. Klein. “An Attempt to Assess the Quality of Large Eddy Simulation in the Context of Implicit Filtering.” In: *Flow, Turbulence and Combustion* 75.1-4 (2005), pp. 131–147. DOI: 10.1007/s10494-005-8581-6.

[69] M. Hadžiabdić and K. Hanjalić. “Vortical structures and heat transfer in a round impinging jet.” In: *Journal of Fluid Mechanics* 596.2008 (2008), pp. 221–260. DOI: 10.1017/S002211200700955X.

[70] I. Celik, M. Klein, and J. Janicka. “Assessment Measures for Engineering LES Applications.” In: *Journal of Fluids Engineering* 131.3 (2009), p. 031102.

[71] B. J. Geurts and J. Fröhlich. “A framework for predicting accuracy limitations in large-eddy simulation.” In: *Physics of Fluids* 14.6 (2002). DOI: 10.1063/1.1480830.

[72] T. Brandt. “Study on numerical and modelling errors in LES using a priori and a posteriori testing.” PhD thesis. Helsinki University of Technology, 2007, p. 185. ISBN: 9789512286348.

[73] S. B. Pope. “Ten questions concerning the large-eddy simulation of turbulent flows.” In: *New Journal of Physics* 6 (2004). DOI: 10.1088/1367-2630/6/1/035.

[74] C. M. Garcia, P. R. Jackson, and M. H. Garcia. “Confidence intervals in the determination of turbulence parameters.” In: *Experiments in Fluids* 40.4 (2006), pp. 514–522. DOI: 10.1007/s00348-005-0091-8.

[75] S. E. Gant. *Large-Eddy Simulation of Gas Dispersion in a Room.* Tech. rep. 0. 2006.

[76] D. K. Lilly. “The Representation of Small-Scale Turbulence in Numerical Simulation Experiments.” In: *Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences* November (1967), pp. 195–210. DOI: 10.5065/D62R3PMM. arXiv: D62R3PMM [10.5065].

[77] A. Kempf, R. P. Lindstedt, and J. Janicka. “Large-eddy simulation of a bluff-body stabilized nonpremixed flame.” In: *Combustion and Flame* 144.1-2 (2006), pp. 170–189. DOI: 10.1016/j.combustflame.2005.07.006.

[78] W. L. Oberkampf and T. G. Trucano. *Validation Methodology in Computational Fluid Dynamics.* Tech. rep. June. Albuquerque, New Mexico: Sandia National Laboratories, 2000.

[79] J. Cadafalch, C. Pérez Segarra, R. Cònsul, and A. Oliva. “Verification of Finite Volume Computations on Steady State Fluid Flow and Heat Transfer.” In: *Journal of Fluids Engineering* 124 (2002), pp. 11–21.

[80] R. V. Wilson, F. Stern, H. W. Coleman, and E. G. Paterson. “Comprehensive Approach to Verification and Validation of CFD Simulations—Part 2: Application for Rans Simulation of a Cargo/Container Ship.” In: *Journal of Fluids Engineering* 123.4 (2001), p. 803. DOI: 10.1115/1.1412236.

[81] R. Wilson, J. Shao, and F. Stern. “Discussion: Criticisms of the “Correction Factor” Verification Method 1.” In: *Journal of Fluids Engineering* 126.4 (Sept. 2004), pp. 704–706. DOI: 10.1115/1.1780171. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/126/4/704/5903430/704_1.pdf.

[82] T. Xing and F. Stern. *Factors of Safety for Richardson Extrapolation.* Tech. rep. Iowa City: IIHR-Hydroscience and Engineering, 2009, p. 66. DOI: 10.1115/1.4001771.

[83] S. F. Xing Tao. "Factors of Safety for Richardson Extrapolation." In: *ASME. J. Fluids Eng* 132(6) (2010). DOI: 10.1115/1.4001771.

[84] L. Eça, M. Hoekstra, and P. J. Roache. "Verification of Calculations : an Overview of the 2nd Lisbon Workshop." In: *18th AIAA Computational Fluid Dynamics Conference.* Vol. June. 2005, pp. 1–12. ISBN: 1563478994. DOI: doi:10.2514/6.2005-4728.

[85] L. Eça and M. Hoekstra. "Discretization Uncertainty Estimation based on a Least Squares version of the Grid Convergence Index." In: *Proceedings of the Second Workshop on CFD Uncertainty Analysis, Lisbon.* 2006, pp. 1–27.

[86] T. Xing and F. Stern. "Comment on "A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies" (L. Eça and M. Hoekstra, Journal of Computational Physics 262 (2014) 104-130)." In: *Journal of Computational Physics* 301 (2015), pp. 484–486. DOI: 10.1016/j.jcp.2015.08.046.

[87] L. Eça and M. Hoekstra. "Reply to Comment on "A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies" (L. Eça and M. Hoekstra, Journal of Computational Physics 262 (2014) 104-130)." In: *Journal of Computational Physics* 301 (2015), pp. 487–488. DOI: 10.1016/j.jcp.2015.08.046.

[88] C. Negrato, T. Lloyd, T. van Terwisga, G. Vaz, and R. Bensow. "Numerical study of cavitation on a NACA0015 hydrofoil: solution verification." In: *VII International Conference on Computational Methods in Marine Engineering. MARINE2017.* Vol. May. 2017, pp. 1–12.

[89] F. S. Pereira, L. Eça, and G. Vaz. "Verification and Validation exercises for the flow around the KVLCC2 tanker at model and full-scale Reynolds numbers." In: *Ocean Engineering* 129 (2017), pp. 133–148. DOI: 10.1016/j.oceaneng.2016.11.005.

[90] B. Vink, J. Schot, G. Vaz, and S. Toxopeus. "A Verification and Validation Study of CFD Simulations for the Flow Around a Tug." In: *NuTTS-2017, Wageningen, the Netherlands*. Vol. October. 2017, pp. 1–6.

[91] I. C. Tolias et al. "Best Practice in Numerical Simulation and CFD Benchmarking. Results from the SUSANA Project." In: vol. September 2017. 2017.

[92] R. Dutta and T. Xing. "Quantitative Solution Verification of Large Eddy." In: *2nd Thermal and Fluid Engineering Conference*. TFEC-IWHT2017-18330. Las Vegas, Nevada, 2017, p. 4.

[93] R. Dutta and T. Xing. "Five-equation and robust three-equation method for solution verification of large eddy simulations." In: *Journal of Hydrodynamics* 30.1 (2018), pp. 23–33. DOI: 10.1007/s42241-018-0002-0.

[94] C. Roy and M. Hopkins. "Discretization Error Estimates Using Exact Solutions to Nearby Problems." In: *41st Aerospace Sciences Meeting and Exhibit*. 2003, pp. 1–12. ISBN: 978-1-62410-099-4. DOI: 10.2514/6.2003-629.

[95] C. J. Roy and W. L. Oberkampf. "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing." In: *Computer Methods in Applied Mechanics and Engineering* 200.25-28 (2011), pp. 2131–2144. DOI: 10.1016/j.cma.2011.03.016.

[96] P. Dadvand. "A framework for developing finite element codes for multi-disciplinary applications." Doctoral. Polytechnic University of Catalonia, 2007, p. 290.

[97] J. Cotela. "Applications of Turbulence Modeling in Civil Engineering." Doctoral. Polytechnic University of Catalonia, Jan. 2016, p. 203.

[98] D. Hertwig, G. Patnaik, and B. Leitl. "LES validation of urban flow, part I: flow statistics and frequency distributions." In: *Environmental Fluid Mechanics* 17.3 (2017), pp. 521–550. DOI: 10.1007/s10652-016-9507-7.

[99] D. Hertwig, G. Patnaik, and B. Leitl. “LES validation of urban flow, part II: eddy statistics and flow structures.” In: *Environmental Fluid Mechanics* 17.3 (2017), pp. 551–578. DOI: 10.1007/s10652-016-9504-x.

[100] M. Wacławczyk, Y.-F. Ma, J. M. Kopeć, and S. P. Malinowski. “Novel approaches to estimating turbulent kinetic energy dissipation rate from low and moderate resolution velocity fluctuation time series.” In: *Atmospheric Measurement Techniques Discussions* (2017), pp. 1–17. DOI: 10.5194/amt-2016-401.

[101] D. Nicolaides, D. R. Honnery, and J. Soria. “Autocorrelation Functions and the Determination of Integral Length with Reference to Experimental and Numerical Data.” In: *15th Australasian Fluid Mechanics Conference* 1.December (2004), pp. 1–4.

[102] L. Eca, G. Vaz, and M. Hoekstra. “Iterative Errors in Unsteady Flow Simulations : Are they Really Negligible ?” In: *NuTTS-2017, Wageningen, the Netherlands.* October. 2017, pp. 1–5.

[103] D. Haughton and J. Haughton. “Bootstrapping.” In: *Living Standards Analytics.* New York, NY: Springer New York, 2011, pp. 221–234. ISBN: 978-1-4614-0384-5. DOI: 10.1007/978-1-4614-0385-2_11.

[104] R. Theunissen, A. Di Sante, M. L. Riethmuller, and R. A. Van Den Braembussche. “Confidence estimation using dependent circular block bootstrapping: Application to the statistical analysis of PIV measurements.” In: *Experiments in Fluids* 44.4 (2008), pp. 591–596. DOI: 10.1007/s00348-007-0418-8.

[105] D. N. Politis and H. White. “Automatic Block-Length Selection for the Dependent Bootstrap.” In: *Econometric Reviews* 23.1 (2004), pp. 53–70. DOI: 10.1081/ETC-120028836.

[106] A. Patton, D. N. Politis, and H. White. “Correction to automatic block-length selection for the dependent bootstrap by D. Politis and H. White.” In: *Econometric Reviews* 28.4 (2009), pp. 372–375. DOI: 10.1080/07474930802459016.

[107] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap.* Mono. Stat. Appl. Probab. London: Chapman and Hall, 1993.

[108] G. Stephanie. *Bootstrap Sample: Definition, Example Kernel Description.* `http://http://www.statisticshowto.com/bootstrap-sample/`. Accessed: 2018-03-27. 2016.

[109] D. Karlis. *An introduction to Bootstrap Methods.* 17th Conference of Greek Statistical Society. Apr. 2004.

[110] C. Shalizi. *Lecture notes in The Bootstrap.* Feb. 2011.

[111] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. DOI: `10.1017/CBO9780511802843`.

[112] C. Evans. *scikits-bootstrap.* Version 1.0.0. Jan. 8, 2019.

[113] E. J. Allen. "Application of Richardson Extrapolation to Linear Functional Equations with Mildly Smooth Solutions." In: *International Journal of Computer Mathematics* 47.3-4 (1993), pp. 239–250. DOI: `10.1080/00207169308804181`.

[114] J. O. Rawlings, S. G. Pantula, and D. a. Dickey. *Applied Regression Analysis: A Research Tool.* 1998, p. 671. ISBN: 0387984542. DOI: `10.1007/b98890`.

[115] T. Phillips and C. J. Roy. *Global Deviation Uncertainty Estimator Applied to the 2017 Workshop on Estimation of Discretization Errors.* Las Vegas, Nevada, 2017.

[116] R. Codina and J. Blasco. "Stabilized finite element method for the transient Navier–Stokes equations based on a pressure gradient projection." In: *Computer Methods in Applied Mechanics and ...* 182 (2000), pp. 277–300. DOI: `10.1016/S0045-7825(99)00194-2`.

[117] R. Codina, J. Principe, and M. Ávila. "Finite element approximation of turbulent thermally coupled incompressible flows with numerical sub-grid scale modelling." In: *International Journal of Numerical Methods for Heat and Fluid Flow* 20 (2010), pp. 492–516. DOI: `10.1108/09615531011048213`.

[118] R. Codina. “Stabilized finite element approximation of transient incompressible flows using orthogonal subscales.” In: *Computer Methods in Applied Mechanics and Engineering* 191.39-40 (2002), pp. 4295–4321. DOI: 10.1016/S0045-7825(02)00337-7.

[119] R. Codina. “Pressure Stability in Fractional Step Finite Element Methods for Incompressible Flows.” In: *Journal of Computational Physics* 170.1 (2001), pp. 112–140. DOI: 10.1006/jcph.2001.6725.

[120] R. Codina. “A Stabilized Finite Element Method for Generalized Stationary Incompressible Flows.” In: *Computer Methods in Applied Mechanics and Engineering* 190.20-21 (2001), pp. 2681–2706. DOI: 10.1016/S0045-7825(00)00260-7.

[121] R. Codina and S. Badia. “On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems.” In: *Computer Methods in Applied Mechanics and Engineering* 195.23-24 (2006), pp. 2900–2918. DOI: 10.1016/j.cma.2004.06.048.

[122] I. Hanzlicek. “Staggered Algorithms for Fluid-Structure Interaction Staggered Algorithms for Fluid-Structure Interaction.” Master Thesis. Technical University of Munich, 2015, p. 71.

[123] T. J. Hughes. “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods.” In: *Computer Methods in Applied Mechanics and Engineering* 127.1-4 (1995), pp. 387–401. DOI: 10.1016/0045-7825(95)00844-9.

[124] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. “The variational multiscale method—a paradigm for computational mechanics.” In: *Computer Methods in Applied Mechanics and Engineering* 166.1-2 (1998), pp. 3–24. DOI: 10.1016/S0045-7825(98)00079-6.

[125] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. “An alpha modification of Newmark’s method.” In: *International Journal for Numerical Methods in Engineering* 15.10 (Oct. 1980), pp. 1562–1566. DOI: 10.1002/nme.1620151011.

[126] M. Arnold and O. Brüls. “Convergence of the generalized-α scheme for constrained mechanical systems.” In: *Multibody System Dynamics* 18.2 (2007), pp. 185–202. DOI: 10.1007/s11044-007-9084-0.

[127] D. Fauconnier, C. Bogey, and E. Dick. “On the performance of relaxation filtering for large-eddy simulation.” In: *Journal of Turbulence* 14.1 (2013), pp. 22–49. DOI: 10.1080/14685248.2012.740567.

[128] W. M. van Rees, A. Leonard, D. I. Pullin, and P. Koumoutsakos. “A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers.” In: *Journal of Computational Physics* 230.8 (2011), pp. 2794–2805. DOI: 10.1016/j.jcp.2010.11.031.

[129] M. E. Brachet, D. Meiron, S. Orszag, B. Nickel, R. Morf, and U. Frisch. “The Taylor-Green vortex and fully developed turbulence.” In: *Journal of Statistical Physics* 34.5-6 (1984), pp. 1049–1063. DOI: 10.1007/BF01009458.

[130] S. Bhatt and M. Student. “Solution of the Taylor-Green Vortex Problem Using Artificial Compressibility Method in Generalized Curvilinear Co-ordinates.” In: ().

[131] D. Fauconnier, C. De Langhe, and E. Dick. “Construction of explicit and implicit dynamic finite difference schemes and application to the large-eddy simulation of the Taylor-Green vortex.” In: *Journal of Computational Physics* 228.21 (2009), pp. 8053–8084. DOI: 10.1016/j.jcp.2009.07.028.

[132] A. Mastellone, F. Capuano, S. Benedetto, and L. Cutrone. “Problem C3 . 5 Direct Numerical Simulation of the Taylor-Green Vortex at Re = 1600.” In: *TILDA Tests* 0 (2015), pp. 1–5.

[133] J. R. Bull and A. Jameson. “Simulation of the Taylor–Green Vortex Using High-Order Flux Reconstruction Schemes.” In: *AIAA Journal* 53.9 (2015), pp. 2750–2761. DOI: 10.2514/1.J053766.

[134] C. Bajer. “Time integration methods - still questions.” In: *Theoretical Foundations of Civil Engineering* 10 (Jan. 2002), pp. 45–54.

[135] P. Beaudan and P. Moin. "Numerical experiments on the flow past a circular cylinder at sub-critical Reynolds number." In: *Stanford University* Report No. TF-62 (1994), pp. 1–262.

[136] A. G. Kravchenko and P. Moin. "Numerical studies of flow over a circular cylinder at ReD=3900." In: *Physics of Fluids* 12.2 (2000), pp. 403–417. DOI: 10.1063/1.870318.

[137] P. Parnaudeau, J. Carlier, D. Heitz, and E. Lamballais. "Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900." In: *Physics of Fluids* 20.8 (2008), pp. 1–14. DOI: 10.1063/1.2957018.

[138] J. Franke. "Large eddy simulation of the flow past a circular cylinder at ReD=3900." In: *Journal of Wind Engineering and Industrial Aerodynamics* 90 (2002), pp. 1191–1206. DOI: 10.1016/S0167-6105(02)00232-5.

[139] H. Ouvrard, B. Koobus, A. Dervieux, and M. V. Salvetti. "Classical and variational multiscale LES of the flow around a circular cylinder on unstructured grids." In: *Computers and Fluids* 39.7 (2010), pp. 1083–1094. DOI: 10.1016/j.compfluid.2010.01.017.

[140] S. Wornom, H. Ouvrard, M. V. Salvetti, B. Koobus, and A. Dervieux. "Variational multiscale large-eddy simulations of the flow past a circular cylinder: Reynolds number effects." In: *Computers and Fluids* 47.1 (2011), pp. 44–50. DOI: 10.1016/j.compfluid.2011.02.011.

[141] L. Bruno, D. Fransos, N. Coste, and A. Bosco. "3D flow around a rectangular cylinder: A computational study." In: *Journal of Wind Engineering and Industrial Aerodynamics* 98.6-7 (2010), pp. 263–276. DOI: 10.1016/j.jweia.2009.10.005.

[142] K. Lam and L. Zou. "Experimental study and large eddy simulation for the turbulent flow around four cylinders in an in-line square configuration." In: *International Journal of Heat and Fluid Flow* 30.2 (2009), pp. 276–285. DOI: 10.1016/j.ijheatfluidflow.2009.01.005.

[143] V. D'Alessandro, S. Montelpare, and R. Ricci. "Detached-Eddy simulations of the flow over a cylinder at Re = 3900 using OpenFOAM." In: *Computers & Fluids* June (2016). DOI: 10.1016/S0168-0102(03)00212-8.

[144] T. Kawata, Y. Naka, and S. Obi. "Simultaneous measurement of fluctuating velocity and pressure in the near wake of a circular cylinder." In: *Experiments in Fluids* 55.5 (2014). DOI: 10.1007/s00348-014-1731-7.

[145] F. Pereira, G. Vaz, and L. Eca. "An assessment of Scale-Resolving Simulation models for the flow around a circular cylinder." In: *Turbulence, Mass and Heat Transfer 8, Sarajevo, Bosnia* 2013 (2015), pp. 1–17. DOI: 10.13140/RG.2.1.2758.4489.

[146] C. Norberg. "Fluctuating lift on a circular cylinder: Review and new measurements." In: *Journal of Fluids and Structures* 17.1 (2003), pp. 57–96. DOI: 10.1016/S0889-9746(02)00099-3.

[147] P. W. Bearman and E. D. Obasaju. "An experimental study of pressure fluctuations on fixed and oscillating square-section cylinders." In: *Journal of Fluid Mechanics* 119 (June 1982), pp. 297–321. DOI: 10.1017/S0022112082001360.

[148] D. A. Lyn, S. Einav, W. Rodi, and J.-H. Park. "A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder." In: *Journal of Fluid Mechanics* 304 (1995), pp. 285–319. DOI: 10.1017/S0022112095004435.

[149] S. Nanda. "Flow past a square-prism : A numerical study." Master Thesis. Technical University of Delft, 2016.

[150] A. K. Saha, K. Muralidhar, and G. Biswas. "Experimental study of flow past a square cylinder at high Reynolds numbers." In: *Experiments in Fluids* 29.6 (Dec. 2000), pp. 553–563. DOI: 10.1007/s003480000123.

[151] T. Khademinezhad, P. Talebizadeh, and H. Rahimzadeh. "Numerical Study of Unsteady Flow around a Square Cylinder in Compare with Circular Cylinder." In: Dvm ().

[152] S. Dutta, P. K. Panigrahi, and K. Muralidhar. “Experimental Investigation of Flow Past a Square Cylinder at an Angle of Incidence.” In: *Journal of Engineering Mechanics* 134 (2008), pp. 788–803. DOI: `10.1061/(ASCE)0733-9399(2008)134:9(788)`.

[153] T. Arslan, G. K. El Khoury, B. Pettersen, and H. I. Andersson. “Simulations of flow around a three-dimensional square cylinder using LES and DNS.” In: *The Seventh International Colloquium on Bluff Body Aerodynamics and Applications* Ldv (2012), pp. 909–918.

[154] B. Gera, P. K. Sharma, and R. K. Singh. “CFD analysis of 2D unsteady flow around a square cylinder.” In: *Int. J. Appl. Eng. Res.* 1.3 (2010), pp. 602–610.

[155] M. Ricci, L. Patruno, I. Kalkman, S. de Miranda, and B. Blocken. “Towards LES as a design tool: Wind loads assessment on a high-rise building.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 180 (2018), pp. 1–18. DOI: `https://doi.org/10.1016/j.jweia.2018.07.009`.

[156] M. Péntek, A. Winterstein, M. Vogl, P. Kupás, K.-U. Bletzinger, and R. Wüchner. “A multiply-partitioned methodology for fully-coupled computational wind-structure interaction simulation considering the inclusion of arbitrary added mass dampers.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 177 (2018), pp. 117–135. DOI: `https://doi.org/10.1016/j.jweia.2018.03.010`.

[157] M. S. Thordal, J. C. Bennetsen, and H. H. H. Koss. “Review for practical application of CFD for the determination of wind load on high-rise buildings.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 186 (2019), pp. 155–168. DOI: `https://doi.org/10.1016/j.jweia.2018.12.019`.

[158] *Tokyo Polytechnic University Aerodynamic Database.* `http://wind.arch.t-kougei.ac.jp/system/eng/contents/code/tpu`. Accessed: 2018-10-30.

[159] AIJ. “Wind Loads.” In: *AIJ Recommendations for Loads on Buildings*. Ed. by Architectural Institute of Japan. 4th. Editi. Tokyo: Architectural Institute of Japan, 2005. Chap. CHAPTER 6, pp. C6–1 – C6–81. ISBN: 978-4-8189-5003-0.

[160] Y. Tamura, T. Ohkuma, H. Kawai, Y. Uematsu, and K. Kondo. “Revision of AIJ Recommendations for Wind Loads on Buildings.” In: May 2004, pp. 1–10. ISBN: 978-0-7844-0700-4. DOI: `10.1061/40700(2004)60`.

[161] J. Mann. “Wind field simulation.” In: *Probabilistic Engineering Mechanics* 13.4 (1998), pp. 269–282.

[162] H. Alsofi. “CWE Simulation of Dynamic Wind Loading on Wide-Span Membrane Structures.” Master Thesis. Technische Universität München, 2013.

[163] M. S. Andre. “Aeroelastic Modeling and Simulation for The Assessment of Wind Effects on a Parabolic Trough Solar Collector.” Doctoral. Technical University of Munich, May 2018, p. 171. ISBN: 978-3-943683-50-9.

[164] Y. Tominaga, A. Mochida, S. Murakami, and S. Sawaki. “Comparison of various revised k–ϵ models and LES applied to flow around a high-rise building model with 1:1:2 shape placed within the surface boundary layer.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 96.4 (Apr. 2008), pp. 389–411. DOI: `10.1016/j.jweia.2008.01.004`.

[165] D. Randerson. “Atmospheric science and power production.” In: (July 1984). DOI: `10.2172/6503687`.

[166] *Y plus wall distance estimation*. `https://www.cfd-online.com/Wiki/Y_plus_wall_distance_estimation`. Accessed: 2020-02-21.

[167] B. Efron and B. Narasimhan. “The automatic construction of bootstrap confidence intervals.” In: *Journal of Computational and Graphical Statistics* 0.ja (2020), pp. 1–32. DOI: `10.1080/10618600.2020.1714633`. eprint: `https://doi.org/10.1080/10618600.2020.1714633`.

[168] *Workshop on Estimation of Discretization Errors Based on Grid Refinement Studies.* `http://web.tecnico.ulisboa.pt/ist12278/Discretization/Workshop_discretization_2017.htm`. Accessed: 2018-07-30.

[169] H. R. Kunsch. “The Jackknife and the Bootstrap for General Stationary Observations.” In: *The Annals of Statistics* 17.3 (1989), pp. 1217–1241.

[170] C. Rogers, W. L. Oberkampf, J. S. Kaizer, and R. Crane. *Its Too Complex to Validate !* Minneapolis, 2018.

[171] I. B. Celik and D. R. Parsons. “Prediction of discretization error using the error transport equation.” In: *Journal of Computational Physics* 339.304 (2017), pp. 96–125. DOI: `10.1016/j.jcp.2017.02.058`.

[172] B. J. Geurts and J. Fröhlich. “A framework for predicting accuracy limitations in large-eddy simulation.” In: *Physics of Fluids* 14.6 (2002). DOI: `10.1063/1.1480830`.

[173] F. Jörg, H. Antti, S. Heinke, and C. Bertrand. *BEST PRACTICE GUIDELINE FOR THE CFD SIMULATION OF FLOWS IN THE URBAN ENVIRONMENT QUALITY ASSURANCE AND IMPROVEMENT OF.* May. 2007, pp. 1–52. ISBN: 3000183124.

[174] G. Bitsuamlak and E. Simiu. “CFD’s potential applications : a wind engineering perspective.” In: 2008 (2010).

[175] F. Stern, R. Wilson, and J. Shao. “Quantitative V & V of CFD simulations and certiÿcation of CFD codes.” In: *International Journal for Numerical methods in Fluids* 50.August 2005 (2006), pp. 1335–1355. DOI: `10.1002/fld.1090`.

[176] L. Eça and M. Hoekstra. “Code Verification of unsteady flow solvers with method of manufactured solutions.” In: *International Journal of Offshore and Polar Engineering* 18.2 (2008), pp. 120–126. DOI: `10.1115/1.1412235`.

[177] D. Pelletier. “Verification, validation, and uncertainty in computational fluids dynamics.” In: *Canadian Journal of Civil Engineering* 37.7 (2010), pp. 1003–1013. DOI: `10.1139/L10-032`.

[178] L. Eca, M. Hoekstra, P. J. Roache, and H. Coleman. “Code verification, solution verification and validation: an overview of the 3rd Lisbon workshop.” In: *19th AIAA Computational Fluid Dynamics* June (2009), AIAA 2009–3647. DOI: 10.2514/6.2009-3647.