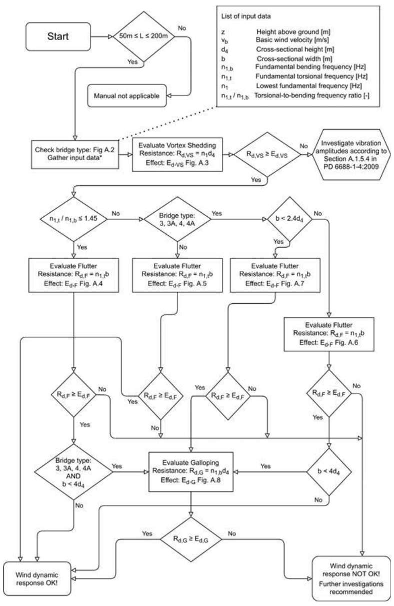
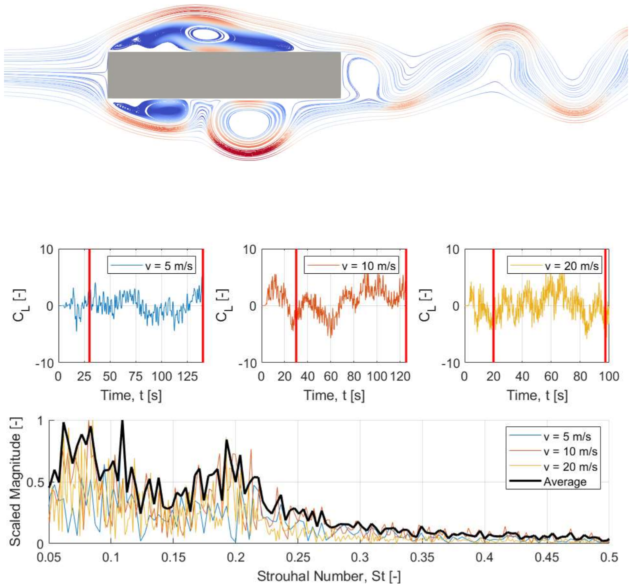
CHALMERS
UNIVERSITY OF TECHNOLOGY

Empirical Methods

Numerical Analysis





Wind Dynamic Assessment Methods for Medium-span Bridges

A comprehensive review of empirical and numerical approaches

Master's thesis in Master Programme Structural Engineering and Building Technology

LUKAS EHN
SVEN LUNDELL

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

MASTER'S THESIS 2021:52

Wind Dynamic Assessment Methods for Medium-span Bridges

A comprehensive review of empirical and numerical approaches

LUKAS EHN
SVEN LUNDELL

Department of Mechanics and Maritime Sciences
Division of Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Wind Dynamic Assessment Methods for Medium-span Bridges
A comprehensive review of empirical and numerical approaches
LUKAS EHN
SVEN LUNDELL

Supervisor: Christoffer Svedholm, Structural Engineer at ELU
Supervisor: Per Söderström, Structural Engineer at ELU
Examiner: Professor Thomas Abrahamsson, Department of Mechanics and Maritime Sciences

Master's Thesis 2021:52
Department of Mechanics and Maritime Sciences
Division of Dynamics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: To the left is a flowchart from the quick reference guide in Appendix A, presented in Chapter 4. To the right is a CFD simulation of the vortex shedding around a rectangle and graphs of the corresponding data from Chapter 7.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Wind Dynamic Assessment Methods for Medium-span Bridges
A comprehensive review of empirical and numerical approaches
LUKAS EHN
SVEN LUNDELL
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Generally, bridge engineers are unfamiliar with wind dynamics as it falls in-between the fields of structural engineering and fluid dynamics. Therefore, there is a need to summarize the field in a digestible manner. Procedures for wind dynamic assessments of medium-span bridges (e.g. bridges with longest spans of 50 to 200 metres) are investigated by studying both the current norm in Sweden, and an international alternative. A quick reference guide for wind dynamic assessment is developed, simplifying the procedure for bridge engineers. It offers significant time savings, especially in early stages of design, and it can prevent unexpected issues in later stages. However, to verify its reliability large scale testing on bridges is recommended. Additionally, possibilities of further analysis using computational fluid dynamics is investigated. Simulation data show some promising results and with further development, the methodology could provide better estimations than the norm. Conclusively, two useful tools for wind dynamic assessment of bridges are developed, and with further work, application in practice is possible for both methods.

Keywords: Aeroelastic instability, Bridge engineering, Computational fluid dynamics, Detached-eddy simulation, Eurocode, OpenFOAM, Strouhal number, Structural Engineering, Vortex induced vibrations, Vortex shedding.

Acknowledgements

We wish to acknowledge the effort of all those involved in the making of this thesis. Firstly, we would like to thank ELU Konsult AB, who proposed the subject of this thesis, and for providing guidance and support. We would like to especially extend our gratitude to our supervisors Christoffer Svedholm and Per Söderström, structural engineers at ELU, attending countless meetings and giving invaluable opinions and insights based on their experience as bridge engineers.

We would also like to thank our examiner Thomas Abrahamsson, professor in Structural Dynamics at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology, for his encouragement and willingness to aid us through the entire process.

Lastly, we would like to extend our gratitude to Håkan Nilsson, professor in Fluid Dynamics at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology, for answering questions regarding CFD which our supervisors and examiner were unfamiliar with.

Lukas Ehn

Sven Lundell

Gothenburg, June 2021

Contents

List of Figures

List of Tables

1
Introduction

The year is 1940 and the original Tacoma Narrows Bridge, crossing Puget Sound in Washington State, has just collapsed after violently twisting in the wind during an intense autumnal storm (Gaal, 2016). In the aftermath, the field of wind dynamics on bridges was born. Advances in material science and the ever-growing need of bridging larger obstacles means that bridges designed today are longer than ever before. For prestigious long suspension and cable stayed bridges wind dynamic assessments, in the form of wind tunnel testing, are code of practice. However, the wind dynamic phenomena are not only limited to the longest bridges. Therefore, the current Swedish norm for bridge design stipulate that the wind dynamic response for all bridges where the longest span exceeding 50 metres must be analysed.

Section 8.2 of Eurocode 1:4, published by the Swedish Institute for Standards [SIS] (2009), treats wind dynamic assessments of medium-span bridges, i.e., bridges with span lengths between 50 and 200 metres. Guidance on how to empirically assess the wind dynamic response for bridges within this range is given in the informative Annex E. However, the annex is ill-suited for bridges and the Swedish Transport Agency prohibits the use of several parts, without giving any further guidance (Transportstyrelsen, 2018). Wind tunnel testing is a reliable experimental alternative to employing the empirical methods in the Eurocode. For long suspension bridges the immense cost and time investment of conducting wind tunnel tests are justified, but for medium-span bridges they are not a viable alternative.

Generally, bridge engineers lack knowledge of the complexities of wind dynamics as the field lies in-between structural engineering and fluid dynamics. Therefore, the need to map and present available assessment methods in the field in a digestible manner is identified. The findings indicate that there are three main wind dynamic phenomena relevant for medium-spam bridges, presented in Figure 1.1.

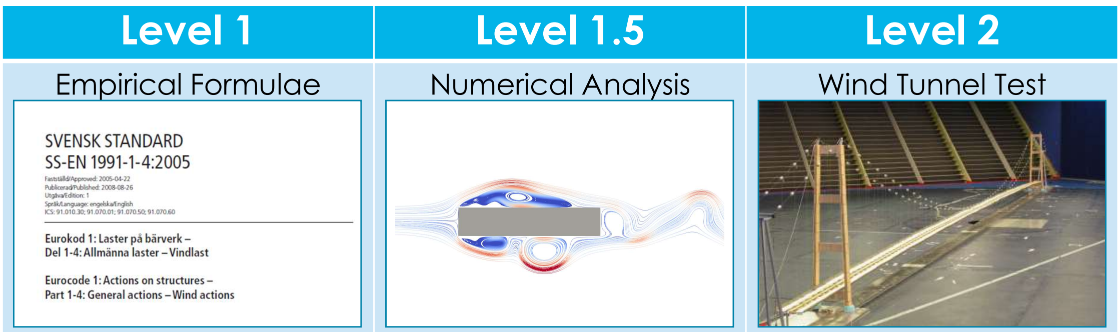


Figure 1.1: Wind dynamic assessment methods based on level of complexity.

Assessment can be performed at two levels of complexity, with an intermediate level separating the first and the second. An intricate review of the first level is conducted, comparing two empirical approaches. Based on the first level, an empirical quick reference guide for wind dynamic assessment, aimed at assisting bridge engineers in early design phases, is produced. Furthermore, the intermediate level is investigated, and a method using computational fluid dynamics (CFD) is proposed.

1.1 Aim and Objectives

The aim of this thesis is to investigate and summarize procedures for wind dynamic assessment of medium-span bridges. The goal is to present procedures of varying complexity, that bridge engineers unfamiliar to the field can easily apply in practice. To achieve this, the following objectives are identified:

- Identify and define relevant wind dynamic phenomena for medium-span bridges.
- Investigate the empirical procedure in the current Swedish norm and compare with international alternatives.
- Investigate numerical and experimental procedures.
- Compile procedures at different levels of complexity.

1.2 Methodology

In order to understand the dynamic structural response of bridges subjected to wind flows, a comprehensive literature study was conducted. The physics of relevant phenomena, associated terminology, and their design implications was compiled. Examples of bridges where wind dynamic phenomena caused significant issues or collapse were studied. Additionally, the current research front was reviewed, and findings of relevant articles and papers have been summarized.

Next, the current Swedish norm (SIS, 2009) and the national annexes were studied in detail. An international alternative was found in the British Annex to Eurocode (British Standards Institute [BSI], 2009). The methods in the norms were compared, and based on the findings a quick reference guide was developed, resting on the methods in the British Annex. It guides bridge engineers to design curves for relevant wind dynamic phenomena through the means of a flowchart, simplifying the assessment procedure.

While researching possible numerical alternatives for deeper analysis, it was found that vortex shedding could be simulated with CFD to more accurately determine the Strouhal number. To ensure that educated choices were made, a theoretical background of CFD was gathered and compiled. The open source CFD software OpenFOAM was used for two-dimensional simulations of virtual wind tunnels, analysing the variation of the lift coefficient over time to extract the Strouhal number. The procedure was verified against reference cases, both from literature and tabulated data in the norms.

1.3 Limitations

The norm stipulate that the wind dynamic response of medium-span bridges, i.e., bridges with longest spans of 50 to 200 metres, must be investigated. Therefore, the main focus of this thesis is methods for wind dynamic assessments of medium-span bridges. Furthermore, the norm is not directly applicable to arch, suspension, cable-stayed and movable bridges, as well as bridges with strong curves. Hence, these types of bridges are not considered in this thesis. Additionally, the wind dynamic response of individual members of the bridges are not considered.

The purpose of the numerical simulation method is to be used by bridge engineers in practice, and their computational resources are often limited. Hence, the method is limited to 2D in order to reduce the computational cost otherwise associated to 3D CFD simulations. Furthermore, the numerical simulations are limited to the phenomena vortex shedding, as it was considered most relevant for medium-span bridges.

1.4 Outline

The thesis is divided into two main parts to guide the reader through the contents. Where the first part treats wind dynamic assessment of bridges on a comprehensive level, the second part dives deeper into the analysis of one specific phenomena, highlighting possibilities of applying new techniques to the field. The parts are tied together in a mutual discussion and conclusion.

Part I treats the theory and norms of wind dynamics of bridges. In **Chapter 2**, the aerodynamic phenomena are defined and explained. Also, the respective research fronts are summarized. In **Chapter 3**, the methods for wind dynamic assessment in the current Swedish norm, Eurocode 1:4, is presented. Furthermore, an alternative method from the British Annex to Eurocode is introduced. In **Chapter 4**, a quick reference guide for wind dynamic assessment of bridges, based on the method in the British Annex, is presented. The entirety of the quick reference guide, and an accompanying background document, are appended in **Appendices A** and **B**.

Part II treats numerical analysis of vortex shedding. In **Chapter 5**, an introduction to the field of computational fluid dynamics relevant for bridge engineers is given. In **Chapter 6**, an approach to study vortex shedding in bridges is presented. In **Chapter 7**, simulation results are compared against data from literature, presented in detail in **Appendix C**, in order to verify the approach. In **Appendix D**, MATLAB code for data processing is presented. In **Appendix E**, directory structure and files for simulations in OpenFOAM are presented.

The **Discussion and Conclusion** is mutual for **Parts I** and **II**. In **Chapter 8**, the results and the implications of both parts are discussed in detail. In **Chapter 9**, the findings of the thesis are summarized in conjunction with possibilities for future research.

Part I

Wind Dynamics of Bridges; Theory and Norms

2
Wind Dynamic Phenomena

Wind dynamics, in other industries known as aerodynamics, is the study of air flows that interact with solid bodies (Simiu & Yeo, 2019). The flow around a body will produce lift and drag forces acting on its center of lift. Lift forces are generated by a pressure difference due to the body acting as a divider, where a low pressure zone is developed on one side the body and a high pressure zone on the other. The force acts towards the low pressure zone and its magnitude is described by a lift coefficient. Drag forces arise in the opposite direction of the flow due to its volume disturbing the flow field. The magnitude of the drag force depends on how streamlined the body is, described by a drag coefficient, where a lower value corresponds to a more streamlined body. Bodies with high drag coefficients, i.e., non-streamlined, are called bluff. In structural engineering, bluff body aerodynamics is of interest as few structures are designed to be aerodynamically efficient. Although large differences to the aviation industry exist, more or less the same phenomena affect the wings of airplanes and bridges. This is visualized in Figure 2.1, where an airfoil is compared to a bridge deck.

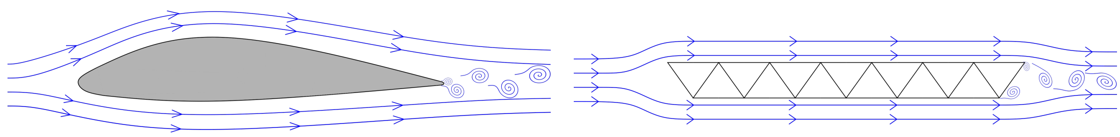


Figure 2.1: Bluff body aerodynamics analogy of an airfoil and a bridge deck.

One difference between airfoils and bridges is that the former travels through the air while the latter is stationary. However, physically, it is the relative wind velocity that is of importance when studying the response. Some phenomena are more intuitive and discernible when studying airfoils rather than bridges. Therefore, airfoils can be used to illustrate and explain some of the phenomena of relevance in bridge engineering.

The field of wind dynamics of bridges is complex with extensive and sometimes confusing and ambiguous terminology. The motion patterns of some phenomena are similar and it can be difficult to distinguish them. In order to establish the terminology of this thesis, a comprehensible graphical overview of the dynamic effects on bridges due to wind flow is presented in Figure 2.2.

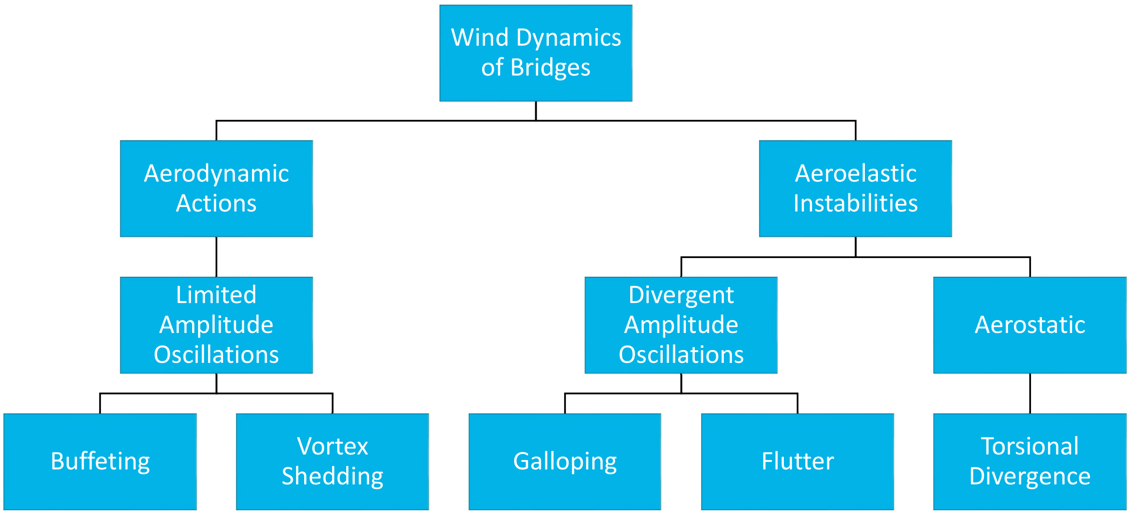


Figure 2.2: Overview of wind dynamics of bridges.

Wind dynamics of bridges can be divided into aerodynamic actions and aeroelastic instabilities. The aerodynamic actions are limited amplitude oscillations, and can be further subdivided into buffeting and vortex shedding, leading to vortex induced vibrations (VIVs). While limited amplitude oscillations may cause noticeable displacements, they are generally not a direct cause of structural failure. Instead, they may initiate divergent amplitude oscillations. The instabilities and VIVs are of aeroelastic nature, meaning that the influence of the aero-part, i.e., wind flow, is coupled with the elastic motions of the structure. The term aeroelastic instability includes both oscillatory phenomena of diverging amplitude, galloping and flutter, and the static phenomenon torsional divergence. These three phenomena may, by their divergent nature, cause failure if not interrupted. In Figure 2.3, the motion patterns of galloping, torsional flutter and classical flutter are illustrated.

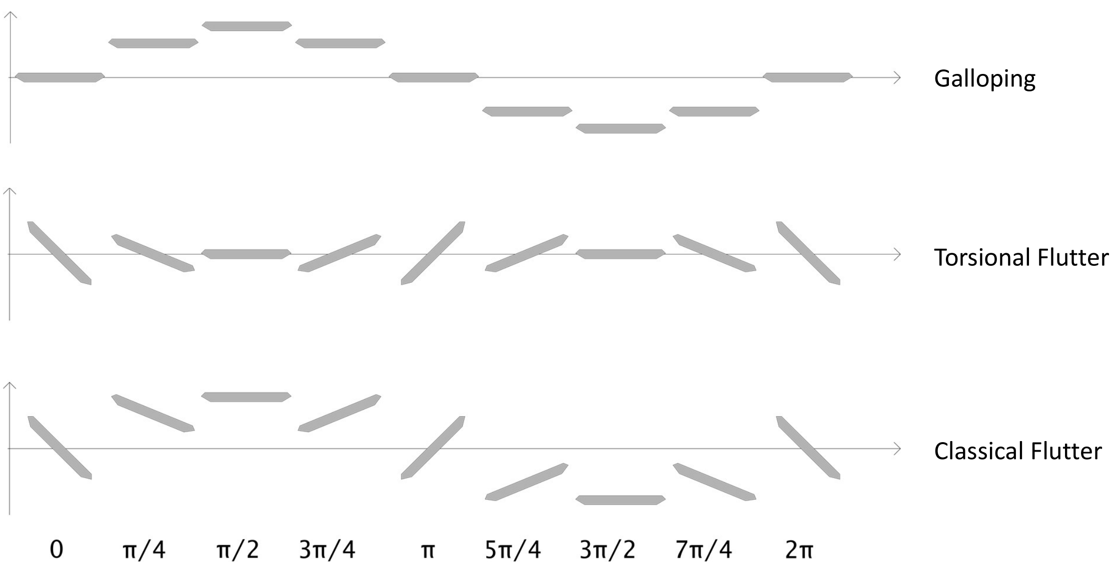


Figure 2.3: One oscillation period for the motion patterns related to each oscillatory aeroelastic instability of a bridge deck cross-section.

Galloping corresponds to pure longitudinal bending motion which appears vertical in the section view of Figure 2.3. Torsional flutter, also known as stall flutter, is pure torsion and classical flutter is coupled bending and torsion (De Miranda, 2016). As per the terminology of Eurocode, both torsional flutter and classical flutter are denoted as flutter.

A schematic diagram depicting the relation of the oscillatory phenomena, with vibration amplitude as a function of wind velocity, is illustrated in Figure 2.4. It shows the ranges of wind velocities where the respective phenomena are expected to dominate the vibration amplitude.

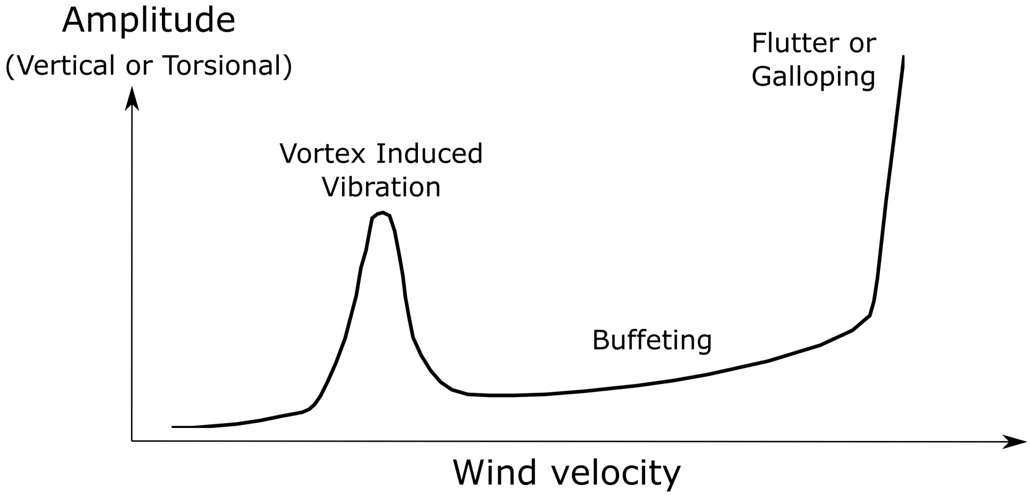


Figure 2.4: Wind velocity ranges at which the oscillatory phenomena are dominant with vertical and or torsional response. Reproduced from illustration by Prof. Fujino from the University of Tokyo.

It is apparent that the ranges of when the phenomena are relevant overlaps. As seen in Figure 2.4, the peak for VIVs occur at low wind velocities. However, as it is limited in amplitude it can in some cases be allowed to occur. For the instabilities flutter and galloping, the amplitude diverges and therefore is never allowed. Note that the vibration amplitude can either be vertical or torsional. This is because, similarly to natural frequencies, aeroelastic phenomena are associated with either vertical or torsional movement. A further note is that torsional divergence is of static nature. Hence, it is not visible in the figure as there is no associated amplitude. In the following sections, the aerodynamic actions and aeroelastic instabilities will be defined and visualized, and their effect on bridges described. Also, the respective research fronts will be summarized.

2.1 Aerodynamic Actions

The aerodynamic actions lead to limited amplitude oscillations in the form of either buffeting or VIVs. Both phenomena are external dynamic actions, acting on a body or structure and are of turbulent nature. However, while buffeting stems from the natural turbulence of the wind, VIVs are caused by turbulent vortices in the wake of a bluff body.

2.1.1 Buffeting

Structures, similarly to bushes and trees, sway in the wind, albeit with smaller motions. The swaying motion stems from fluctuations in the air pressure acting on the object due to wind gusts, generated by atmospheric turbulence (Larose & Larsen, 2015). Resulting structural vibrations are called buffeting and, as previously mentioned, it is not an aeroelastic phenomena. This means that the structural response does not influence the interaction with the wind.

In bridge engineering, most analyses of structural response due to wind assume constant wind velocity, considering the gusty and turbulent nature of wind through partial factors. The statistical modelling approach of buffeting was developed in the 1960s by professor Alan G. Davenport, a renowned contributor to the field of wind engineering, and it is the basis of the current theory of buffeting response (Cheynet et al., 2016). It is based on relating the gust velocity to the mean wind velocity and it divides the structural response into resonant and background parts, where the resonant response accounts for 70-80 % of the response according to Larose & Larsen (2015). Generally, buffeting response is of interest for slender and flexible bridges, such as suspension bridges. The turbulence of the wind flow depends significantly on the topography in the vicinity of the bridge, i.e., terrain or presence of other structures. Buffeting due to turbulence in the wake of an adjacent bluff body is called wake buffeting.

Although the buffeting theory was introduced more than half a century ago, few full scale studies to verify the theory have been conducted. According to Cheynet et al. (2016), the conducted studies have all been flawed as the statistical significance of them were insufficient. Specifically, the duration of the tests were too short. One of the studies, conducted by Katsuchi et al. (2002), during six hours of typhoon conditions of the Akashi Kaikyō bridge in Japan partially verified the theory. However, Cheynet et al. (2016) deemed it irrelevant for European bridge engineering in part due to the short time duration, but also due to the typhoon conditions which are limited to the western parts of the pacific ocean. In their own research, Cheynet et al. (2016) measured the buffeting response and wind conditions of the Lysefjord Bridge in Norway. During two tests, each with a duration of 24 hours, two main wind directions, approximately 180 degrees apart, were observed. Good correlation between model and measurements were found for one wind direction but less so for the other, in part due to differing turbulence properties between the two directions. Hence, a case-by-case approach was recommended as buffeting is so dependent on local conditions, such as the terrain.

2.1.2 Vortex Shedding

Vortex shedding is a phenomena where vortices form in the wake of a bluff body subjected to wind flow (Simiu & Yeo, 2019). As seen in Figure 2.5, the vortices forming in the wake of the bridge deck have alternating rotational direction, and therefore produce fluctuating upwards and downwards lift force on the body. The oscillatory motion of the body generated by the lift forces is what is known as VIVs.

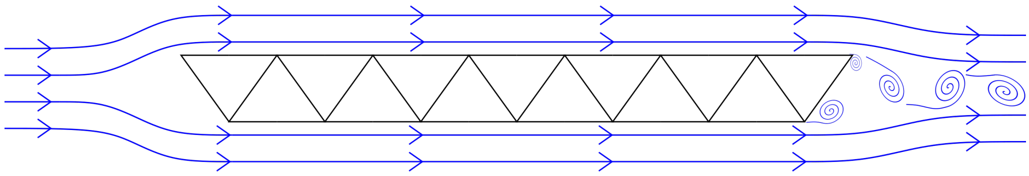
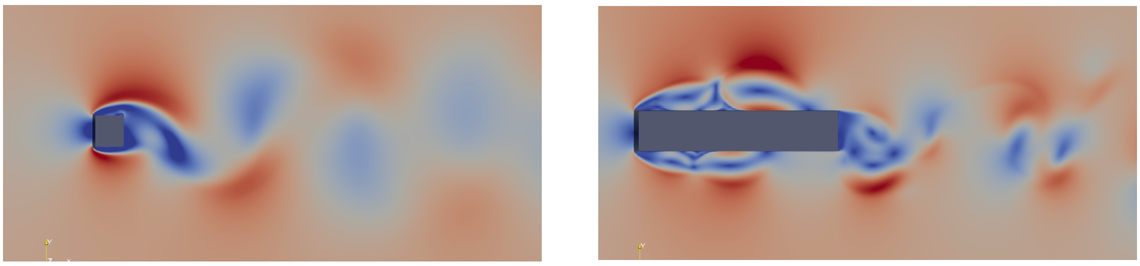


Figure 2.5: Illustration of vortex shedding around a bridge deck.

Note that vortices originate from sharp edges, where a low pressure area is formed. According to Bruno & Khris (2003), there are two types of vortex shedding; separated and reattached. The former is typical for compact bodies, such as cylinders and squares, producing orderly von Kármán vortex streets in the wake of the body. The latter is typical for wider bodies, such as most bridge cross-sections, producing vortices that detach and then reattach to the body further downstream. When these vortices interact with those forming behind the body, a more complex and chaotic wake pattern with large variations in vortex size and frequency is formed. In Figure 2.6, the two types of vortex shedding are visualized with separated type in (a) and reattached type in (b).



(a) **(b)**

Figure 2.6: Velocity flow fields for visualization of separated type vortex shedding (a) and reattached type vortex shedding (b). Blue and red colour indicate low and high velocity, respectively.

The vortex shedding frequency, n_{vs}, i.e., the frequency of fluctuation in lift force direction, is determined by Equation 2.1

$$n_{vs} = \frac{USt}{D} \tag{2.1}$$

where U is the wind velocity, St is the Strouhal number and D is a characteristic body dimension (Simiu & Yeo, 2019). For bridges D is the cross-sectional height. The Strouhal number is a dimensionless number depending on Reynolds number and body geometry. Vortex shedding can lead to large amplitude oscillations when the vortex shedding frequency is in proximity to a natural frequency of the bluff body, where the VIVs are reinforced (Bourguet et al., 2011). This is called lock-in, which is visualized in Figure 2.7 for the first two natural frequencies, n_1 and n_2. As VIVs can be both vertical and torsional, the degree of freedom (DOF) with the lowest natural frequency is governing, meaning that n_1 and n_2 can be either vertical or torsional.

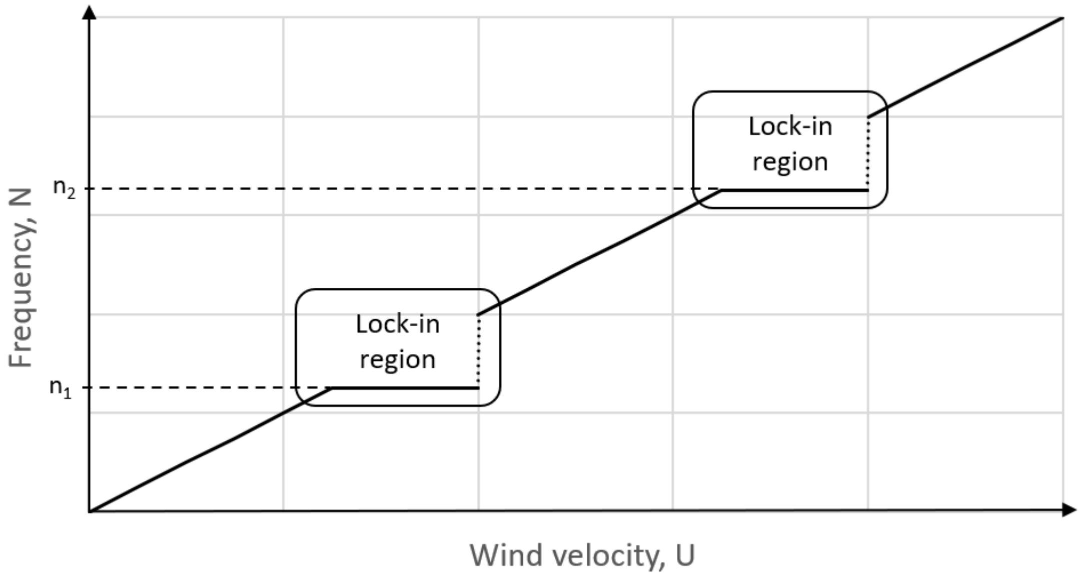


Figure 2.7: Graph of vortex shedding frequency for an arbitrary bluff body.

As vortices form in the wake of all bluff bodies, VIVs can be a significant issue in bridge design, especially for bridge decks. However, the conditions for when lock-in occurs varies dependent on the characteristics of the cross-section and the natural frequencies of the bridge. In design, the aim is to ensure that lock-in does not occur for the wind velocities the bridge is subjected to.

The amplitude of displacements caused by VIVs at lock-in may be relatively large, but in general it does not put the structural integrity at risk (Gimsing & Larsen, 1992). It may, however, give rise to physical discomfort for users and, in long-term, wear in bearings and joints. While it is not alone capable of causing structural failure, it may initiate more damaging phenomena, in the form of aeroelastic instability, which was the case in the well known Tacoma Narrows Bridge (Gaal, 2016). Flutter and its role in the catastrophe will be treated in a subsequent chapter.

Three other examples of large amplitude oscillations due to VIVs are the Rio-Niterói Bridge in Brazil in episodes between 1980 to 1998, the Trans-Tokyo Bay Bridge in Japan in 1995 and the Volgograd Bridge in Russia in 2010. Notably, none of these three lead to structural failure and, due to precise and effective countermeasures, they are still operational to this day (Corriols, 2015). Figure 2.8 is a photograph of the oscillations, with amplitudes of about 70 cm, of the Volgograd bridge in 2010.

Figure 2.8: Oscillations of the Volgograd Bridge in May 2010. Licensed under CC-BY.

A common denominator of the three bridges are their steel box girder cross sections with low deck-width-to-span-length ratios, i.e. high slenderness. This gave the structures low natural frequencies, for example in the range of 0.3-0.6 Hz for the Rio-Niterói bridge (Battista & Pfeil, 2000). The vortex shedding frequency matched this range, inducing lock-in effects for wind velocities in the region of 15-17 m/s, a condition that was met several times between 1980 and 1998. A suggested mitigation measure was to alter the cross-sectional shape by installing aerodynamic appendages in order to alter the wake frequency. However, this proved ineffective, and for all three bridges the ultimately decided upon countermeasure was installing mass dampers inside the box-sections (Corriols, 2015).

Wind tunnel testing is an effective tool to evaluate bridge response due to VIVs, but there are some limitations. Firstly, wind tunnel testing is an expensive and extensive measurement technique requiring both carefully crafted models and advanced equipment. Secondly, the scaled models generate some inaccuracies as the flow of air also must be scaled down (Wu et al., 2019). The Reynolds number is an important parameter that describes how turbulent the wind flow is, where higher number corresponds to more turbulent flow. Due to the scaling, the wind flow is more turbulent in reality than what is reproduced in wind tunnel testing. The Reynolds number is often assumed to not influence the air flow around a bridge deck, but according to Larsen & Schewe (1998) it is in some cases not negligible. Specifically, bluff bodies with sharp edges, such as bridge box girders, violate the assumption. In these cases, the lift and drag coefficients may be inaccurately estimated.

The alternative to wind tunnel testing is to simulate a wind tunnel using Computational Fluid Dynamics (CFD). CFD is proving to be a very useful tool for bridge engineering applications without the financial drawbacks of wind tunnel tests. While the use is widespread in other industries such as the aviation and automotive industries, it has yet been widely implemented in bridge engineering. However, its application is of ever increasing interest. On the other hand, there are still some limitations as the simulations are computationally expensive and turbulent flows are challenging to model (Wu et al., 2019).

Due to the shortcomings of wind tunnel testing and CFD simulations, Wu et al. (2019) sought to establish a semi-empirical model for VIVs of bridge decks. This was accomplished by using sinusoidal input describing functions (SIDF) which approximates nonlinear systems as quasi-linear by neglecting higher-order components. For example, as the lock-in effect is the most influential factor for VIVs, components of the system not relevant for lock-in can be neglected. Based on comparisons with experimental results from case studies, conclusions were drawn that the SIDF approach was sufficiently accurate at predicting vertical VIVs. However, modelling of torsional VIVs was not satisfactory and requires additional investigation.

2.2 Aeroelastic Instabilities

The aeroelastic instability phenomena are, once certain conditions are met, divergent and sustained by internal self-excited forces. This entails that the elastic structural response increases the response due to the phenomena, leading to divergence. In bridge engineering, the three relevant aeroelastic instability phenomena are galloping, flutter, and torsional divergence. They can, in practice, be distinguished by their respective motion pattern or amplitude and frequency.

2.2.1 Galloping

Galloping describes low frequency oscillations with large amplitude, in the order of at least a cross-sectional dimension of the body (Simiu & Yeo, 2019). The motion pattern for galloping of a bridge deck is illustrated in Figure 2.9 where the deck moves upwards and downwards in a bouncing motion perpendicular to the wind direction.

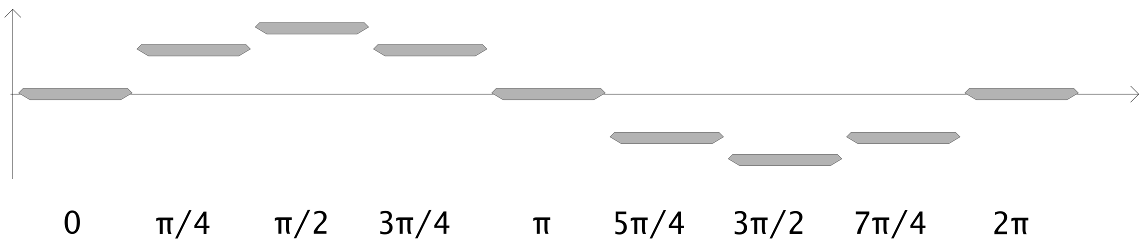


Figure 2.9: One oscillation period of galloping motion of a bridge deck cross-section.

Wake galloping, also known as interference galloping, is a separate aeroelastic phenomenon where oscillations are induced in a cylindrical body by turbulence in the wake of an adjacent but not connected cylindrical body (Dielen & Ruscheweyh, 1995). It is of interest for closely grouped chimneys and power-cables, but it is rarely relevant in bridge engineering. However, assessing the susceptibility for ordinary galloping of a bridge deck is an important part of a wind dynamic assessment.

The Den Hartog stability criterion can be used to assess the galloping stability of a bridge deck and is presented in Equation 2.2

$$\left[\frac{dC_L(\alpha)}{d\alpha} + C_D(\alpha)\right]_{\alpha=0} < 0 \tag{2.2}$$

where a body is unstable if the expression on the left is smaller than zero (Simiu & Yeo, 2019). C_L and C_D are the lift and drag coefficients, respectively, and α is the angle of attack of the wind flow. In empirical calculations, the factor of galloping instability, a_G, is often used as a substituted to the stability criterion. It closely related to Equation 2.2, and it is defined as

$$a_G = -\left[\frac{dC_L(\alpha)}{d\alpha} + C_D(\alpha)\right]_{\alpha=0} \tag{2.3}$$

Similarly to the parameters of importance for torsional divergence, the parameters of galloping can be determined while the body is at rest. For a bridge deck, an angle of attack of zero degrees corresponds to wind parallel to the deck. As seen in Figure 2.10, when the angle of attack exceeds a certain threshold the slope of the lift coefficient becomes negative.

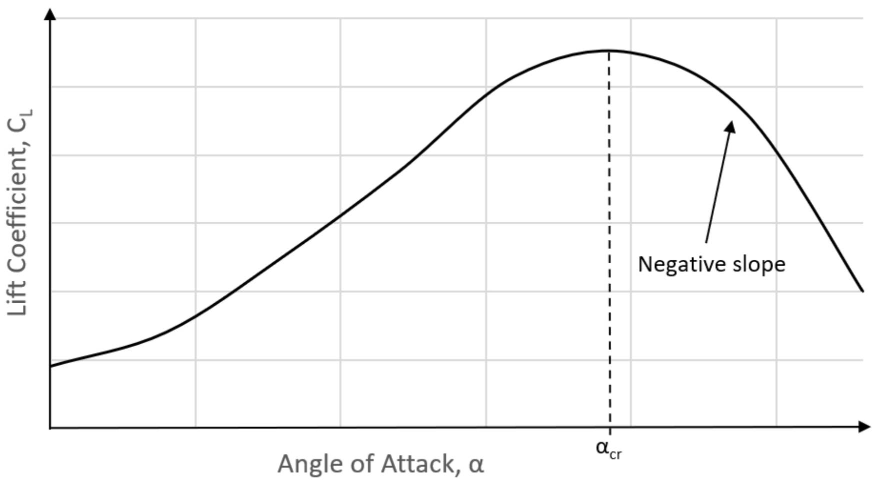


Figure 2.10: General relation between lift coefficient and angle of attack.

Galloping was first discovered by den Hartog in the 1930s, when he observed an oscillatory motion in partially ice-covered power lines. With ice build up on one side of the cable, a profile similar to an airfoil is formed altering its aerodynamic properties. Today, it is a well known phenomena in the power line industry and has been studied extensively. In recent years, super-long suspension bridges have been on the rise, especially in China, and the straits and canyons being bridged offer increasingly challenging wind conditions (Chen et al., 2020). Longer spans amount to longer main cables, and problems with galloping has arisen during the construction phase. The safety issues connected to galloping in the construction phase has been studied on the Xihoumen bridge, where it was found that problems with galloping may arise. In a finished state, the main cables are built up of hundreds of strands encased in a circular tube. While circular profiles are not susceptible to galloping, the unenclosed partially built up main cable is. The main cables are built up gradually, from the bottom up, by adding one strand at a time. So, at the early stages, the cross-section of the combined strands resembles either a triangle or a half circle. In other words, the cross-section at some stages resembles that of an airfoil. Moreover, as the main cables are not loaded during the construction phase, the cables are less tensioned and therefore are less stiff. Hence, they are more prone to galloping which can lead to safety issues.

An important parameter when studying galloping is the Scruton number, which describes the mass-damping interaction between a body and fluid (Bartoli et al., 2020). Heavy and damped bodies have high Scruton numbers, while light and undamped bodies have low numbers. For example, a cast-in-situ concrete bridge has a higher Scruton number than a steel truss bridge. There are two prevalent types of galloping, quasi-steady and unsteady galloping. The former is evaluated, with high precision, using the den Hartog instability criterion (Equation 2.2), but it requires a certain wind velocity to be valid. According to Wawzonek (1979), the influence of vortex shedding invalidates the quasi-steady theory for wind velocities below 2.5 times the wind velocity at which the lock-in phenomenon of vortex shedding occurs. For bodies that are lightweight and have low stiffness, i.e., a low Scruton number, the galloping instability threshold may be inaccurately modelled by quasi-steady theory (Mannini, 2020). A better suited theory is unsteady galloping, combining the influence of galloping and vortex shedding. However, the theory is underdeveloped due the complex interaction.

A study on unsteady galloping, analysing a pedestrian bridge in the UK, found that for bridges with low Scruton numbers, there is a strong tendency of interaction between VIV and galloping (Bagnara et al., 2017). A particular flaw of the studied bridge is the parapets, which form a U-shaped cross-section, enabling generation of wind vortices. One suggested solution was to use porous barriers as parapets which partially ventilates the trapped vortices. The galloping response improved, however, the altered wind flow decreased the critical wind velocity at which flutter arise, as the wind flow is flattened. Another study, by a Croatian research team, found the influence of wind barriers to be negligible for cable-supported bridges with regards to galloping (Buljac et al., 2017). A more recent research study focused on composite bridges, where steel box girders have been found susceptible to galloping during launching, when their Scruton number is low (Bartoli et al., 2020). Their attempt to model unsteady galloping behaviour was largely unsuccessful as the complex behaviour observed in wind tunnel testing was not replicated.

2.2.2 Flutter

For either torsional or classical flutter to ensue, a small perturbation that disturbs the equilibrium of the body is required. This perturbation often comes in the form of VIVs, as flutter is always accompanied by vortex shedding (Simiu & Yeo, 2019). The motion pattern of flutter for a bridge deck is illustrated in Figure 2.11.

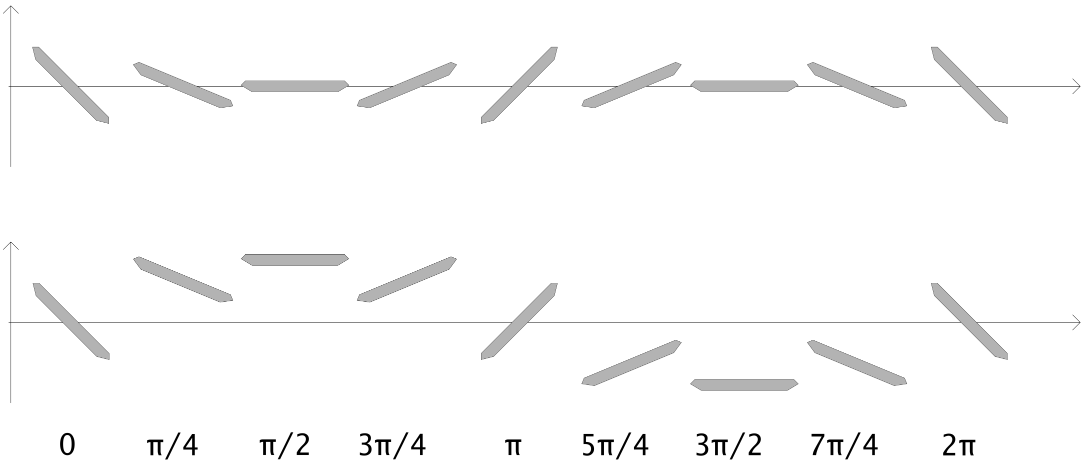


Figure 2.11: One oscillation period of the motion pattern for torsional flutter (top) and classical flutter (bottom) of a bridge deck cross-section.

The aeroelastic stability of a body describes how susceptible it is to flutter (Simiu & Yeo, 2019). Small perturbations invoke self-excited forces that returns the body to a state of equilibrium, due to mechanical damping. According to commonly used models, based on the linear model proposed by Scanlan and Tomko (1971), wind velocities exceeding a critical value, denoted as the flutter velocity, causes the self-excited forces to shift the equilibrium state of the body. This corresponds to a negative aerodynamic damping effect, resulting in growing oscillation amplitudes, i.e., divergence. This is also known as hard flutter where constant, or increasing, wind speeds always leads to structural failure. Non-divergent flutter is called soft flutter. In 1940, hard flutter led to the collapse of the original Tacoma Narrows Bridge, illustrated in Figure 2.12.

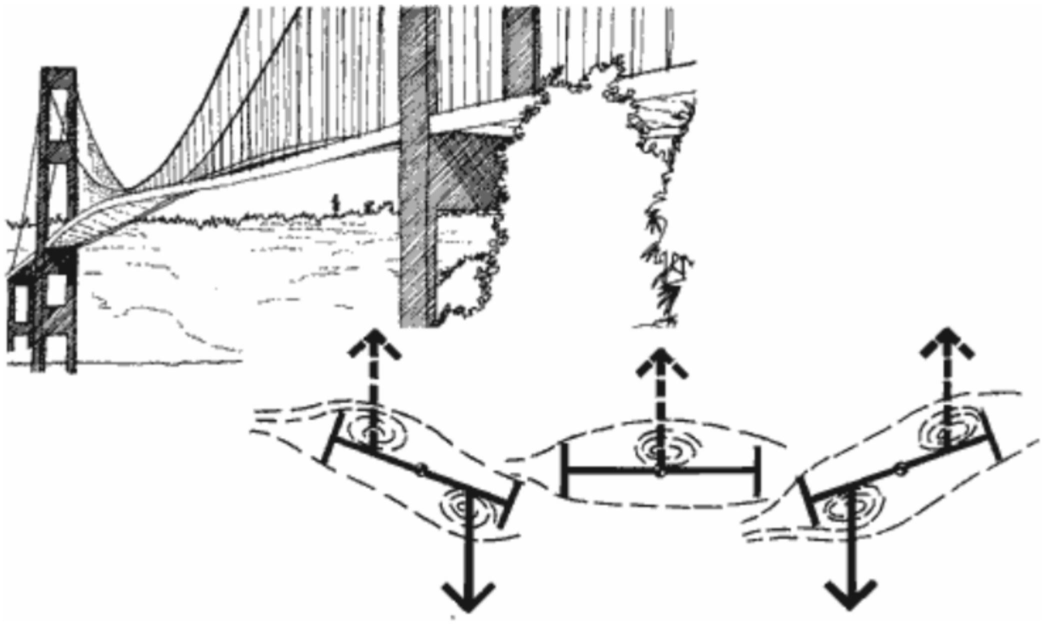


Figure 2.12: Sketch of vortices forming on the Tacoma Narrows bridge, leading to classical flutter ultimately resulting in collapse. Courtesy of Dr. Allan Larsen, chief engineer at COWI DK.

The original Tacoma Narrows bridge was given the nickname "Galloping Gertie" by the construction workers as it galloped in the wind during construction (Gaal, 2016). Ultimately, however, it was flutter that caused the structural collapse initiated by severe torsional stiffness degradation. Galloping led to a cable band, the connection between a hanger and the main cable, sliding on the main cable, creating an asymmetric hanger arrangement. This enabled the torsional motion of the deck as vortices formed within the H-shaped section, causing VIVs that lead to divergent flutter. Nowadays, H-shaped cross-sections are rarely used, due to their poor flutter performance and low torsional stiffness, resulting in a low flutter velocity. According to Simiu & Yeo (2019), the flutter velocity for bridge decks, U_c, can be determined with Equation 2.4

$$U_c = \frac{Bn_1}{K_c} \tag{2.4}$$

where B is the width of the deck, n_1 is the the fundamental frequency and K_c is the non-dimensional reduced frequency. K_c depends on aeroelastic parameters called flutter derivatives, that can only be accurately estimated with wind tunnel testing or coupled fluid-structure interaction simulations. The flutter derivatives describe the structural response in the vertical, torsional and horizontal DOFs. The torsional flutter derivative of the Tacoma Narrows Bridge, with an H-shaped cross-section having an inherently low torsional stiffness, generated negative damping for a relatively low wind velocity of 20 m/s. The day of the catastrophe, this velocity was exceeded and torsional flutter ensued leading to the dramatic collapse captured in Figure 2.13.

Figure 2.13: Collapse of the Tacoma Narrows Bridge. (James Bashford / The News Tribune, 1940)

The tools needed to analyze bridges with respect to flutter were not available in the 1940s. Scanlan and Tomko (1971) developed the first widespread methodology to evaluate flutter derivatives, also known as aerodynamic derivatives, for bridge decks in 1971. They are estimated using wind tunnel tests at a range of velocities and motion frequencies (Siedziako & Øiseth, 2018). The standard procedure involves only motion in one DOF at one velocity per test, resulting in a large number of tests required in order to obtain estimations for all derivatives.

Developments in the last decade of CFD application for bridge engineering has opened the door to numerically determine the flutter derivatives which is of great interest as wind tunnel testing is generally expensive. Gu & Zhu (2014) achieved good correlation with wind tunnel test results for both a hexagonal plate and a real bridge deck. As the Scanlan based models are only able to describe linear aeroelastic behaviour, due to their linear nature (Gao et al., 2020), the models can predict the onset of flutter but are unable to include the influence of aeroelastic nonlinearity, i.e., complex effects of higher order. This is significant for bridge decks, and especially for intricately engineered cross-sections such as twin box girders. The higher order effects may contribute with additional damping, meaning that divergence can be prevented even after the flutter velocity has been reached, i.e., soft flutter.

While available methods for CFD simulations have seen rapid development in the last decade, the accuracy is not satisfactory for the most prestigious bridge projects. Generally, all super-long and most long-span bridges are subject to wind tunnel testing to study their aerodynamic behaviour. The standard identification procedure requires several test configurations, which is expensive and time consuming. An improved identification procedure was presented by Siedziako & Øiseth (2018), where all derivatives are estimated from a single test by subjecting the body to a general random motion, activating all DOFs, and a single wind velocity. As only a single test is needed, this is a significant step of the optimization. However, a more advanced forced vibration setup and validation of test results against reference data is required.

All bridge decks have some sort of vertical obstructions in the form of traffic barriers, railings, parapets and so on. A recent study, by Bai et al. (2020), investigated the influence of the obstructions on flutter and VIVs as well as the possibility of using them as passive aerodynamic measures. In the design of the bridge deck of the Hong Kong–Zhuhai–Macau Bridge, a central upward stabilizer was used as a mitigation measure. The results of CFD simulations on the bridge deck with alternative mitigation measures are presented in Figure 2.14.

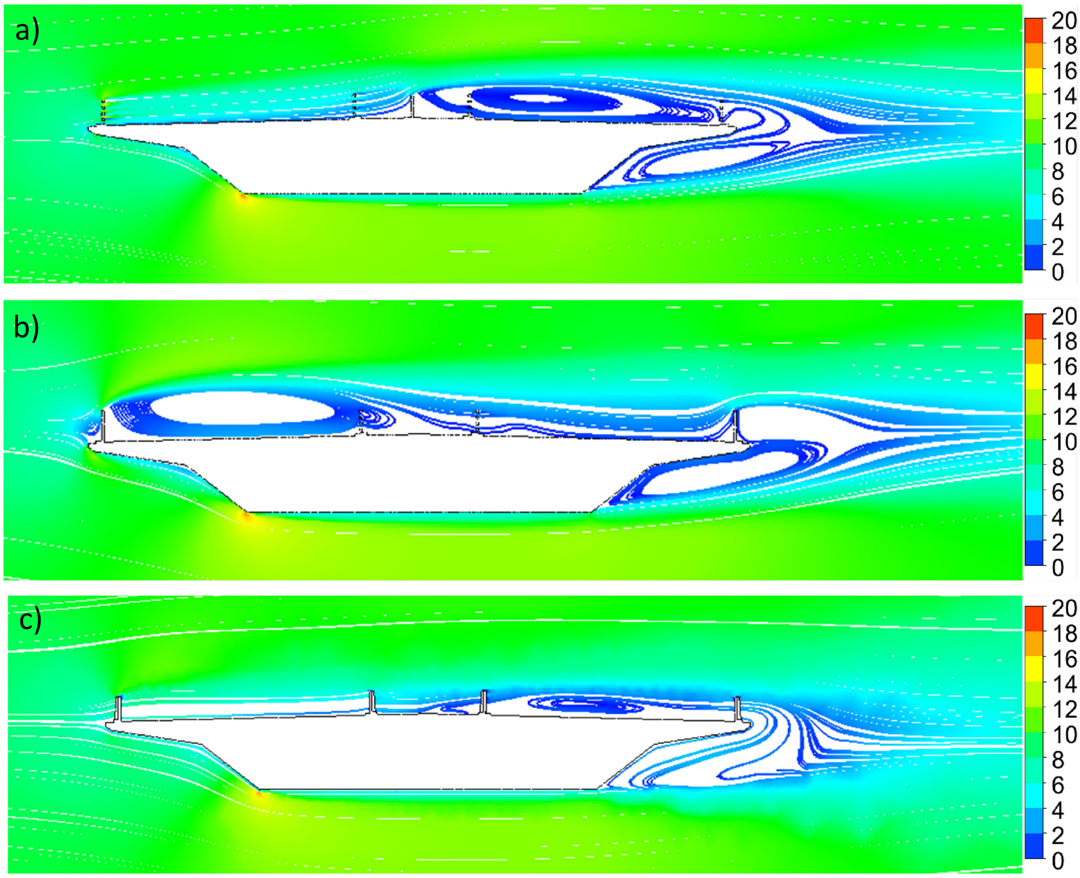


Figure 2.14: CFD simulated velocity flow fields in m/s for the bridge deck of the Hongkong-Zhuhai-Macao Bridge with a) upward central stabilizer b) sealed side traffic barrier c) partially sealed side and central traffic barrier. Courtesy of Guoji Xu, professor at Southwest Jiaotong University.

The partially sealed traffic barriers ventilates the vortices, decreasing their size significantly and thus reducing VIVs. Wind tunnel testing was conducted to find the optimal sealing form and it was concluded that improvements on both VIVs and flutter behaviour are possible with good design choices. However, at certain angles of attack the flutter velocity is reduced for partially sealed traffic barriers.

2.2.3 Torsional Divergence

Torsional divergence, or aerostatic divergence, is the result of a positive feedback loop where the angle of attack of the wind flow grows as the torsional resistance of a body is exceeded, causing rotation (Andersen et al., 2016). Due to the aeroelastic moment, caused by wind, acting with an eccentricity from its torsional centre, and with a certain angle of attack, the body rotates in order for it to obtain equilibrium. As the body rotates, the angle of attack increases, thereby increasing the aeroelastic moment and thus the rotation angle. The torsional divergence of an airfoil is illustrated in Figure 2.15.

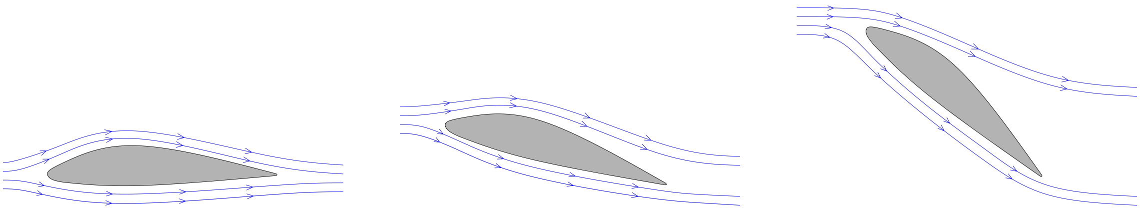


Figure 2.15: Illustration of an airfoil at a) a stable angle b) the critical angle where flow separation begins c) an angle greater than the critical one with ongoing torsional divergence, i.e., stalling.

Note that for airfoils, the divergence results in stalling as the air flows above and below are separated, giving a sudden loss of lift force. A certain relative wind velocity, together with either an angle of attack that is not parallel to the body or an initial rotation of the body, is required for the divergence to initiate. The critical torsional divergence velocity, U_{div}, depends on parameters that can be determined while the body is stationary (Simiu & Yeo, 2019). It is determined with Equation 2.5

$$U_{div} = \sqrt{\frac{2k_\alpha}{\rho B^2 \frac{dC_M}{d\alpha}\Big|_{\alpha=0}}} \tag{2.5}$$

where k_α is the torsional stiffness, ρ is the fluid density, B is the width of the body and C_M is the aerodynamic moment coefficient about the elastic axis. The phenomena of torsional divergence is in general only found in flat bluff bodies with low torsional stiffness, such as airfoils. Hence, it is also relevant for bridge decks with large width-to-height ratios, found mostly in suspension and cable-stayed bridges. An example of a structural model of a bridge deck with torsional stiffness, k_α, subjected to the wind flow, U, is visualized in Figure 2.16.

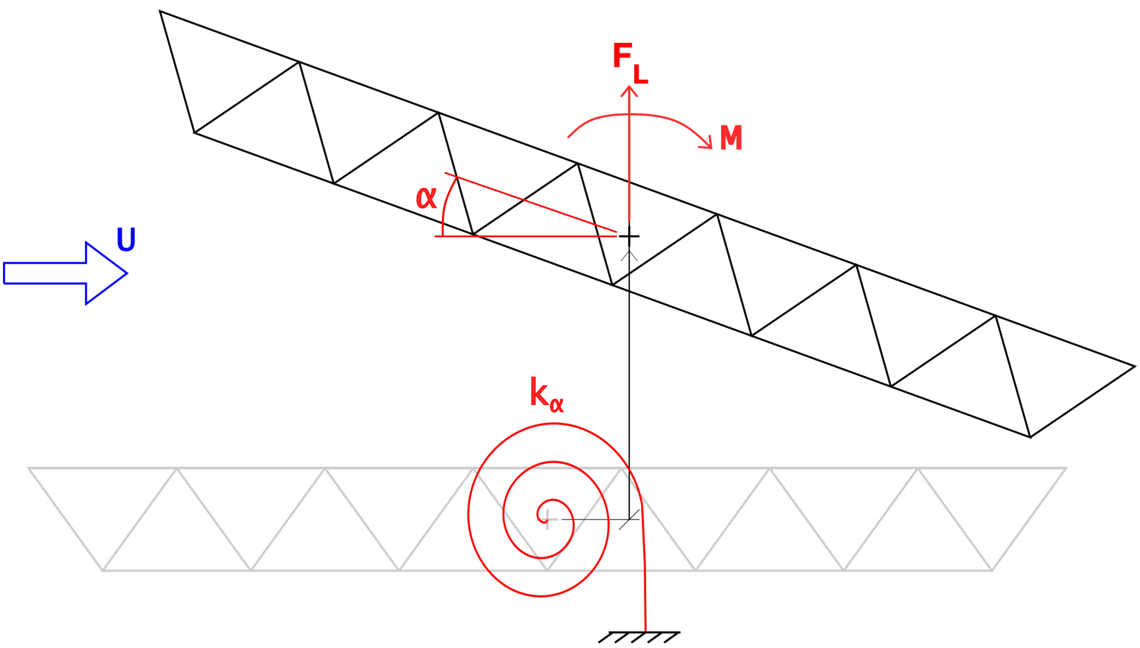


Figure 2.16: Sketch of a bridge deck undergoing torsional divergence.

It must be noted that structural collapse due to torsional divergence is only possible if the rotation of the bridge deck is restrained below a critical angle (Andersen et al., 2016). It corresponds to the angle at which the aerodynamic response of the bridge is similar to that of a stalling airfoil, where the lift force is lost. If it is unrestrained, an oscillatory motion can occur. Generally, oscillations in bridges should be minimized as they may cause excessive wear and tear in bearings and reduce the fatigue life of the structure.

A model of the Xihoumen Bridge in China was rigorously analysed through wind tunnel testing and nonlinear FEM to determine the effects of torsional divergence for super-long suspension bridges (Ge et al., 2013). Given that a significant portion of the torsional stiffness of a suspension bridge stems from the tension in the main cables, a conclusion was drawn that if the tension was lost, severe stiffness degradation would occur. Hence, if sufficient lift force on the bridge deck is generated by the wind to make the main cables stress-less, torsional divergence may occur. For the Xihoumen bridge, the critical wind velocity was determined as approximately 100 m/s, depending on the angle of attack. While it was determined not to be an issue in this case, Ge et al (2013) proposed measures to ensure tension in the main cables if the critical wind velocity is deemed too low. An elevation view of the Xihoumen bridge is presented in Figure 2.17.

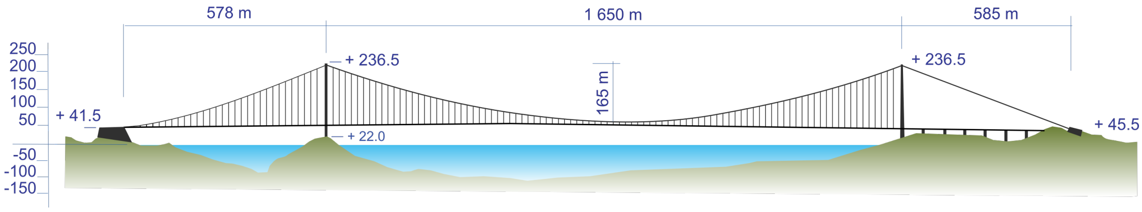


Figure 2.17: Xihoumen bridge by Roulex 45, distributed under a CC-BYSA 3.0 licence.

The current research regarding torsional divergence is primarily centred around super-long suspension bridges. For example, the previously mentioned and already built Xihoumen Bridge, the East Great Belt bridges and the proposed bridges for fjord-crossings for Coastal Highway Route E39 in Norway and the crossing of the strait of Gibraltar (Andersen et al., 2016; Andersen & Brandt, 2018). For these kinds of bridges, minimizing the mass of the bridge deck is of utmost importance to become economically feasible. However, this increases flexibility and decreases torsional stiffness. The aerodynamic instability of a triple-box girder for this purpose was investigated by Andersen & Brandt (2018), where the challenge was to ensure that neither flutter nor torsional divergence occurs. Through the means of nonlinear finite-element analysis and extensive wind tunnel testing, it was shown that satisfactory aeroelastic performance was obtained for low torsional-to-vertical frequency ratios, through the means of nonlinear finite-element analysis and extensive wind tunnel testing.

3

Current Norms for Wind Dynamic Assessment of Bridges

Eurocode 1:4, published by the Swedish Standards Institute [SIS] (2005), is the current norm for design of bridges with regard to wind actions in Sweden, and the national choices are stipulated by Transportstyrelsen (2018:57) and by Trafikverket (2019:3). Wind actions on bridges is treated in Section 8, and Section 8.2 treats dynamic effects. Section 8 is only applicable to bridges consisting of a single deck of constant depth. However, Transportstyrelsen allows for use of applicable sections of the norm as guidance for other bridge types. Trafikverket (2019:3) states that the dynamic response of bridges with spans longer than 50 metres must be assessed. An upper limit of Eurocode 1:4 is that it is not to be used for spans exceeding 200 metres. Hence, the empirical methods are limited to bridges with longest spans in the range of 50 to 200 metres, i.e, medium-span bridges. Guidance on how to assess the dynamic response for bridges within this range is given in the informative Annex E.

Due to Eurocode 1:4 being a general norm applicable to a wide range of structures by design, it is inevitable that some guidelines are less suited to specific structures. This is especially true for Annex E, where, for example, some formulae are derived for tall chimneys. Therefore, Highways England, the British counterpart to Transportstyrelsen, still use methods developed before the implementation of the Eurocodes. These methods are published in Annex A of the British Annex (BSI, 2009), and are tailored specifically for wind dynamic assessment of bridges.

For bridges not satisfying empirical requirements, the British annex recommends wind tunnel testing on scaled models. This method is a reliable option to experimentally assess the wind dynamic response of bridges, which is the norm for prestigious super-long bridges (Belloli, Diana, & Rocchi, 2014). The bridge models are in various scales, ranging from 1:200 for entire bridges and 1:20 for sections. However, the models require meticulous scaling of the material properties in order to replicate the eigenfrequencies of the real bridge. Furthermore, as the models are to scale, the wind flow and its turbulence is also scaled down. Hence, the influence of wind turbulence cannot be captured, and there may arise discrepancies. Wind tunnel testing is an alternative that is rarely used for medium-span bridges, mostly due to economical reasons as the expense of wind tunnel testing cannot be justified compared to adjusting the design.

In cases where the requirements for either vortex shedding or galloping are not met, recent advancements in the field of CFD provides alternative methods to reliably estimate certain parameters with good accuracy. Therefore CFD is becoming an increasingly viable option to assess dynamic performance as it can be used to justify higher capacity than what the empirical method in the norm predict. A promising alternative currently in development is coupled fluid-structure interaction (FSI) simulations (Braun & Sangalli, 2020). With an increased computational cost to normal CFD, it is the virtual equivalent to experimental wind tunnel testing, where the movement of the structure is simulated in conjunction with the wind flow. This option provides designers a complete tool for dynamic analyses of bridges, without the use of empirical or experimental methods. However, it is not feasible for applications in practice due to immense computational costs, especially for medium-span bridges.

3.1 Buffeting

Vibrations of the structure arising due to buffeting is not considered in the wind dynamic assessment in Eurocode 1:4 (SIS, 2005). However, the influence of buffeting is considered in Section 6, either as a structural factor to be applied on calculated static wind loads or as wake buffeting for certain conditions. Wake buffeting is irrelevant for dynamic assessments of bridges with longest spans in range of 50 to 200 metres.

3.2 Vortex Shedding

There are, in general, two parts included in the assessment of vortex shedding around bridges. The first part is related to the critical vortex shedding velocity at which lock-in occurs, and the second to assess vibration amplitudes and accelerations due to VIVs. However, the second part is only relevant if requirements in the first part are not met.

3.2.1 Empirical Method in Eurocode

Vortex Shedding is treated in the informative Annex E.1 in Eurocode 1:4 (SIS, 2005). Transportstyrelsen (2019:3), providing the Swedish national annex regarding Eurocodes on road and rail infrastructures, states that it is not allowed to use Annex E.1. The approaches described in Annex E.1 were developed for use on chimneys and similar structures, making it ill-suited for bridge design. No further guidance regarding assessment of vortex shedding is given by Transportstyrelsen. In EKS 11, the corresponding Swedish national annex for applications of the Eurocodes on buildings, the use of Annex E.1 is also prohibited. In previous editions, no further guidance was given either, but EKS 11 now refers back to the old norm BSV 97.

3.2.2 Empirical Method in British Annex

The British counterpart of Transportstyrelsen, Highways England Co. LTD, also identifies the flaws of Annex E.1. Therefore, the old national norm preceding the Eurocodes has been kept in use with minor adjustments. Alternative empirical approaches to the Eurocode, based on wind tunnel test data of bridge decks, are presented in the British Annex (BSI, 2009). The critical wind velocity check, described in Annex E.1.2, and the second approach for physical discomfort check, described in Annex E.1.5.3, are adapted. The steps are:

1. Check span length-to-height ratio. If larger than 6, continue with following steps. Otherwise, vortex shedding need not be investigated.
2. Calculate critical wind velocity for vortex shedding, v_{crit}.
3. Check that the critical wind velocity is more than 1.25 times larger than the mean wind velocity; $v_{crit} > 1.25v_m$. If the inequality holds true, no further action is needed. Otherwise, continue with Step 4 and 5.
4. Calculate maximum predicted amplitude, y_{max}, due to vortex shedding.
5. Calculate dynamic sensitivity parameter, K_D, and check against comfort criteria.

The critical wind velocity for which lock-in occurs is determined with Equation 3.1

$$v_{crit} = \frac{d_4 n_1}{St} \tag{3.1}$$

where d_4 is the cross-sectional height of the bridge, n_1 is the cross-wind fundamental frequency in bending or torsion, whichever is lowest, and St is the Strouhal number defined in Section A.1.3.2. As the Strouhal number for most bridges in this method is set to the value 1/6.5, there is potential to ascertain higher capacity if a lower Strouhal number can be justified. The annex states that the Strouhal may be gathered from an attached diagram, indicating that it may also be gathered by other means. One alternative method to determine the Strouhal number is to simulate the wind flow around a bridge section with CFD and measure the vortex shedding frequency. This frequency can then be converted to a Strouhal number which may be lower than 1/6.5.

For the critical wind velocity, an approximation of the maximum vibration amplitude, y_{max}, is determined according to Section A.1.5.4.3 with varying formulae for vertical and torsional vibrations. It is used to estimate the sensitivity parameter, K_D, determined with Equation 3.2

$$K_D = y_{max} n_1^2 \tag{3.2}$$

where y_{max} is the maximum predicted deflection and n_1 the fundamental frequency. K_D is an acceleration in mm/s^2 that is compared to criteria for physical comfort of pedestrians. Furthermore, an investigation of the structural response due to effective loading from vortex shedding should be conducted for all bridges where K_D is greater than or equal to 12.5.

3.3 Galloping

Galloping is pure longitudinal bending, corresponding to vertical translation in a section view of a bridge deck. Therefore, the vertical bending frequency of the bridge is a very important parameter when studying the phenomena. This is reflected in the empirical formulae used in both Eurocode 1:4 and the British Annex. The British annex also assesses torsional flutter of bridge decks in conjunction with galloping, unlike Eurocode 1:4 that only assesses galloping.

3.3.1 Empirical Method in Eurocode

The simplified method, proposed in Eurocode (SIS, 2005), to evaluate the risk of galloping is given in Section 2 of the informative Annex E. The major steps are:

1. Calculate onset wind velocity of galloping, v_{CG}.
2. Check that the onset wind velocity of galloping is more than 1.25 times larger than the mean wind velocity; $v_{CG} > 1.25 v_m$.
3. Check that onset wind velocity of galloping is not close to critical vortex shedding velocity; $0.7 < \frac{v_{CG}}{v_{crit}} < 1.5$.

The complexity lies in determining the onset wind velocity as it depends on some parameters found in standardized tables and figures, where the correct choice may not be obvious. It is determined with Equation 3.3

$$v_{CG} = \frac{2Sc}{a_G} n_{1,y} b \tag{3.3}$$

where Sc is the Scruton number defined in Annex E.1.3.3, $n_{1,y}$ is the first vertical natural frequency, determined approximately in Section 2 of Annex F or through solution of the eigenvalue problem with FEM. For bridge decks, this corresponds to the fundamental bending frequency. The factor of galloping instability, a_G, can be determined using Table E.7 where the width, b, is also defined based on the type of cross-section. If the cross-section shape does not correspond to those listed in the table, a_G may be set to 10. Alternatively, by determining the drag and lift coefficients for a range of attack angles using CFD, a_G can be determined with Equation 2.3. Then, a more accurate onset wind velocity of galloping can be calculated with Equation 3.3.

3.3.2 Empirical Method in British Annex

The British Annex (BSI, 2009) uses the same simplified method as Eurocode 1:4 for individual members. However, an alternative procedure developed specifically for bridge decks is given in Section A.2.4. This is useful as bridge decks are often incompatible with the cross-sections that the factor of galloping instability is given for in Eurocode. Furthermore, the galloping section is expanded by distinguishing vertical and torsional motion, corresponding to galloping, and torsional flutter, i.e., stall flutter. The steps of the procedure are:

1. Determine bridge type according to Figure A.3.

2. Calculate onset wind velocity, v_G, for torsional and, if relevant, vertical motion.

3. Calculate wind storm velocity, v_{WO}.

4. Check that the smallest onset wind velocity is larger than the wind storm velocity; $v_G > v_{WO}$.

For vertical motion, only relevant for some bridge types and with certain cross-section width-to-height ratios, the onset velocity is determined using Equation 3.4

$$v_g = v_{Rg} n_{1,b} d_4 \tag{3.4}$$

where $n_{1,b}$ is the fundamental bending frequency, d_4 is the height of the bridge cross-section and v_{Rg} is the reduced velocity defined as

$$v_{Rg} = \frac{C_g(m\delta_s)}{\rho d_4^2} = \frac{1}{2} C_g Sc \tag{3.5}$$

where C_g is a factor, defined as either 1 or 2, based on bridge type and geometry. Comparing Equation 3.4 to 3.3 from Eurocode 1:4, the similarities are evident. The factor C_g corresponds to $2/a_G$, and it simplifies the risk assessment of bridge decks as a_G does not need to be identified.

The onset velocity for torsional motion is relevant for all bridge types and is determined, depending on bridge type, by either Equation 3.6 or 3.7

$$v_g = 3.3 n_{1,t} b \tag{3.6}$$

$$v_g = 5.0 n_{1,t} b \tag{3.7}$$

where $n_{1,t}$ is the fundamental torsional frequency and b is the width of the bridge. The equations are similar to Equation 3.4, but the constants 3.3 and 5.0 have been empirically determined based on data from wind tunnel tests on a variety of bridge types.

The wind storm velocity denotes the wind velocity that the bridge must be stable for with respect to divergent amplitude phenomena. It is defined in Equation 3.8

$$v_{WO} = K_{1U} K_{1A} v_m(z) \left(1 + 2 I_v(z) \sqrt{B^2}\right) \tag{3.8}$$

where K_{1U} is an uncertainty factor with a default value of 1.1 and K_{1A} is a factor taking climactic region into consideration, which for locations in the UK, and Sweden, is set to 1.25. The turbulence intensity factor, I_v, and the background factor, B^2, are defined by Transportstyrelsen in the National Annex. The wind storm velocity is used for both galloping and flutter checks and if the criteria is not met, stability must be verified through wind tunnel testing.

3.4 Flutter

Flutter, in Eurocode 1:4 referring to classical flutter, is a coupling of bending and torsional motion of the bridge deck, i.e., galloping and torsional flutter. There is a considerable dissimilarity between how flutter is treated in Eurocode 1:4 and the British annex. Eurocode 1:4 states three conditions that indicate a risk of flutter, if all criteria is fulfilled. The British annex, on the other hand, present empirical formulae to assess the phenomena.

3.4.1 Empirical Method in Eurocode

In Eurocode 1:4 (SIS, 2005), flutter is treated in Annex E.4, together with torsional divergence. No formulae is given to determine a critical wind velocity for flutter. Instead, a structure is deemed prone to flutter and torsional divergence if three criteria are met. Otherwise, no further check is required. The criteria are:

1. The structure has a flat shape with height-to-width ratio smaller than 0.25.
2. Position of torsional axis fulfils certain geometrical conditions.
3. The fundamental frequency is torsional, or the lowest torsional frequency of the structure is lower than two times the fundamental translational frequency.

If all criteria are met there is a risk of flutter, and then Eurocode suggest seeking expert advice.

3.4.2 Empirical Method in British Annex

The British Annex (BSI, 2009) uses the same approach as Eurocode 1 for plate-like structures, but, similarly to galloping, they have developed a procedure to assess the flutter response specifically for bridge decks, in Section A.4.4. The onset velocity of flutter for bridge decks is determined in Equation 3.9

$$v_f = v_{Rf} n_{1,t} b \tag{3.9}$$

where $n_{1,t}$ is the fundamental torsional frequency and b is the cross-sectional width of the bridge deck. The reduced flutter velocity, v_{Rf}, is defined as

$$v_{RF} = 1.8\Big[1 - 1.1\Big(\frac{n_{1,b}}{n_{1,t}}\Big)^2\Big]^{1/2}\Big(\frac{mr}{\rho b^3}\Big)^{1/2} \tag{3.10}$$

where $n_{1,b}$ is the fundamental bending frequency, ρ is the air density, m is the mass per unit length and r is the radius of gyration of the cross-section. Note that v_{RF} can not be less than 2.5.

The bridge deck is considered stable with regard to flutter if the onset wind velocity for flutter is larger than the wind storm velocity, determined in Equation 3.8. If the criteria is not met, stability must be verified through wind tunnel testing.

3.5 Torsional Divergence

In Eurocode 1:4 (SIS, 2005), torsional divergence is treated in Annex E.4, in conjunction with flutter with the criteria stated in Section 3.4.1. However, unlike with flutter, a formula to determine the critical wind velocity for torsional divergence is given. It is estimated with Equation 3.11

$$v_{div} = \left[\frac{2k_\theta}{\rho d^2 \frac{dC_M}{d\theta}} \right]^{1/2} \tag{3.11}$$

where k_θ is the torsional stiffness, ρ is the density of air and d is the width of the bridge deck. The factor $dC_M/d\theta$ is the gradient of the aerodynamic moment coefficient. The critical divergence velocity should be more than two times larger than the mean velocity, $v_m(z)$.

The method in the British annex (BSI, 2009) does not differ from the method presented in Eurocode 1, where no specific section is dedicated to bridges. Therefore, it is concluded that torsional divergence is not relevant for bridges with longest spans in the range of 50 to 200 metres.

4

Quick Reference Guide

A quick reference guide for wind dynamic assessment of bridges is developed based on the empirical formulae in the British annex to Eurocode (BSI, 2009), for use by bridge engineers. It consists of a flowchart guiding the user to a number of checks, in the form of design curves, based on input data. The quick reference guide in its entirety with an accompanying background document, describing its development in detail, are presented in Appendices A and B, respectively.

4.1 Development and Example of Derivation

Four different checks are required for a complete wind dynamic assessment of bridges according to the British Annex. In total, six design curves are produced. Here, the derivation of the design curve for vortex shedding is presented. The design curves for all checks are derived in a similar manner, starting with a requirement to be fulfilled. For vortex shedding, the requirement reads

$$v_{crit} > 1.25 v_m \tag{4.1}$$

where v_m is the mean wind velocity and the critical vortex shedding velocity, v_{crit}, is calculated as

$$v_{crit} = \frac{n_1 d_4}{St} \tag{4.2}$$

The mean wind velocity depends on several other parameters, that in turn depend on site conditions. Rewriting and rearranging Equation 4.1, and inserting all parameters, yields

$$n_1 d_4 = 1.25 k_r St \cdot \ln\left(\frac{z}{z_0}\right) v_b \tag{4.3}$$

The left-hand-side is henceforth defined as the capacity for vortex shedding, $R_{d,VS}$, and the right-hand-side as the effect for vortex shedding, $E_{d,VS}$. Several choices and conservative assumptions are made to simplify the effect as to only depend on the bridges height above ground, z, and the basic wind velocity, v_b. The resulting effect is presented in Equation 4.4.

$$E_{d,VS} := 0.037 \ln\left(\frac{z}{0.05}\right) v_b \tag{4.4}$$

Using the relationship in Equation 4.4, the design curves in Figure 4.1 are produced in MATLAB.

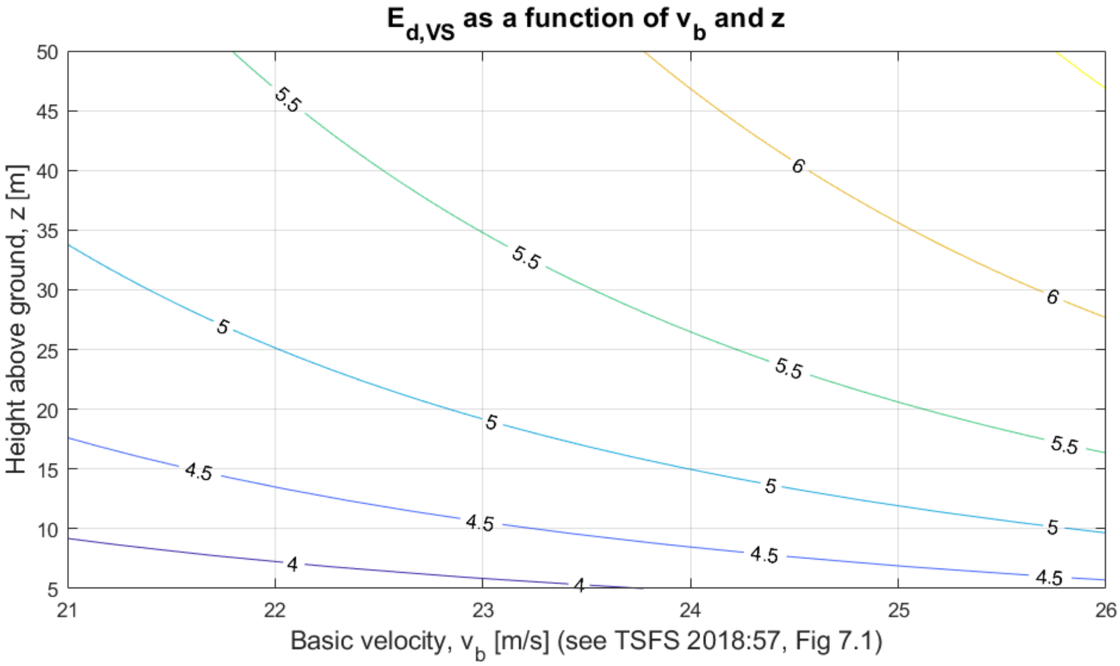


Figure 4.1: Design curves for $E_{d,VS}$ to be compared with $R_{d,VS} = n_1 d_4$. Valid for all bridge types. Linear interpolation is allowed.

Once an effect, $E_{d,VS}$, has been extracted from Figure 4.1 and a capacity has been calculated with $R_{d,VS} = n_1 d_4$, the check is performed by confirming that

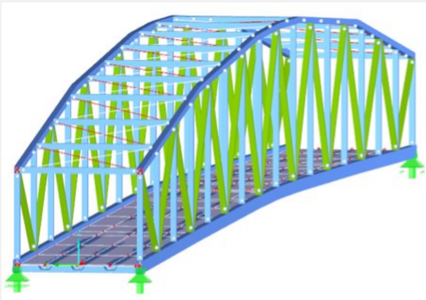
$$R_{d,VS} > E_{d,VS} \tag{4.5}$$

If the inequality is true the check has been passed, otherwise the bridge has unsatisfactory wind dynamic response. For vortex shedding in particular, investigation of vibration amplitudes must be conducted. However, for the other checks, failure to meet requirements entails redesign of the bridge. Detailed derivation procedures for all checks, where all choices and assumptions are explained, are presented in Appendix B.

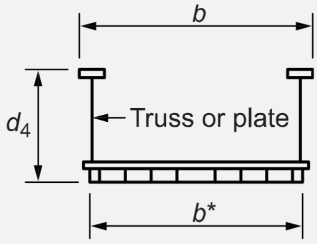
4.2 Example of Application

An application example of the quick reference guide on a bridge is presented on the following pages. Note that only two checks are required for this specific bridge. Certain bridge types in the British Annex (BSI, 2009) do not require check of galloping, and the checks for torsional flutter and classical flutter is combined into one check. Furthermore, only six unique input parameters are required to assess the wind dynamic response of a bridge.

Wind Dynamic Assessment of Pedestrian Bridge Using Quick Reference Guide



Consider the pedestrian steel truss bridge shown to the left. It has a a main span of 60 metres and is located in Falköping. The bridge deck is 4 metres wide and the truss height varies from 3 to 5 metres. However, as solid glass barriers are mounted on both sides of the otherwise permeable trusses, it is the height of the barriers that are of interest for wind flow. The distance from the top of the barriers to the bottom of the bridge is approximately 2.4 metres. The bridge sits 8 metres above the ground and the basic wind velocity in Falköping is 24 m/s, according to Transportstyrelsen (2018:57). From an FE-analysis, the fundamental frequencies for bending and torsion are determined as 2.71 Hz and 4.24 Hz, respectively.

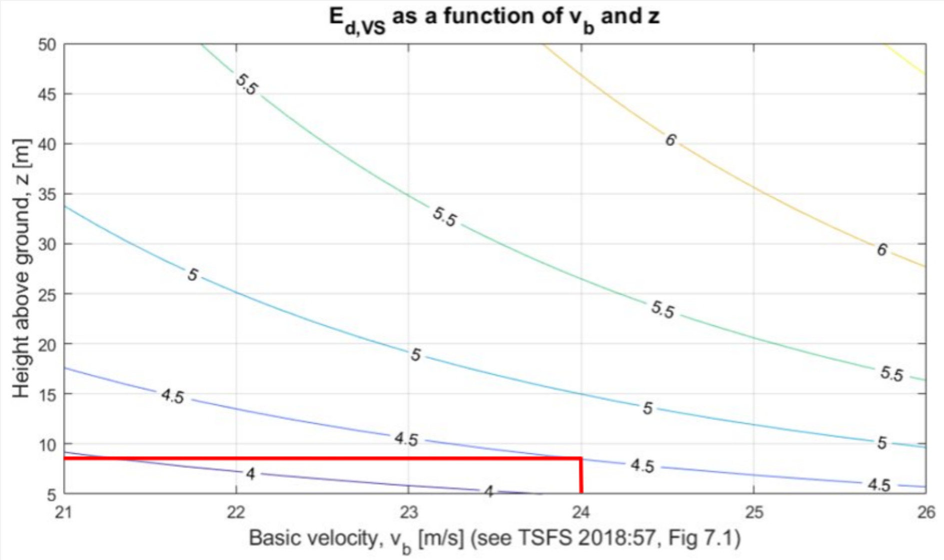


h) Bridge type 5

The first step of the quick reference guide is to confirm that the span length of the bridge is between 50 and 200 metres, which it is. Next is to determine the bridge type from Figure A.2. As seen on the side, it is identified as Bridge type 5. Also, the required input parameters are gathered and compiled in the table below.

Parameter	Variable	Value
Height above ground	z	8.0 m
Basic wind velocity	v_b	24 m/s
Cross-sectional height	d_4	2.4 m
Cross-sectional width	b	4.0 m
Fundamental bending frequency	$n_{1,b}$	2.71 Hz
Fundamental torsional frequency	$n_{1,t}$	4.24 Hz
Lowest fundamental frequency	n_1	2.71 Hz
Torsional-to-bending frequency ratio	$n_{1,t}/n_{1,b}$	1.56

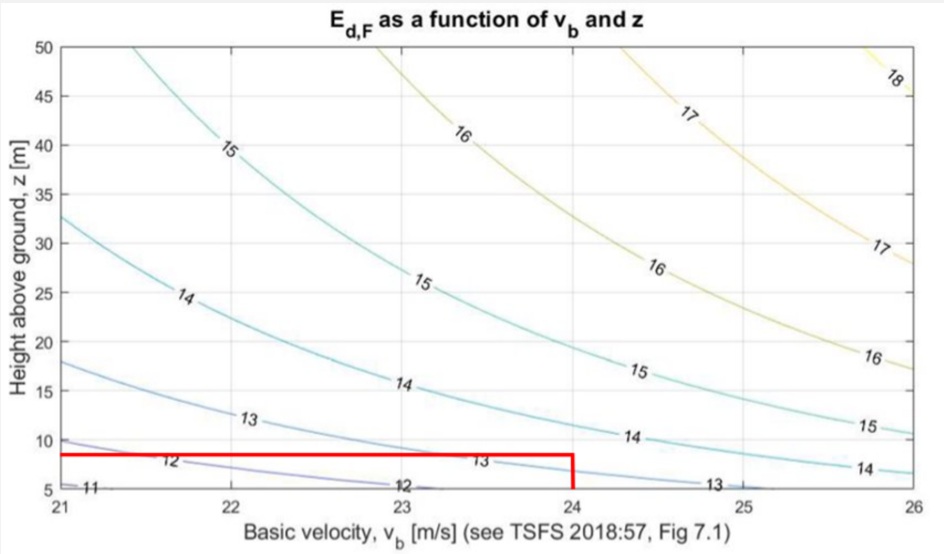
The next step is to evaluate vortex shedding. The resistance is determined as $R_{d,VS} = n_1 d_4 = 2.71 \cdot 2.4 = 6.5$ and the effect, $E_{d,VS}$, is extracted from the figure below.



The red lines intersect at approximately $E_{d,VS} = 4.5$. Hence, we can conclude that no risk of vortex shedding exists as $R_{d,VS} = 6.5 > 4.5 = E_{d,VS}$.

Moving on, we check the torsional-to-bending frequency ratio. As it is larger than 1.45, we move right in the flowchart and check if the bridge is of type 3, 3A, 4 or 4A. As it is of type 5, we evaluate flutter with Figure A.5.

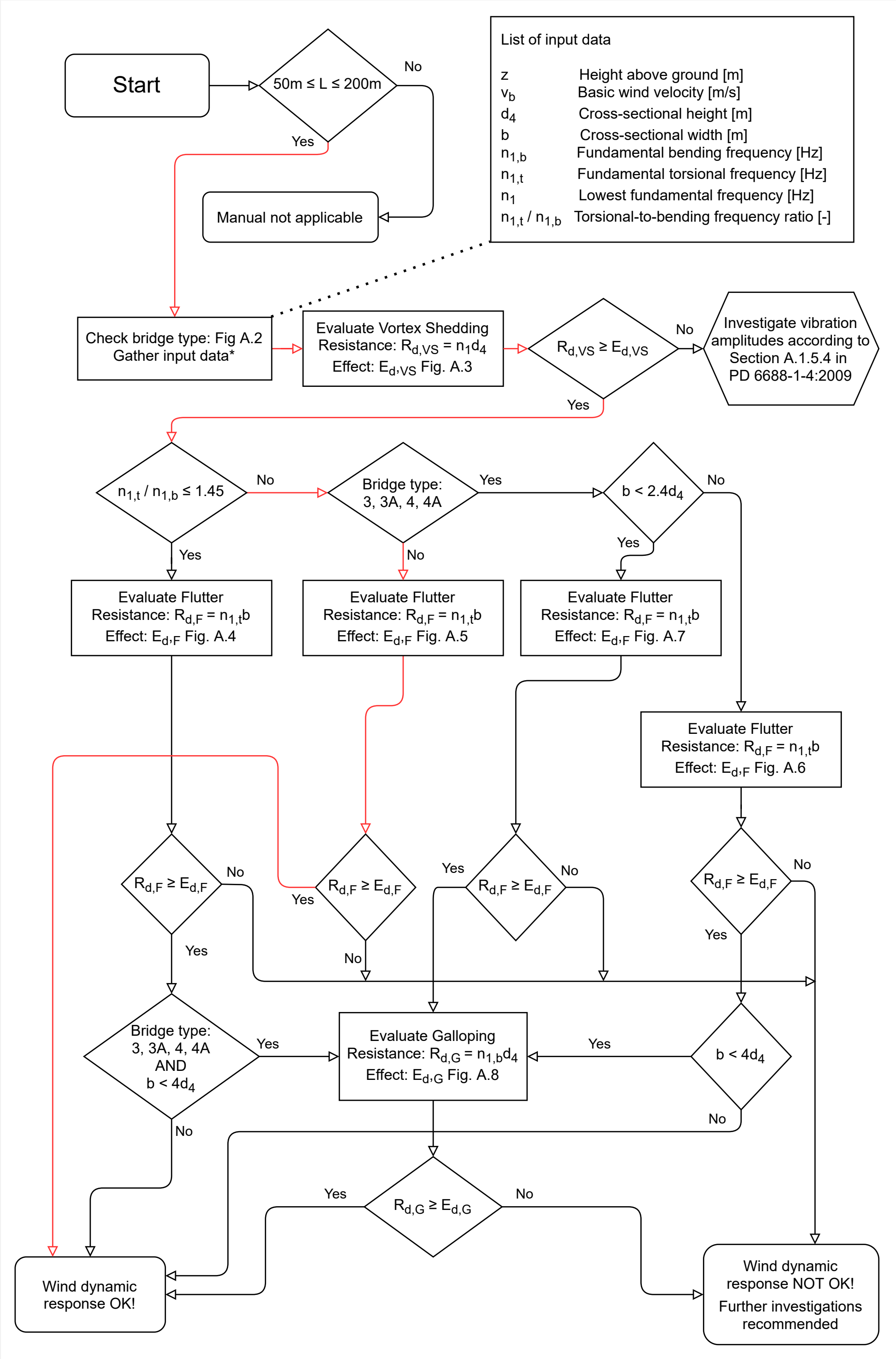
The resistance is determined as $R_{d,F} = n_{1,t}b = 4.24 \cdot 4 = 17.0$ and the effect, $E_{d,F}$, is extracted from the figure below.



The red lines intersect at approximately $E_{d,F} = 13.4$. Hence, we can conclude that no risk of flutter exists as $R_{d,F} = 17.0 > 13.4 = E_{d,F}$.

Following the flowchart, it is apparent that no more checks are necessary and the wind dynamic response of the bridge is OK!

The path taken through the flowchart is marked in red below.



4.3 Comments and Further Analysis

While the bridge in this example fulfils all checks, it is important to keep in mind that failure to meet the requirements of the quick reference guide is not always equivalent to an unsatisfactory wind dynamic response, especially if the checks are almost satisfied. If one or more checks are not fulfilled, detailed calculations according to the procedure in the British Annex (BSI, 2009), is recommended. Some guidance for such calculations can be found in the background document in Appendix B.

Studying the empirical equations of the British Annex (BSI, 2009), a possibility to justify higher capacity for vortex shedding than the norm estimates is identified. In the norm, the Strouhal number is identified as a conservative parameter, where CFD simulations could be used to determine it more precisely. According to Larsen & Walther (1998), the Strouhal number for bridges is defined as

$$St = \frac{v}{d_4 \cdot f_{cr}} \tag{4.6}$$

where v is the wind velocity in metres per second, d_4 is the bridge height and f_{cr} is the critical vortex shedding frequency.

In section A.3 of the British Annex (BSI, 2009), two figures are presented to retrieve empirical Strouhal numbers for rectangles and bridge decks, respectively. In Figures 4.2 and 4.3, slightly reproduced versions of the Strouhal number figures from the British Annex are presented.

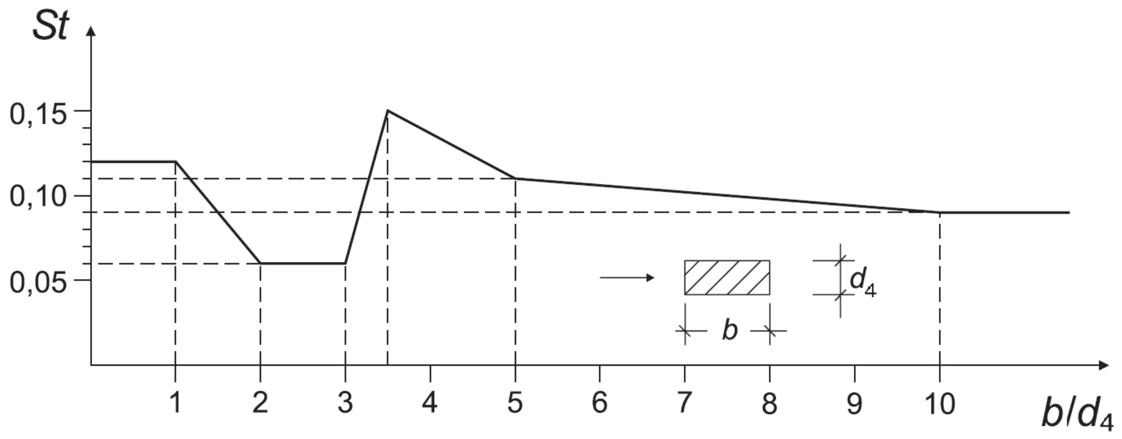


Figure 4.2: Strouhal number for rectangles with sharp corners. Reproduced from Figure A.1 of PD 6688-1-4:2009.

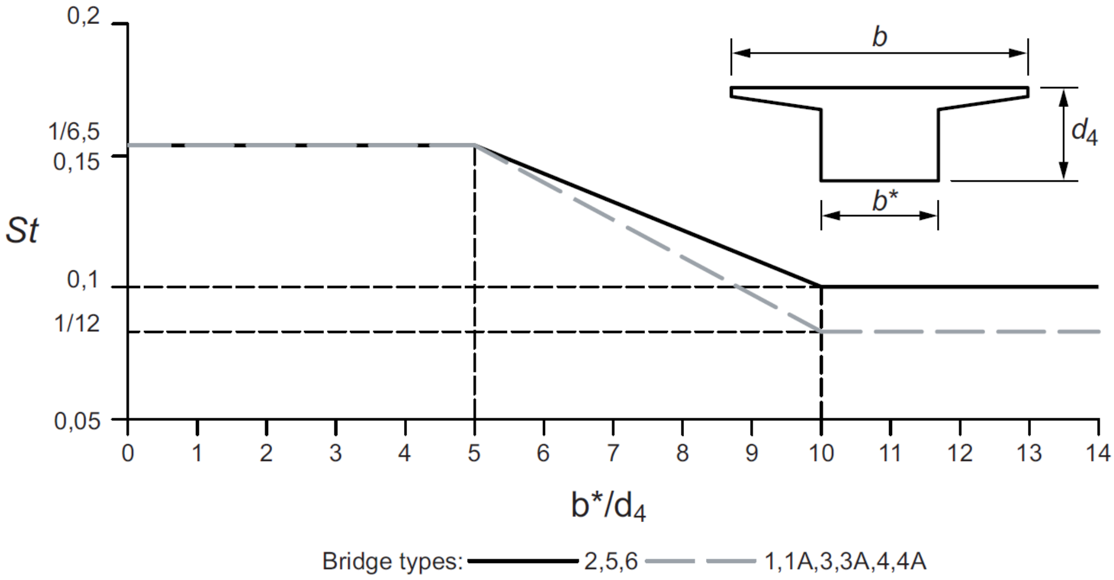


Figure 4.3: Strouhal number for bridge cross-sections. Reproduced from Figure A.2 of PD 6688-1-4:2009.

Comparing Figures 4.2 and 4.3, it is apparent that the Strouhal number for bridges with ratios b^*/d_4 below 5 are governed by the peak of Strouhal number for rectangles with ratios b/d_4 around 3.5. However, for rectangles of other ratios the Strouhal number is lower. This indicates that many bridges with ratios b^*/d_4 below 5 could have Strouhal numbers below 1/6.5. The possible increased capacity due to a decreased Strouhal number is apparent when studying the formula for critical vortex shedding velocity in Equation 4.7.

$$v_{crit} = \frac{n_1 d_4}{St} \tag{4.7}$$

As the critical vortex shedding velocity is inversely proportional to Strouhal number, a decrease in Strouhal number of 10 % yields a 10 % increase in critical vortex shedding velocity.

Part II

Numerical Analysis of Vortex Shedding

5

Computational Fluid Dynamics

The field of computational fluid dynamics (CFD) uses numerical methods to solve the Navier-Stokes equations in 3D to simulate fluid flow. Most air flows encountered in reality are of turbulent nature (Davidson, 2021). Pronounced examples can be found in the flows around and behind airplanes, and in the wake of high-speed trains. However, wind flow around buildings and other structures is also commonly turbulent.

5.1 Navier-Stokes Equation

Fluid dynamics is based on the Navier-Stokes equation, which is derived by applying Newtons law of motion on fluids. The equations describe, together with a continuity equation, the motion of all fluids in space over time. In Figure 5.1, the Navier-Stokes equation in vector form is presented. The terms, their physical significance and correlation to Newtons law of motion are highlighted.

$$\overbrace{\underbrace{\rho}_{\text{Fluid density}}}^{\text{Mass}} \cdot \overbrace{\left(\underbrace{\frac{\partial \bar{\mathbf{v}}}{\partial t}}_{\text{Change in velocity over time}} + \underbrace{(\bar{\mathbf{v}}\cdot\nabla)\bar{\mathbf{v}}}_{\text{Fluid speed and direction}}\right)}^{\text{Acceleration}} = \overbrace{\underbrace{\rho\bar{\mathbf{g}}}_{\text{External forces}} - \underbrace{\nabla p}_{\text{Pressure gradient}} + \underbrace{\mu\cdot\nabla^2\bar{\mathbf{v}}}_{\text{Internal stress forces}}}^{\text{Force}}$$

Figure 5.1: The Navier-Stokes equation, with description of the terms.

While the equation is known, it has only been smoothly solved in two dimensions by Olga Ladyzhenskaya in 1958. In three dimensions, only conditional proofs of smoothness have been presented. The Navier-Stokes equation is one of the millennium prize problems with a one million dollar reward for an unconditional solution (Clay Mathematics Institute, 2021). The complexity in the solution stems from the convection term, $(\bar{\mathbf{v}}\cdot\nabla)\bar{\mathbf{v}}$, which dominates the solution in turbulent flows. To determine whether a flow is laminar or turbulent, the Reynolds number is used. It is a dimensionless parameter, defined as the ratio between inertial and viscous force. In aerodynamics, air flow transitions from laminar to turbulent at Reynolds numbers of approximately 2300 (Davidson, 2021). In boundary layers, i.e., the thin film of air closest to the surface of an object, the transition occurs at approximately 500,000. For low Reynolds numbers, i.e., for cases where the diffusion term, $\mu \cdot \nabla^2 \bar{\mathbf{v}}$, is dominant over the convection term, $(\bar{\mathbf{v}} \cdot \nabla)\bar{\mathbf{v}}$, the equation can be smoothly solved in 3D. However, for turbulent flows, which is the most common occurrence, the flow is chaotic and currently impossible to smoothly solve.

5.2 Solving the Navier-Stokes Equations

In order to approach the solution of the Navier-Stokes equations numerically, discretization is required. The most common method is the finite volume method (FVM), where domains with known boundary conditions are isolated and partitioned into a mesh of cells with a finite volume (SimScale, 2016). While other methods are possible, such as the finite element method (FEM), the decisive advantage of using FVM is that the solution will always fulfil the continuity condition. Note that, as FVM is three dimensional by nature, true two dimensional simulations are not possible. Rather, a quasi-two dimensional domain, with a thickness of one cell, can be utilized.

In turbulent conditions, the range of the spatial and temporal scales of flows are wide (Zhiyin, 2015). Spatial scales represent vortex size and temporal scales represent velocities, dissipation and frequencies of the vortices. In Figure 5.2, the three major numerical techniques to treat the wide range are presented.

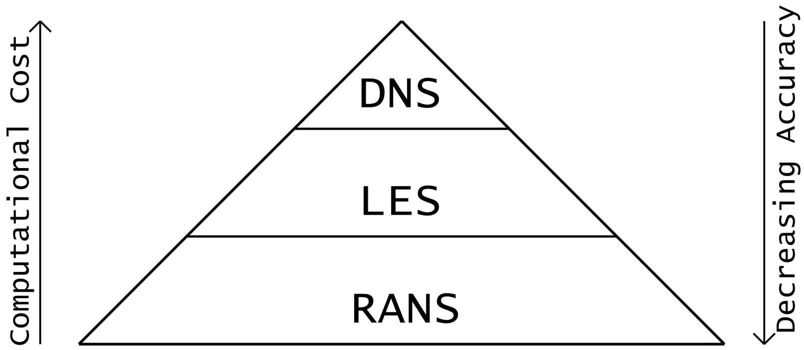


Figure 5.2: The three major numerical CFD techniques for solving the Navier-Stokes equations.

At the top of the pyramid resides direct numerical simulations (DNS), accurately resolving all turbulence scales. However, even for simple laminar cases the computational cost is very high and the use of supercomputers is necessary. As such, DNS is limited to the very front of the field and not yet applicable for industry use. Consequently, the need for approximative techniques further down the pyramid is evident. They use turbulence models to a varying extent, sacrificing accuracy in favor of lower computational cost. Large-eddy simulations (LES) filters out the smallest length scales in favour of directly computing the largest scales. Hence, the largest source of turbulence is accurately simulated but the computational cost remains significant. The third and most widespread alternative, due to its low computational cost, is the Reynolds Averaged Navier-Stokes (RANS) approach.

5.3 Reynolds-Averaged Navier-Stokes

RANS implements time averaging on the Navier-Stokes equation and utilizes turbulence models, essentially providing a steady-state solution which is sufficient for many industrial applications. Due to the time averaging, RANS is the only modelling method that is physically suitable for 2D simulations (Davidson, 2021). As the number of cells needed for 2D simulations is approximately the cube root squared of the number of cells in 3D simulations, it is apparent that 2D simulations are preferable as long as the accuracy is sufficient. However, as the eddies are enclosed, i.e., trapping energy that would otherwise dissipate in the unsolved direction, the magnitudes of the drag- and lift coefficients may be overestimated.

RANS time averages both mean and turbulent motion. A variation of RANS is unsteady-RANS (URANS), which does not time average the mean motion and is therefore able to capture variations in the mean motion of the flow (Davidson, 2021). Therefore, URANS can be used to simulate von Karman vortex streets, i.e., vortex shedding, for flows where the time scale of the mean flow is much larger than the time scale of the turbulence. For example, laminar vortex shedding in the wake of a cylinder.

5.3.1 RANS Based Turbulence Models

The three most frequently used turbulence models in the RANS family are the K-Epsilon ($k-\epsilon$), K-Omega ($k-\omega$) and K-Omega Shear Stress Transport ($k-\omega$ SST) models, with some respective variants. Common for all RANS based models is that all turbulence effects are modelled, i.e., in addition to the conservation equations, partial differential equations are used to capture turbulence history effects in the fluid. The K-Epsilon model is a two-equation model, meaning that two transport equations are used, describing the turbulent kinetic energy, k, and the turbulent dissipation rate, ϵ (SimScale, 2020a). The K-Epsilon model captures free-stream flows with good accuracy, but struggles in resolving large pressure gradients, flow separations and flows with strong curvatures, occurring, for example, near walls. The K-Omega model, similarly to the K-Epsilon model, also uses two equations to account for the turbulence history effects; one for turbulent kinetic energy, k, and one for specific turbulent dissipation rate, ω (SimScale, 2020b). It is a model appropriate for low Reynolds numbers, meaning that it resolves the field near walls well. However, this model is highly susceptible to turbulence at the inlet. An effective remedy of the respective shortcomings in both models is to combine them, and utilizing them in the regions where they are most accurate. This is the basis of the K-Omega SST model, depicted in Figure 5.3.

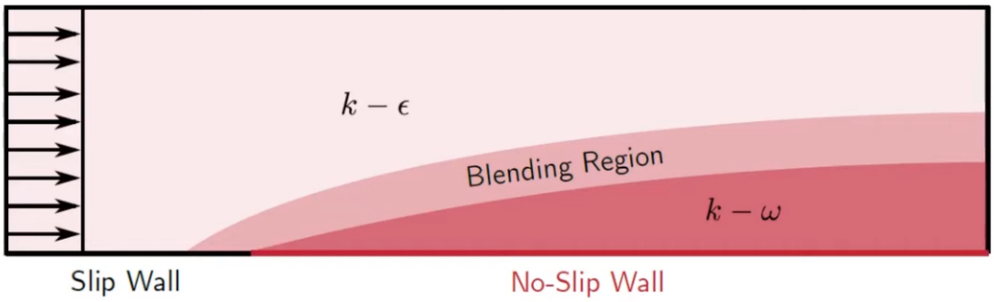


Figure 5.3: Graphical representation of K-Omega SST turbulence model. Courtesy of Aidan Wimshurst, Senior Engineer in CFD.

By blending the models their respective disadvantages are mitigated, yielding a robust and versatile turbulence model (SimScale, 2020b). For example, adverse pressure gradients and flow separation are captured with good accuracy. Hence, the turbulence model K-Omega SST is widely used in the industry.

5.3.2 Initial Turbulence Conditions

Prior to initializing simulations employing the K-Omega SST turbulence model, the initial inlet turbulence condition parameters, k and ω, need to be defined (SimScale, 2020b). The turbulent dissipation rate, ϵ, can be determined with k and ω and therefore does not need to be specified. The inlet turbulent kinetic energy is determined according to Equation 5.1

$$k = \frac{3}{2}(v \cdot I)^2 \tag{5.1}$$

where v is the mean free-stream velocity and I is the inlet turbulence intensity. For highly turbulent flows, such as wind flows, the turbulence intensity lies in the range of 5 to 20 percent. The specific turbulent dissipation rate, ω, is defined according to Equation 5.2

$$\omega = \frac{\sqrt{k}}{l} \tag{5.2}$$

where l is the characteristic turbulent length scale. Physically, the characteristic turbulent length scale corresponds to the size of the largest eddies (CFD Online, 2012). For vortex shedding in 2D, a good estimate is the cross-wind dimension of the geometry, as the size of the eddies formed behind the geometry cannot be larger than this.

5.3.3 Wall Functions

In proximity to boundaries inside the domain, denoted as walls, the fluid flow is more turbulent and viscous effects are significant, yielding sharp gradients of the velocity and dissipation profiles (Davidson, 2021). Closest to the wall, viscous effects dominate, and at a certain distance, turbulence effects described by the log-law govern the flow. To capture this behaviour, a very fine mesh can be used, but this requires significant computational resources. As these resources are finite and preferably allocated to other regions of the domain, the use of wall functions is an efficient alternative for modelling of complex flows.

Wall functions are empirically derived equations that approximate the influence of the viscous sublayer, rather than resolving it explicitly (SimScale, 2018). However, in order to obtain reliable results, certain requirements of a dimensionless distance parameter, y^+, must be satisfied. It describes the relation between the viscous stress and turbulence stress, similar to the Reynolds number. For low y^+ values, viscous stresses dominate the flow. The viscous stresses decrease with increasing values of y^+, and conversely turbulent stresses increase. At values of around 30, the viscous stresses are approximately 1 percent of the turbulence stresses. Hence, it is a lower bound for the y^+ value, assuring that the center of the boundary layer cell closest to the wall lies within the log-law region. An upper bound is set to 300, as values exceeding 300 yield poor resolution of the wall. As such, a y^+ value between 30 and 300 is a verification that the mesh resolves the wall correctly.

5.4 Hybrid LES/RANS Technique

URANS is an insufficient technique for simulations of certain phenomena, such as vortex shedding of a streamlined body. In these cases, the assumption that the time scale of the mean flow is much larger than the time scale of the turbulence is no longer true. Hence, neglecting the influence of the turbulence over time produces steady-state solutions to inherently transient phenomena. Similar to the blending of the two turbulence models K-epsilon and K-omega into the K-omega SST model, hybrid LES/RANS entails blending of LES and RANS to remedy the limitations of RANS while keeping the computational cost down (Chaouat, 2017). One such model is K-omega SST DES, where DES stands for Detached Eddy Simulation. It detects highly refined regions in the mesh where the cells are small and applies LES in them, i.e., directly simulating the large eddies. In the other regions, including boundary layers, RANS is applied. Therefore, the user controls which regions are modelled with RANS and LES, respectively. Although the computational cost of the hybrid model is relatively low, the high mesh resolution required for LES lead to a significant computational cost compared to pure RANS simulations.

Researchers Mannini & Schewe (2011) conducted a numerical study of vortex shedding around a rectangular cylinder with a width-to-height ratio of 5 using DES. Notably, this ratio is commonly used as a reference case for study of aerodynamics of bridges. A 3D approach, with limited depth, was adopted with a depth-to-width ratio of 1, meaning that some out-of-plane elements are used. The influence of numerical dissipation was studied, as well as different algorithms for discretization of inviscid fluxes. Various levels of correlation to a reference wind tunnel test was achieved, with Strouhal numbers differing by 21.6 % in the worst case, and only 2.7 % in the best case. It is concluded that 3D DES simulations with a limited depth can, with carefully chosen settings, yield reliable results, but it is sensitive to numerical dissipation.

5.5 OpenFOAM

There are many commercial CFD softwares available on the market, often coupled with hefty license fees. The open source software OpenFOAM is an attractive alternative, especially for academic applications. It is a text-based C++ library, with no graphical user interface. OpenFOAM provides a range of solvers based on the Finite Volume Method (FVM), as well as utilities for data manipulation. In Figure 5.4, a visualization of the structure of the OpenFOAM library is presented.

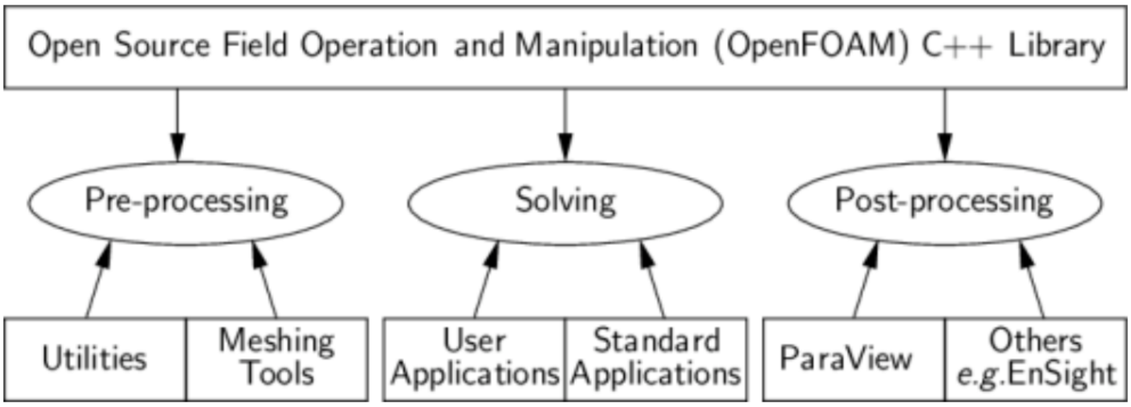


Figure 5.4: Overview of OpenFOAM structure, from the OpenFOAM user guide.

While OpenFOAM offers tools for meshing of domains, it is not sufficient for modelling of complex geometries or structures. For this purpose, CAD-programs are used to model geometries, and a mesh of the geometry is typically generated with a free meshing program, such as GMSH. However, the mesh of the geometry is not used in the simulations, rather it is a required input for creating the mesh of the fluid. This is a key difference between fluid and structural dynamic analyses, i.e., that the structure is not studied, rather the fluid flow around it. An example of this is seen in Figure 5.5, where a geometry with a U-shape is carved out of a 2D virtual wind tunnel.

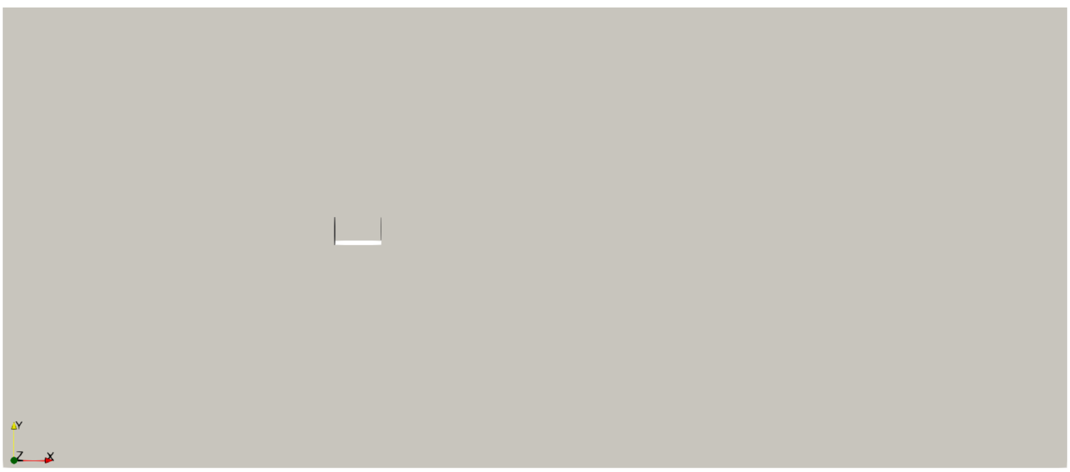


Figure 5.5: 2D wind tunnel in grey, with carved out U-shaped geometry.

In CFD simulations, there is always a choice between performing steady-state or transient analyses. Steady-state simulations are preferable as transient simulations are vastly more computationally expensive. Notably for a bridge engineer, the definition of steady-state and transient conditions differ between the fields of structural dynamics and fluid dynamics. In structural dynamics, a harmonic oscillation is considered steady-state, while in fluid dynamics a harmonically oscillating flow is considered transient. Hence, vortex shedding is a transient phenomena in fluid dynamics.

There are several different solvers in the OpenFOAM library, suitable for different types of problems. For steady-state problems of incompressible fluid flows with turbulence, such as wind load on a structure, the solver *simpleFoam* is suitable. For transient problems of incompressible fluid flows with turbulence, such as vortex shedding in wind flows, the solver *pimpleFoam* can be employed effectively.

6

Methodology for Numerical Analysis of Vortex Shedding

A methodology to analyze vortex shedding of bridge cross-sections using 2D CFD simulations is presented in this chapter. It is structured with the chronology of a simulation, covering pre-processing, solving and post-processing. In Appendix E, the directory structure and typical settings of a simulation is presented.

6.1 Pre-Processing

As meshing of the wind tunnel in OpenFOAM for simulation of vortex shedding around an object entails carving it out, it must first be created in external software. Quasi-2D geometries are modelled in the CAD program Autodesk Inventor Professional with an arbitrary depth, and exported as two separate file formats, one as *.stp* and one as *.stl*. Other CAD-programs can be used, as long as these two file types can be exported. The *.stp* file is loaded into the free meshing program GMSH, where a 2D mesh, with arbitrary refinement is generated. This mesh is then exported as a *.msh* file.

The 2D virtual wind tunnel in OpenFOAM is constructed by making use of the meshing tools and utilities *blockMesh*, *snappyHexMesh* and *extrudeMesh*. A base mesh of the wind tunnel is generated using the *blockMesh*, where the outer dimensions of the tunnel and the size of the coarsest cells is determined. The second utility, *snappyHexMesh*, is an algorithm that carves out the object by refering to the *.stl* and *.msh* files. The refinement level around the object is determined by user input. Further refinement is possible by manually defining regions in which the refinement level is increased. Generally, the refinement is highest near the object. However, the wake behind the object must also be refined in order to capture the effects of the vortices. As *snappyHexMesh* refines in all dimensions, *extrudeMesh* is used to reduce the 3D wind tunnel to quasi-2D, by extruding the mesh in the x-y plane to a thickness of one element.

6.2 Solver Settings

Computational fluid dynamic simulations are complicated and many choices, with significant influence on simulation results, need to be made. These include choice of turbulence model, initial conditions, type of solver algorithms, and tolerances.

As OpenFOAM is first and foremost a text-based C++ library, the possible choices are not always apparent. For some parameters, default values are used if a choice is omitted, which may yield incorrect simulation settings. Therefore, knowledge of available settings and their influence is paramount for reliable results.

6.2.1 Choice of Turbulence Model

An important choice for the 2D numerical analysis of vortex shedding is which turbulence model to use. The feasible alternatives are RANS or LES, where RANS is preferable due to its lower computational cost. Bruno & Khris (2003) states that RANS has only been applied to fully bluff bridge cross-sections with separated type vortex shedding. Subsequently, RANS is insufficient for reattached type vortex shedding, which requires the use of LES turbulence model. However, LES models must be used with caution in 2D as averaging, like in RANS, is not applied. Hence, the influence of 3D effects is neglected which may affect results. An alternative turbulence model is hybrid LES/RANS, specifically K-omega SST DES. By selectively applying LES modelling close to the bridge, reattached vortex shedding is captured while computational costs are kept relatively low. Hence, this model is a good compromise and is chosen to numerically analyse vortex shedding of bridges in 2D. Note that changing between K-omega SST, i.e., pure RANS, and K-omega SST DES only consists of altering the directory *turbulenceProperties* and a few lines of code, due to the similarities of the models.

6.2.2 Boundary Conditions

The boundary conditions for a quasi two-dimensional wind tunnel with flow around an object with boundary denoted wall, are presented in Figure 6.1. As this is a 2D simulation, the front and back faces are set as empty to indicate that the direction is not solved.

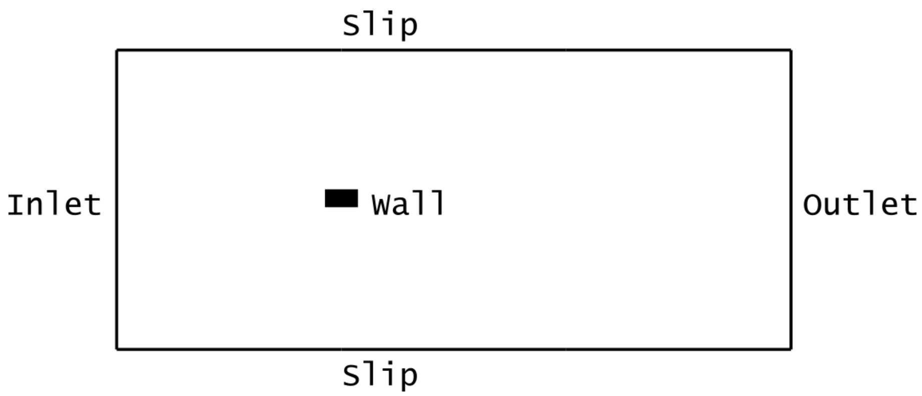


Figure 6.1: Boundary conditions for a quasi two-dimensional wind tunnel.

The inlet velocity in metres per second varies from case to case, depending on what is simulated. The conditions for the other boundaries are otherwise constant for quasi two-dimensional wind tunnels. A summary of the boundary conditions are presented in Table 6.1.

Table 6.1: Boundary conditions in OpenFOAM.

Boundary	**Parameter and Value**
Inlet (air velocity)	v
Outlet (pressure)	$p = 0$ Pa
Slip (air velocity)	$v_n = 0$ m/s (normal), and $v_t \neq 0$ m/s (tangential)
Wall (air velocity)	$v = 0$ m/s (Wall functions are used for k, ω, and $\tilde{\nu}$)

Boundary conditions for the turbulent kinetic energy, k, specific turbulent dissipation rate, ω, and turbulent viscosity, ν_t, must be specified at the wall. For the purpose of reducing the computational cost, wall functions provided in the OpenFOAM repository are used for these parameters in simulations presented in this thesis. The wall functions *kqRWallFunction*, *omegaWallFunction*, and *nutkWallFunction*, are used for k, ω, and ν_t, respectively. As wall functions are used, it is important to ensure y^+ values in the range of 30 to 300.

Before simulations are started, the inlet turbulence conditions need to be defined. In Table C.2, the equations for the turbulence parameters are presented. The turbulence intensity, I, is chosen as 15 % to represent typical wind flow. The characteristic turbulence length, l, is chosen equal to the height of the object, denoted as d_4, as this is approximately the size of the largest vortices. Note that only k and ω are needed inputs for K-Omega SST in OpenFOAM. a

Table 6.2: Inlet turbulence conditions in OpenFOAM for K-Omega SST.

Condition	**Parameter**	**Unit**
Turbulence intensity	$I = 15$	%
Inlet turbulence energy	$k = \frac{3}{2}(v \cdot I)^2$	J/kg
Characteristic turbulence length	$l = d_4$	m
Specific dissipation rate	$\omega = \frac{\sqrt{k}}{l}$	1/s

6.2.3 Solvers in OpenFOAM

The simulation procedure for vortex shedding in OpenFOAM uses two solvers. Firstly, the solver *simpleFoam* is used to find a steady-state solution of the flow field. This solution is then used as initial conditions for the transient solver *pimpleFoam* through the utility *mapFields*. If no mapping is used, the transient solver requires significantly more time to find convergence during the first seconds, if it is even possible. With *pimpleFoam*, vortex shedding initializes quickly. However, data from the first seconds should be omitted in analysis of results as it does not represent the true behaviour of the flow field.

In all transient simulations, the length of the time step is of paramount importance. Using too long time steps will cause divergence, but too short steps increases the computational cost without increasing accuracy. The solver *pimpleFoam* is initialized with a short time step. The length of the time step is then automatically increased in increments up to a limiting criteria set by the user. There are two ways for the user to limit the time step length: by defining a constant maximum time step length, or to define the Courant number. A Courant number of one means that the flow travels one cell length per time step. Hence, the time step is limited by the flow velocity and the length of the smallest cells in the mesh. In order to decrease computational costs, Courant numbers larger than one may be used but some information, of varying importance, is lost.

6.3 Post-Processing

During simulations, the flow field for specific time steps can be studied using ParaView. Furthermore, data of force coefficients can be extracted continuously and studied using MATLAB. With these tools, the user has good control of the simulations and the end time can be adjusted according to what the data indicates. When the simulation is completed, the same tools are used for post-processing of simulation results.

6.3.1 Visualization in ParaView

For post-processing of the simulation data from OpenFOAM, the open source program ParaView is the most common option. In Figure 6.2, a velocity flow field is visualized with ParaView is presented.

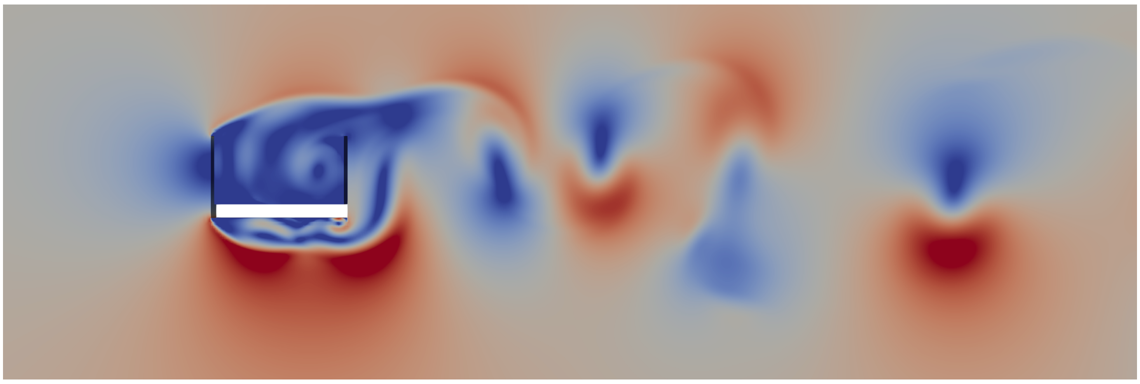


Figure 6.2: Velocity flow field around a U-shaped geometry during vortex shedding, visualized in ParaView. Blue and red colour indicate low and high velocity, respectively.

ParaView is also used to produce animations and movies to display simulations results in a sophisticated manner. It is a versatile tool, useful for visualizing all steps of a simulation, from mesh design to analysis of results.

6.3.2 Data Interpretation

The simulation data is processed and visualized with MATLAB using code developed specifically for this thesis, presented in Appendix D. OpenFOAM prints data of drag and lift coefficients for each time step into the file *coefficients.dat*. The data is loaded into MATLAB, where a fast Fourier transform (FFT) of the lift coefficient yields a frequency spectra, in which governing vortex shedding frequencies are visible. The data is sampled between two times specified in the input, preferably capturing at least ten relatively regular oscillation cycles. The lower limit of ten cycles is chosen to balance the computational time and accuracy of the results. However, a longer sampling time yields more accurate results in general. For each frequency in the range, a corresponding Strouhal number is calculated using Equation 4.6, which are plotted against the magnitude from the FFT, scaled to the governing peak. An example plot, with lift coefficient as a function of time in the top and Strouhal numbers with corresponding magnitude below, is presented in Figure 6.3.

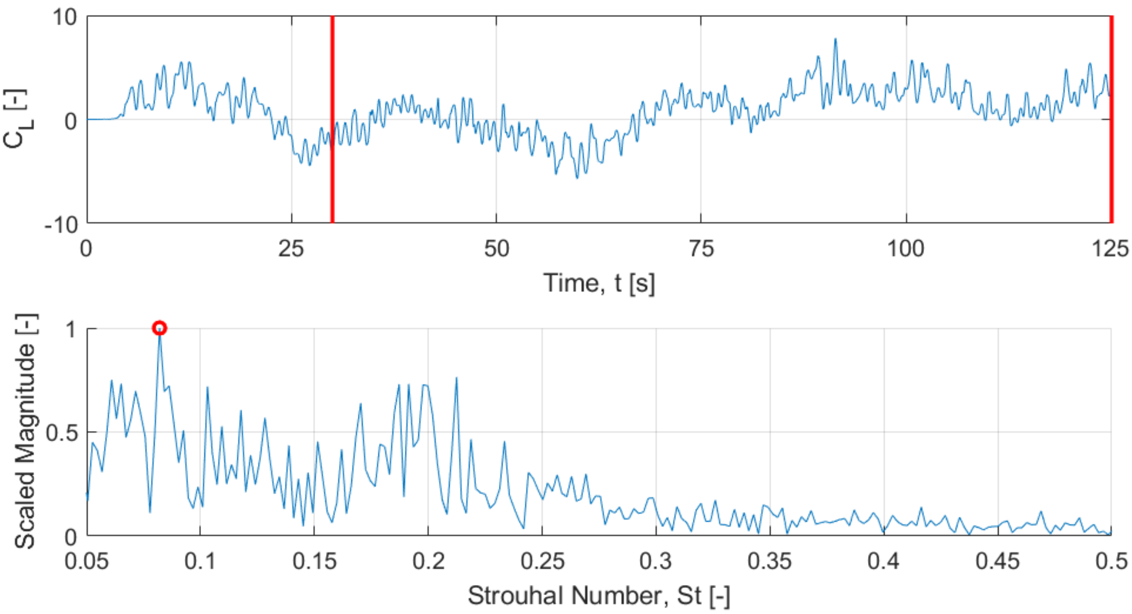


Figure 6.3: Lift coefficient variation over time during vortex shedding and Strouhals number, from an FFT of the lift coefficient.

As seen in Figure 6.3, the correct Strouhal number to extract from the simulation results may not be obvious. For this purpose, two somewhat arbitrary rules to consider are identified. In short, the Strouhal number should be in the range of 0.05 to 0.154. Firstly, peaks below Strouhal numbers of 0.05 should not be considered as they yield very unconservative estimates of the critical vortex shedding velocity. Furthermore, they sometimes describe the frequency of another unsteadiness not corresponding to the phenomena vortex shedding. In Figure 6.4, an example of such an unsteadiness is illustrated.

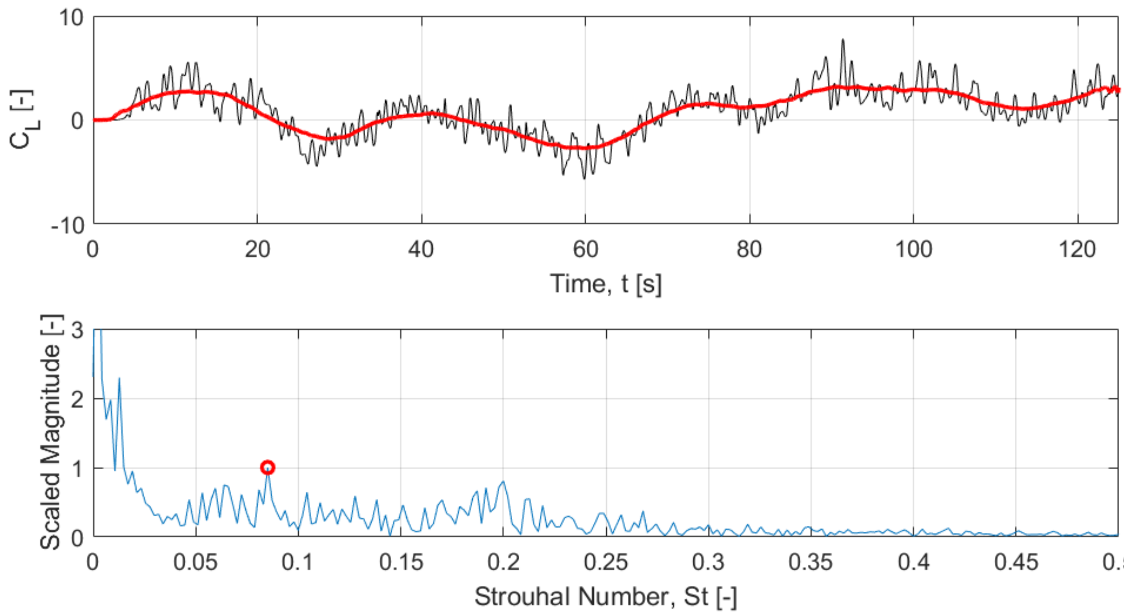


Figure 6.4: Example of application of first rule to extract Strouhals number from simulation results.

The upper graph in Figure 6.4 illustrates the lift coefficient in black and a smoothed lift coefficient in red. In the lower graph of Figure 6.4, depicting an FFT of the lift coefficient, it is apparent that the peaks with Strouhal numbers below 0.05 correspond to the variation in lift coefficient due to the moving average, which is not of interest. Hence, the Strouhal number is extracted as 0.082.

Secondly, if there are multiple clusters of peaks with Strouhal numbers above 0.05, the left-most cluster should be seen as governing and the tallest peak is extracted. This rule is based on a note in the background document to the predecessor of the British Annex, BD 49/01, treating critical vortex shedding velocities and resulting vibrations. It states that, generally, modes with higher critical vortex shedding velocity will govern the response as lower modes are dampened. This rule is used to extract the Strouhal number from Figure 6.5.

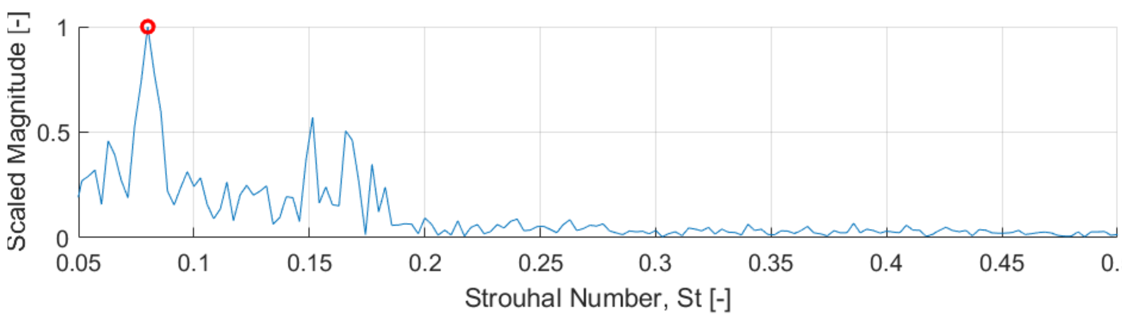


Figure 6.5: Example of application of second rule to extract Strouhals number from simulation results.

As the cluster with Strouhal numbers around 0.16 correspond to a lower mode, they are neglected. Hence, the Strouhal number is extracted as 0.084.

7

Results from Simulations of Vortex Shedding

In order to verify the OpenFOAM methodology of Chapter 6, simulation results are compared with tabulated data from the current norm, Eurocode 1:4 (SIS, 2005). Simulations of rectangles with different width-to-height ratios are presented. Furthermore, transition region from separated to reattached type vortex shedding as well as the influence of wind velocity, is studied.

7.1 Variation of Width-to-Height Ratio

Three rectangles with different width-to-height ratios are studied to verify the methodology. The geometries, boundary conditions, meshes, initial conditions and simulation results are defined and presented in detail in Appendix C. In Table 7.1, a summary of the simulation results are compiled.

Table 7.1: Simulation results of Strouhal numbers and mean drag and lift coefficients, simulated at a wind velocity of 10 m/s, together with tabulated data, for rectangles of various width-to-height ratios, b/d_4.

b/d_4	$C_{D,norm}$	$C_{D,sim}$	$C_{L,sim}$	St_{norm}	St_{sim}	ΔSt
1	2.10	2.66	0.031	0.12	0.084	-30.4 %
2	1.65	1.54	-0.359	0.06	0.058	-3.3 %
5	1.00	1.40	-0.429	0.11	0.082	-25.4 %

It is apparent that the simulation results generally agree poorly with tabulated norm values. For the rectangles with width-to-height ratios 1 and 5, both drag coefficients and Strouhal numbers have differences between simulation and tabulated data of around 30 %. However, for the rectangle with width-to-height ratio 2, the differences are significantly smaller at 6.7 % and 3.3 % for the drag coefficient and Strouhal number, respectively. For symmetric objects such as rectangles with wind flow normal to the cross-wind dimension, the mean lift coefficient should be zero. As seen in Table 7.1, it correlates poorly, especially for width-to-height ratios 2 and 5. It is possible that extending the simulation times would decrease the deviations, giving more physically accurate results. While the Strouhal number correlated well for the rectangle with width-to-height ratio 2, further development of the method is needed to increase the overall accuracy. Subsequent simulations are conducted to identify possible reasons for the discrepancies.

7.2 Transition from Separated to Reattached Type Vortex Shedding

In Chapter 6, the choice is made to use the K-omega SST DES turbulence model. However, according to Table 7.1, this model yields inaccurate results for the rectangle with a width-to-height ratio of 1. This rectangle is sufficiently bluff to produce a von Kármán vortex street, i.e., separated type vortex shedding, enabling a comparison between RANS and DES. Using identical settings except for turbulence model, results from simulations of this rectangle is presented in Figure 7.1.

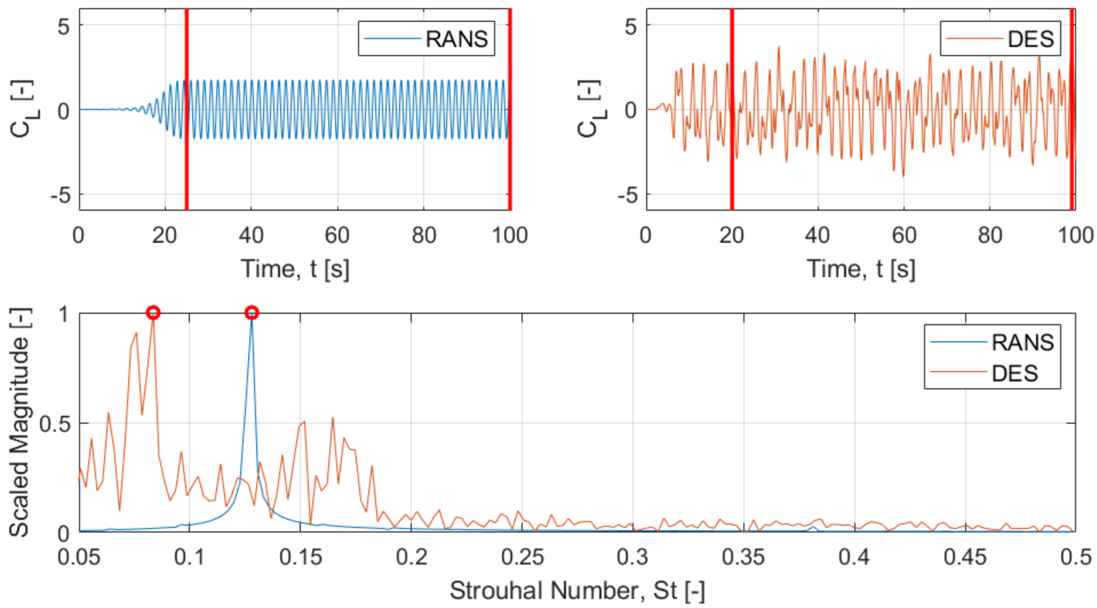


Figure 7.1: Simulation results of rectangle with width-to-height ratio of 1. Top left and right depict lift coefficients determined with RANS and DES, respectively. Bottom depicts Strouhal numbers extracted from respective lift coefficients, between times 25 to 100 and 20 to 99 seconds, respectively.

As observed in Figure 7.1, the graph of the lift coefficient from the RANS simulation exhibits periodic behaviour, with constant amplitude and a single frequency. In contrast, the lift coefficient from the DES simulation is more chaotic, with several signals of varying amplitude. Similar behaviour is observed in the lower graph depicting the corresponding Strouhal numbers. An FFT of the lift coefficient from the RANS simulation yields one single frequency and amplitude, and therefore one single Strouhal number of 0.128, which is expected based on the periodic behavior. Correspondingly, the several visible signals in the lift coefficient of the DES simulation yields several peaks, and therefore Strouhal numbers. However, one distinct peak at 0.084 is observed. Strouhal numbers, mean drag and lift coefficients, and corresponding norm values from both simulations are presented in Table 7.2.

Table 7.2: Mean drag and lift coefficients and Strouhal numbers, together with tabulated norm values, extracted from results of RANS and DES simulations of a rectangle with width-to-height ratio 1 and wind velocity 10 m/s.

Model	$C_{D,norm}$	$C_{D,sim}$	$C_{L,sim}$	St_{norm}	St_{sim}	ΔSt
RANS	2.10	2.00	-0.003	0.12	0.128	-6.67 %
DES	2.10	2.66	0.031	0.12	0.084	-30.4 %

As seen in Table 7.2, RANS yields markedly better results, across all metrics, compared to DES. Both the mean drag coefficient and Strouhal number for the RANS simulation only deviates from the norm with around 5 %. Corresponding deviations are remarkably high at around 30 % for DES, proving to be unreliable for this width-to-height ratio.

Similar simulations for a rectangle with a width-to-height ratio of 2 are performed. A comparison of the lift coefficients from the results are presented in Figure 7.2, in the top left graph using RANS, and in the top right graph using DES. Note that the results for the DES simulation is gathered from Appendix C. The wind flow velocity is 10 m/s, and the boundary conditions, mesh, and initial conditions for the RANS are identical and defined in accordance with Appendix C.

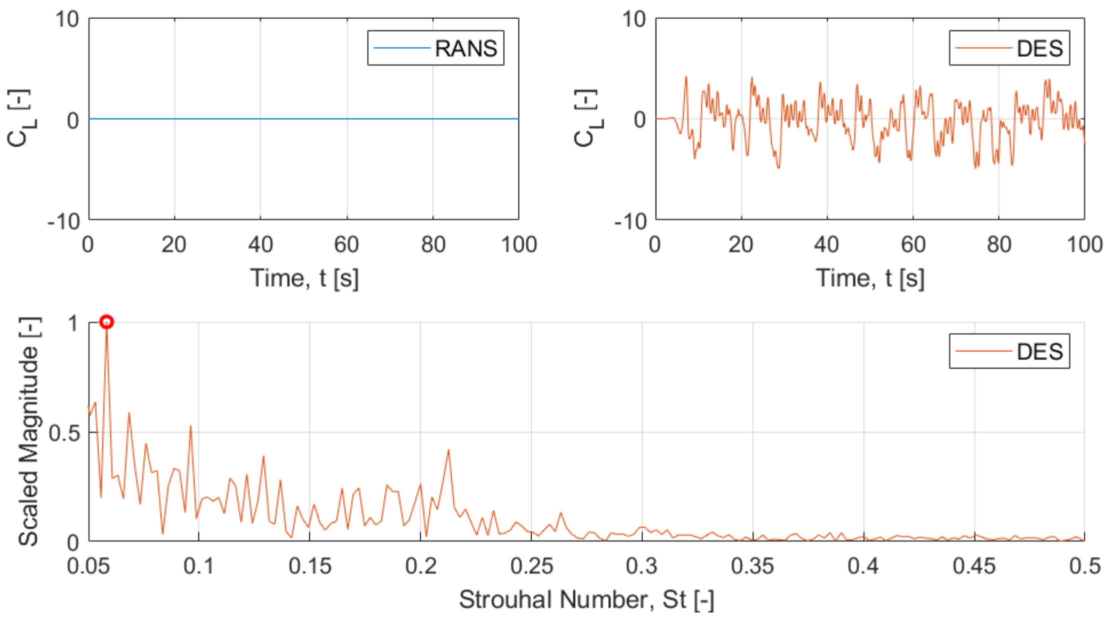


Figure 7.2: Simulation results of rectangle with width-to-height ratio of 2 and wind velocity 10 m/s. Top graphs depict lift coefficients, with RANS to the left and DES to the right. Bottom graph depicts Strouhal numbers from the DES simulation.

As seen in Figure 7.1, there are no visible fluctuations in the lift coefficient from the simulation using RANS. Hence, it appears that reattached type vortex shedding is governing for this width-to-height ratio. This indicates that the transition from separated to reattached type vortex shedding lies between width-to-height ratios of 1 and 2. On the other hand, significant fluctuations are observed from the DES simulation, able to capture this type. Given that no fluctuations are observed in the lift coefficient of the RANS simulation, no Strouhal number can be obtained.

Strouhal number for the DES simulation, mean drag and lift coefficients from both simulations, and corresponding norm values, are presented in Table 7.3.

Table 7.3: Mean drag and lift coefficients and Strouhal numbers, together with tabulated norm values, extracted from results of RANS and DES simulations of a rectangle with width-to-height ratio 2 and wind velocity 10 m/s.

Model	$C_{D,norm}$	$C_{D,sim}$	$C_{L,sim}$	St_{norm}	St_{sim}	ΔSt
RANS	1.65	1.42	0.000	0.06	N/A	N/A
DES	1.65	1.54	-0.359	0.06	0.058	-2.95 %

As seen in Table 7.3, the simulation results with a width-to-height ratio of 2 is the polar opposite to the results for the ratio of 1. For this ratio, DES yields reliable results conforming to the norm values. While not being able to capture reattached vortex shedding, RANS is expected to accurately estimate the mean drag coefficient, which it does not. However, this is not investigated further as the purpose of the method is to determine Strouhal numbers, as opposed to drag coefficients.

7.3 Influence of Wind Velocity

Three different flow velocities are studied on the rectangle with width-to-height ratio 5 to investigate the influence of wind velocity. Using the geometry, boundary conditions and mesh settings from Appendix C, the velocities 5, 10 and 20 m/s are simulated. In Figure 7.3, plots of the lift coefficients with time for each respective simulation, corresponding Strouhal numbers and average Strouhal numbers, based on all three simulations, are presented.

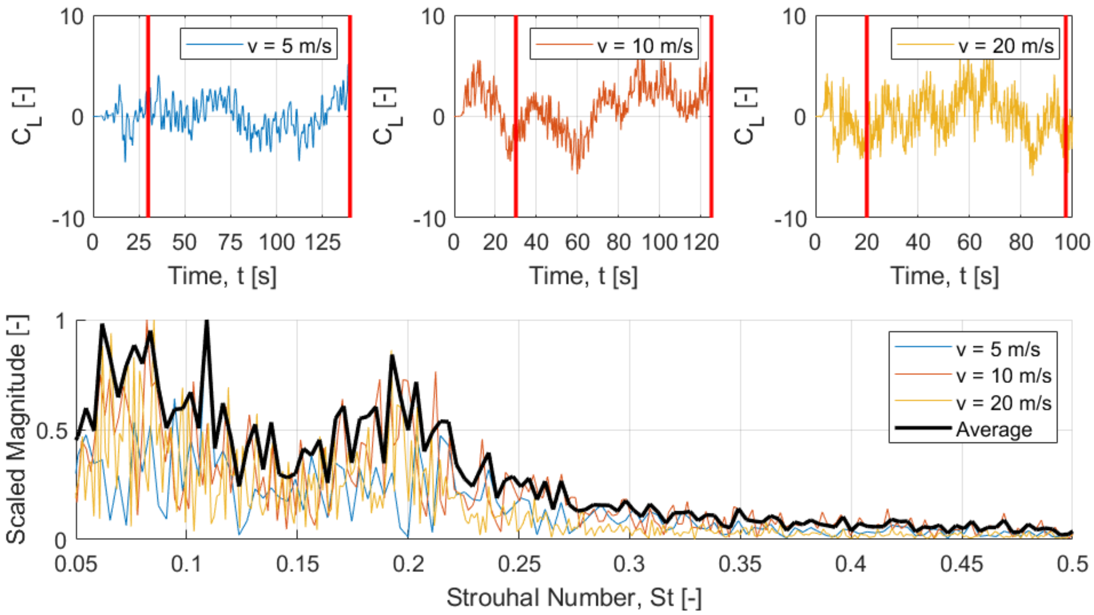


Figure 7.3: Simulation results of rectangle with width-to-height ratio of 5 with wind velocities 5, 10 and 20 m/s. The graphs at the top depict lift coefficients for the respective velocities. Bottom graph shows the average Strouhal number distribution as well as Strouhal numbers for the respective velocities.

As seen in Figure 7.3, the lift coefficients are sampled at different simulation times. This is due to the time scales of vortex shedding depending on flow velocity, where higher velocity yields higher frequencies. Hence, lower velocities require longer simulations to reach the same number of periods and are adjusted accordingly. Studying the average Strouhal number distribution, at least three peaks are of significant magnitude. Notably, there is a peak at a Strouhal number of 0.109 which correlates well with the norm, predicting 0.11. In Table 7.4, the mean drag and lift coefficents, simulated Strouhal numbers and tabulated values are presented.

Table 7.4: Mean drag and lift coefficients and corresponding Strouhal numbers for each respective wind velocity, tabulated norm values, and Strouhal number from average distribution.

Velocity	$C_{D,norm}$	$C_{D,sim}$	$C_{L,sim}$	St_{norm}	St_{sim}	ΔSt
5 m/s	1.00	1.42	0.107	0.11	0.109	-0.8 %
10 m/s	1.00	1.33	0.849	0.11	0.082	-25.4 %
20 m/s	1.00	1.25	0.275	0.11	0.085	-22.5 %
Average					0.109	-0.8 %

As seen in Table 7.4, the Strouhal number for the simulation at 5 m/s correlates well with the norm, differing by less than 1 %. On the contrary, the Strouhal numbers for 10 and 20 m/s respectively differ by more than 20 %. While the Strouhal number correlates well at 5 m/s, the drag coefficient for this simulation is farthest from the tabulated norm value. On the other hand, it has the best mean value of the lift coefficient, closest to zero. Hence, it is apparent that the wind velocity has a significant influence for the studied rectangle.

Discussion and Conclusion

8
Discussion

In this chapter, results and findings from the investigations of the empirical procedures in the norm are discussed. Also, the choices made in the development of the quick reference guide are motivated, and its potential use deliberated. Lastly, the method to numerically analyze vortex shedding with 2D CFD is examined and the implications of the simulation results are discussed.

8.1 Comparison of the Norms

The procedures for wind dynamic assessment in the current Swedish norm, Eurocode 1:4 (SIS, 2005), and the British Annex to Eurocode (BSI, 2009) have been studied and summarized. As expected, several similarities between the procedures exist. Generally, the formulae are constructed on the same basis but Eurocode needs to be applicable to many types of structures. For example, there is an issue with the Eurocode procedure for determining the risk of galloping as the factor of galloping instability, a_G, is only tabulated for a limited number of basic geometrical shapes. Furthermore, no extrapolation is allowed for non-rectangular cross-sections, which bridges often have. The consequence is high values of a_G, and in worst case 10, generating onset velocities for galloping that will rarely meet the requirements. This is remedied with the British approach for bridges that avoids the factor a_G completely by defining nine typical bridge types that most bridges conform to. Furthermore, by separating the check into both vertical and torsional motion the approach is more comprehensive and better suited for bridges.

A second example of the shortcomings of Eurocode 1:4 is how it treats flutter. Due to the complexity of modelling and predicting the structural response due to flutter, it does not provide any further guidance than referring to specialist advice if the conditions for flutter are met. On the other hand, the British Annex mandates calculations, or alternatively wind tunnel tests, of flutter for all bridges regardless if the criteria in Eurocode 1:4 are fulfilled or not. This gives the engineer a better understanding of the flutter response of the bridge deck.

A third example is how the Eurocode procedure assesses vortex shedding. The Swedish national annex prohibits the use of the procedure for assessment of vortex shedding in Eurocode 1:4 without giving further guidance, and uncertainties of how to proceed arises. As most bridge engineers are unfamiliar with wind dynamic assessments, such vagueness unnecessarily complicates the matter. In the British

Annex, however, a procedure with easily understandable formulae and figures for determining relevant input data is given.

Based on these three examples, and that the procedure is developed specifically for bridges, the use of procedure in the British Annex is strongly recommended for bridge engineers. Although the procedure in the British Annex is superior to the one in Eurocode 1:4, wind dynamic assessments according to the British Annex are still complex and it is not always straightforward which checks should be performed.

8.2 Quick Reference Guide

While studying the empirical methods of wind dynamic assessment in the norms, it became obvious that a bridge engineer might be overwhelmed by both the theory and procedure. Even in the British Annex tailored specifically for bridges, it is not immediately obvious which checks to perform. Therefore, the need for a guide with clearly defined procedure was identified. Initially, the goal was to produce a step-by-step guide for the British Annex, where each relevant check is listed together with the required input data and how to obtain it. However, as the guide was compiled, several similarities between the checks where identified and it was noted that some checks were limited to certain bridge types. A parameter study revealed the potential of creating a quick reference guide with user friendly design curves for each check. In the guide, the user is lead through the procedure with a flowchart, and directed to the relevant checks for the bridge. With this guide, the number of inputs is reduced to six unique parameters, from approximately 20, and the number of checks from four to three, and in some cases two. Hence, significant time savings are offered compared to the norm.

A crucial choice was what parameters to use on the x- and y-axis in the design curves. The height above ground, z, and the basic velocity, v_b, heavily influence the requirements in all checks and were therefore chosen. It is also beneficial that all design curves share the same axes, which was achieved by this choice. Other options are to use a product of two parameters, such as $z \cdot St$ in the design curve for vortex shedding. Certain bridges, with $b^*/d_4 > 5$, are currently evaluated over conservatively and this option would remedy that. However, this would complicate the procedure with gains only for a limited amount of bridges. With that said, there is most certainly room for improvement. The guide has been used by one bridge engineer, who gave good feedback, but more testing is needed to identify possible opportunities for improvement.

As seen in the quick reference guide, failure to fulfil the requirements for vortex shedding does not immediately imply that the wind dynamic response of the bridge is not OK. Rather, an investigation of vibration amplitudes during lock-in should be performed and compared to acceleration requirements. For many cases, it involves structural analysis where the effective loading due to vortex shedding needs to be considered. Hence, the detailed procedure for this was omitted from the quick reference guide as keeping the guide simple to use was a high priority during development. As the number of input parameters is kept low, it is especially useful in preliminary design where all parameters are not yet known. Furthermore, in design of medium-span bridges it is self-weight and traffic loads that are governing, rather than wind dynamics. More often than not, wind dynamics is merely a check to fulfil in the final stages of design. As an extensive assessment of wind dynamics requires several inputs not readily available until the final stages of the design it is often omitted in preliminary design. Furthermore, as only bridges with spans exceeding 50 metres require assessment, few bridge engineers are familiar with the procedure. Given the simplicity of the quick reference guide, it is well suited to be used in preliminary design. It can give early indications whether problems related to wind dynamics will arise down the line. Therefore, it saves time both in the control against wind dynamics and in some cases by preventing arduous redesigns in the later stages.

The quick reference guide was not primarily developed with the intention of replacing detailed calculations. However, due to conservative choices in development, there is potential for the guide to be used in practice as basis for final design. While we are confident that the guide is conservative for all practical cases, and therefore could replace detailed calculations, it has not yet been thoroughly tested. Therefore, at the time of writing, some degree of caution is advised.

8.3 Numerical Simulations using OpenFOAM

Prior to diving deeper into the possibilities of simulating vortex shedding using CFD, a choice between investigating vortex shedding or galloping was made. Based on the norm in the British annex, it was identified that galloping was only limited to a select few bridge types and further limited to certain geometrical properties. Additionally, few examples of bridges with problems related to galloping has been found in the literature. On the other hand, vortex shedding is relevant for all bridge types and a few medium-span bridges in recent years have experienced significant VIVs. Therefore, the choice was made to investigate vortex shedding with CFD.

The initial focus of the numerical analysis was to study the possibility of simulating vortex shedding using RANS due to its low computational cost. Limitations of this modelling technique were encountered when studying rectangles of increasing width-to-height ratio, for which reattached type vortex shedding occurs. Therefore, the focus was shifted towards investigating the potential of using a DES based turbulence model. As DES greatly increased the computational cost and time required for each simulation, fewer iterations and refinements of the method was possible. Since some results correlated poorly with literature and norms, it is apparent that the method requires additional work to improve reliability.

Studies on influence of width-to-height ratio and the influence of wind velocity have been performed. Comparing the DES simulations of the different width-to-height ratios, it was found that the results for ratios 1 and 5 agreed poorly to literature. In contrast, strong agreement was found for the ratio of 2, but this is not considered statistically significant. Further investigation of the ratio 1 was performed, where the simulation results indicated that RANS is a better alternative than DES for 2D simulations of separated type vortex shedding. However, as the majority of medium-span bridges have width-to-height ratios larger than 2, it is concluded that RANS is insufficient as it cannot capture the type of vortex shedding relevant for bridges. Therefore, the use of the more advanced turbulence model DES was justified.

Mannini & Schewe (2011) states that rectangles with a width-to-height ratio of 5 are common reference cases for wind dynamic investigations of bridges. Furthermore, the results from simulations with flow velocity of 10 m/s were poor. Hence, this ratio was chosen when studying the influence of wind velocity. The results from simulations at different wind velocities showed varying correlation to literature. Based on these simulations, it is evident that the wind velocity can have a significant influence on the Strouhal number, and other parameters, warranting further investigation.

Another point of interest encountered during the study of the influence of wind velocity was the complexities of data interpretation, specifically extraction of Strouhal number. In Figure 8.1, the graph of Strouhal numbers from the study on influence of wind velocity is presented.

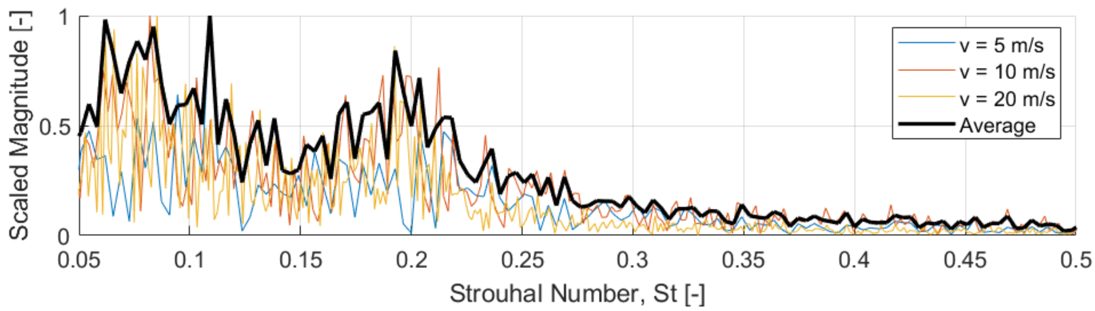


Figure 8.1: Strouhal numbers for simulations of a rectangle with width-to-height ratio of 5, at wind velocities 5, 10 and 20 m/s, and average Strouhal number distribution.

As seen in Figure 8.1, it is not immediately evident what peak is governing. There is a peak at 0.109 in the average distribution graph that correlates well to the expected value of 0.11. The peak mostly stems from the simulation at 5 m/s, but it is amplified by coinciding peaks in the other two simulations. This indicates that the peak is not a pure coincidence. Knowing how well it correlates, one is inclined to extract this value as the governing peak. However, such choices must be based on a rigorous method based on data and experience, not confirmation bias. Furthermore, the sampling interval is chosen somewhat arbitrarily based on when the lift coefficient has become sufficiently regular. The user can manipulate this interval to alter the appearance of the Strouhal number graphs, as it is sensitive to the sampling interval. This should be remedied by increasing simulation times, but this is uneconomical and inefficient. In real applications on bridges, there is no known Strouhal number, and an engineer might be tempted to choose a favourable peak and sampling interval, that yields a Strouhal numbers meeting design requirements. In the current state of the methodology, it can be ambiguous what peak to choose. Therefore, further development is needed to ensure safe design. Ideally, the subjective human input is completely removed and a piece of code extracts Strouhal numbers from the simulation results independently. However, in our experience, this is a challenging task.

In their research, Mannini & Schewe (2011) accurately simulated vortex shedding on a rectangle with width-to-height ratio 5 using DES, proving it is possible. A crucial difference to the simulation method in this thesis is the number of out-of-plane elements. While only one element is used in this thesis, effectively making the simulations 2D, Mannini & Schewe uses a light 3D approach with an out-of-plane depth equal to the width of the rectangle, discretized with a approximately 60 elements. This constitutes a significant increase in computational cost, reducing its applicability for use in the industry. Hence, the 2D method of this thesis aimed to achieve as good accuracy as possible while remaining feasible without needing excessively expensive computer hardware. There is a possibility that the 2D approach employed is insufficient regardless of settings, but the method has not yet been investigated to such an extent that a conclusion can be drawn.

There are several points suitable for further research. Firstly, the number of simulations performed with DES are too few. It would have been desirable to do at least three simulations for each parameter studied to draw well informed conclusions. Furthermore, the influence of inlet turbulence parameters has not been investigated. For instance, the turbulent length scale, i.e., the size of the largest eddies, affects the dissipation of energy in the system. It was approximated based on the cross-wind dimension of the studied objects, but little guidance on how to determine it exist. Moreover, other settings in OpenFOAM that influence the simulation such as number of iterations per time step needs further study.

Another interesting topic to investigate further is the influence of non load-carrying parts of the bridge, such as railings and traffic barriers, as well as traffic on the bridge. Presumably, such obstructions influence the wind flow around a bridge, but the extent of which is unknown. In their study, Bai et al. (2020) found that partially sealed traffic barriers ventilates the vortices, decreasing their size significantly and thus reducing VIVs. Corriols (2015) investigated the influence of traffic on wind flow around bridges with CFD simulations. The results showed a large influence on all studied parameters, including the Strouhal number which was significantly decreased. Therefore, it is obvious that not only the main cross-section of a bridge will influence the wind flow, instigating further research.

A valid criticism of the methodology to study vortex shedding presented by this thesis is the chosen Courant number. In all simulations, a maximum Courant number of 5 and a limiting maximum time step of 0.01 seconds, has been employed. Although, time step length has always been governed by the Courant number. However, the K-omega SST DES turbulence model partially uses LES for which Gerasimov (2016) recommends maximum Courant numbers of 0.5, for good accuracy. As a Courant number of 5 is used, information that greatly impacts the results is lost if the size of the cells where LES is employed is less than 5 times larger than the size of the boundary layer cells. Alterations of the flow velocity in simulations calls for increased or decreased refinement of the boundary layers to achieve y^+ values within the range of 30 to 300, in order to use wall functions. Therefore, the size of the boundary layer is decreased for lower flow velocities resulting in information loss in cells were LES is used.

Following this reasoning, it seems logical to decrease the Courant number to increase accuracy. However, simulations of the flow around the relatively non-complex geometry of a rectangle using a maximum Courant number of 5 resulted in simulation times of around one week, using a PC with 16 GB RAM and an Intel Core i7 processor at 4 GHz. Hence, if the recommended value of 0.5 for LES applications had been used, the simulation time would be approximately ten times longer. Additionally, as the aim of the method is to apply it to bridges with more complex geometries, even longer computational times are expected. As the method is aimed for bridge engineers to use in practice, computational costs are a key parameter which has influenced many choices in this thesis. Computational cost is always at odds with accuracy and reliability and the method of this thesis is, judging by simulation results, not accurate enough in its current state.

9 Conclusion and Recommendations on Further Studies

The purpose of this thesis was to map and present available assessment methods in the field of wind dynamics to bridge the knowledge gap. As the field it lies in-between structural engineering and fluid dynamics, bridge engineers generally are unfamiliar with it. Therefore, the aim was to investigate procedures for wind dynamic assessment of medium-span bridges. In this regard, the thesis was successful as the field has been summarized and presented in a digestible manner.

The first part of the thesis was devoted to investigate and define the relevant wind dynamic phenomena, both in theory and in practice. Based on findings, the method for wind dynamic assessment of bridges in the British Annex (BSI, 2009) is recommended. Additionally, a quick reference guide for use by bridge engineers was produced based on this method. The guide could prove to be a useful tool, allowing for quick assessments indicating possible issues in early design stages.

The second part of the thesis was dedicated to the investigation of the possibility to analyse vortex shedding using CFD. The method was developed for use by bridge engineers in practice where computational resources are limited. Much progress has been made, and the method has been narrowed down to one turbulence model that can be effectively employed. Some simulations showed promise, but over all the results were not reliable enough to be of use in practice. However, with further development of the method, it should be possible to justify higher capacity for vortex shedding than what the norm predicts with good reliability.

Regarding further studies, there are three main points of interest. Firstly, thorough testing of the quick reference guide through application on a large number of bridges, side by side with detailed calculations, is of importance to verify its reliability. Then, it could be motivated to use it as a replacement of the calculations in real projects, even in the final design. Secondly, methods for estimation of vibration amplitudes due to vortex shedding and possible implementation in the quick reference guide, further simplifying wind dynamic assessments of bridges, could be investigated. Thirdly, further work on the method for simulating vortex shedding with CFD is needed. Promising results show that potential for improvement exist, and with refinement, it could be a useful tool for bridge engineers.

Bibliography

Andersen, M. S., Brandt, A., Hansen, S. O., & Johansson, J. (2016). Aerodynamic stability of long span suspension bridges with low torsional natural frequencies. *Engineering Structures, 120*. pp. 82-91. https://doi.org/10.1016/j.engstruct.2016.04.025

Andersen, M. S., & Brandt, A. (2018). Aerodynamic Instability Investigations of a Novel, Flexible and Lightweight Triple-Box Girder Design for Long-Span Bridges. *Journal of Bridge Engineering, 23*(12). https://doi.org/10.1061/(ASCE)BE.1943-5592.0001317

Bagnara, A., Cammelli, S., & Navarro, J. G. (2017). VIV-galloping interaction of the deck of a footbridge with solid parapets. *Procedia Engineering, 199*. pp. 1290-1295. https://doi.org/10.1016/j.proeng.2017.09.313

Bai, H., Ji, N., Xu, G., & Li, J. (2020). An alternative aerodynamic mitigation measure for improving bridge flutter and vortex induced vibration (VIV) stability: Sealed traffic barrier. *Journal of Wind Engineering and Industrial Aerodynamics, 206*. Article e2020.104302. https://doi.org/10.1016/j.jweia.2020.104302

Bartoli, G., Chen, C., Mannini, C., & Thiele, K. (2020). Experimental study and mathematical modeling on the unsteady galloping. *Journal of Wind Engineering and Industrial Aerodynamics, 203*. Article e2020.104170. https://doi.org/10.1016/j.jweia.2020.104170

Battista, R. C., & Pfeil, M. S. (2000). Reduction of vortex-induced oscillations of Rio-Niterói bridge by dynamic control devices. *Journal of Wind Engineering and Industrial Aerodynamics, 84*(3). pp. 273-278. https://doi.org/10.1016/S0167-6105(99)00108-7

Belloli, M., Diana, G., & Rocchi, D. (2014). Wind tunnel: a fundamental tool for long-span bridge design. *Structure and Infrastructure Engineering, 11(4)*. https://10.1080/15732479.2014.951860

Bourguet, R., Karniadakis G. E., & Triantafyllou, M. S. (2011). Lock-in of the vortex-induced vibrations of a long tensioned beam in shear flow. *Journal of Fluids and Structures, 27*(5-6), pp. 838-847. https://doi.org/10.1016/j.jfluidstructs.2011.03.008

Braun, A. L., & Sangalli L. A. (2020) A fluid-structure interaction model for numerical simulation of bridge flutter using sectional models with active control devices. Preliminary results. *Journal of Sound and Vibration, 477*. Article e2020.115338. https://doi.org/10.1016/j.jsv.2020.115338

British Standards Institute [BSI]. (2009). *Background information to the National Annex to BS EN 1991-1-4 and additional guidance* (PD 6688-1-4:2009).

Bruno, L., & Khris, S. (2003). The validity of 2D numerical simulations of vortical structures around a bridge deck. *Mathematical and Computer Modelling, 37*. pp. 795-828. https://doi.org/10.1016/S0895-7177(03)00087-6

Buljac, A., Kozmar, H., Pospíšil, S., & Macháček, M. (2017). Flutter and galloping of cable-supported bridges with porous wind barriers. *Journal of Wind Engineering and Industrial Aerodynamics, 171*, pp. 1717-1737. https://doi.org/10.1016/j.jweia.2017.10.012

CFD Online. (15 June 2012). Turbulence length scale. CFD Online. https://www.cfd-online.com/Wiki/Turbulence_length_scale

Chaouat, B. (2017). The State of the Art of Hybrid RANS/LES Modeling for the Simulation of Turbulent Flows. *Flow Turbulence Combust 99, (279–327)*. https://doi.org/10.1007/s10494-017-9828-8

Chen, Z., Huang, Z., Hua, X., Tang, Y., & Wang, C. (2020). Post-critical behavior of galloping for main cables of suspension bridges in construction phases. *Journal of Fluids and Structures, 101*. Article e2020-103205. https://doi.org/10.1016/j.jfluidstructs.2020.103205

Cheynet, E., Jakobsen, J. B., & Snæbjörnsson, J. (2016). Buffeting response of a suspension bridge in complex terrain. *Engineering Structures, 128*. pp. 474-487. https://doi.org/10.1016/j.engstruct.2016.09.060

Clay Mathematics Institute. (March 30, 2021). *The Millennium Prize Problems*. https://www.claymath.org/millennium-problems/millennium-prize-problems

Corriols, A. S. (2015). *Vortex-induced vibrations on bridges: the interaction mechanism in complex sections and a new proposed semi-empirical model.* [Doctoral dissertation, Technical University of Madrid], UPM Digital Archive. http://oa.upm.es/39052/

Davidson, L. (2021). *Fluid mechanics, turbulent flow and turbulence modeling.* https://tinyurl.com/wt9rmfaw

De Miranda, M. (2016). Long-span bridges. In A. Pipinato (Ed.), Innovative Bridge Design Handbook: Construction, Rehabilitation and Maintenance (pp. 383-425). Oxford: Butterworth-Heinemann

Dielen, B., & Ruscheweyh, H. (1995). Mechanism of interference galloping of two identical circular cylinders in cross flow. *Journal of Wind Engineering and Industrial Aerodynamics, 54-55*. pp. 289-300. https://doi.org/10.1016/0167-6105(94)00048-I

Gaal, R. (2016). November 7, 1940: Collapse of the Tacoma Narrows Bridge. *APS News, 25*(10). https://www.aps.org/publications/apsnews/201611/physicshistory.cfm

Gao, G-Z., Zhu, L-D., & Zhu, Q. (2020). Recent advances, future application and challenges in nonlinear flutter theory of long span bridges. *Journal of Wind Engineering and Industrial Aerodynamics, 206*. Article e2020.104307. https://doi.org/10.1016/j.jweia.2020.104307

Ge, Y. J., Yang, Y. X., & Zhang, Z. T. (2013). Torsional stiffness degradation and aerostatic divergence of suspension bridge decks. *Journal of Fluids and Structures, 40*. pp. 269-283. https://doi.org/10.1016/j.jfluidstructs.2013.05.001

Gerasimov, A. (May 2016). Quick Guide to Setting Up LES-type Simulations [PowerPoint-presentation]. ANSYS Sweden AB. https://tinyurl.com/4szzzw7b

Gimsing, N. J., & Larsen, A. (1992). Wind Engineering Aspects of the East Bridge Tender Project. *Journal of Wind Engineering and Industrial Aerodynamics, 42*(1-3), pp. 1405-1416. https://doi.org/10.1016/0167-6105(92)90148-4

Gu, M., & Zhu, Z. (2014). Identification of flutter derivatives of bridge decks using CFD-based discrete-time aerodynamic models. *Wind and Structures An International Journal 18*(3). https://doi.org/10.12989/was.2014.18.3.215

Gu, M., Wang, X., & Quan, Y. (2020). Wind tunnel test study on effects of chamfered corners on the aerodynamic characteristics of 2D rectangular prisms. *Journal of Wind Engineering and Industrial Aerodynamics, 204*. Article e2020.104305. https://doi.org/10.1016/j.jweia.2020.104305

Larose, G. L., & Larsen, A. (2015). Dynamic wind effects on suspension and cable-stayed bridges. *Journal of Sound and Vibration, 334*. pp. 2-28. https://doi.org/10.1016/j.jsv.2014.06.009

Larsen, A., & Schewe, G. (1998). Reynolds number effects in the flow around a bluff bridge deck cross section. *Journal of Wind Engineering and Industrial Aerodynamics, 74-76*. pp. 829-838. https://doi.org/10.1016/S0167-6105(98)00075-0

Larsen, A., & Walther, J. H. (1998). Discrete vortex simulation of flow around five generic bridge deck sections. *Journal of Wind Engineering and Industrial Aerodynamics (77-78)*. pp. 591 - 602. https://doi.org/10.1016/S0167-6105(98)00175-5

Mannini, C., & Schewe, G. (2011). Numerical study on the three-dimensional unsteady flow past a 5:1 rectangular cylinder using the DES approach.

Mannini, C. (2020). Incorporation of turbulence in a nonlinear wake-oscillator model for the prediction of unsteady galloping response. *Journal of Wind Engineering and Industrial Aerodynamics, 200*. Article e2020.104141. https://doi.org/10.1016/j.jweia.2020.104141

Scanlan, R. H., & Tomko, J. J. (1971). Airfoil and bridge deck flutter derivatives. *Journal of Engineering Mechanics, ASCE, 97*, pp. 1717-1737.

Siedziako, B., & Øiseth, O. (2018). An enhanced identification procedure to determine the rational functions and aerodynamic derivatives of bridge decks. *Journal of Wind Engineering and Industrial Aerodynamics, 176*. pp. 131 - 141. https://doi.org/10.1016/j.jweia.2018.03.025

Simiu, E., & Yeo, D. (2019). Wind Effects on Structures: Modern Structural Design for Wind, Fourth Edition. John Wiley & Sons Ltd.

SimScale. (December 2, 2016). *Difference Between FEA and CFD*. SimScale CAE Forum. https://www.simscale.com/forum/t/difference-between-fea-and-cfd/53035

SimScale. (August 5, 2018). *What is y+ (yplus)?*. SimScale CAE Forum. https://www.simscale.com/forum/t/what-is-y-yplus/82394

SimScale. (October 23, 2020a). *K-Epsilon*. SimScale Documentation. https://www.simscale.com/docs/simulation-setup/global-settings/k-epsilon/

SimScale. (October 23, 2020b). *K-Omega and K-Omega SST*. SimScale Documentation. https://www.simscale.com/docs/simulation-setup/global-settings/k-omega-sst/

Swedish Institute for Standards [SIS]. (2005). *Eurokod 1: Laster på bärverk - Del 1-4: Allmänna laster - Vindlast* (SS-EN 1991-1-4:2005). Svenska institutet för standarder. https://www-sis-se.eu1.proxy.openathens.net/produkter/byggnadsmaterial-och-byggnader/byggnadsindustrin/tekniska-aspekter/ssen1991142005/

Trafikverket. (2019:3). *Krav Brobyggande* (TDOK 2016:0204).

Transportstyrelsen. (2018). *Transportstyrelsens föreskrifter och allmänna råd om tillämpning av eurokoder* (TSFS 2018:57). https://www.transportstyrelsen.se/TSFS/TSFS%202018_57.pdf

Wawzonek, M. A. (1979). Aeroelastic behavior of square section prisms in uniform flow. [Masters Thesis, The University of British Columbia]. UBC Thesis and Dissertations. https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0100172

Wu, T., Xu, F., & Zhang, M. (2019). Vortex-induced vibration of bridge decks: Describing function-based model. *Journal of Wind Engineering and Industrial Aerodynamics, 176.* Article e2019.104016. https://doi.org/10.1016/j.jweia.2019.104016

Zhiyin, Y. (2015). Large-eddy simulation: Past, present and the future. *Chinese Journal of Aeronautics, 28.* pp. 11 - 24. https://doi.org/10.1016/j.cja.2014.12.007

Appendices

A

Quick Reference Guide for Wind Dynamic Assessment of Bridges

In this quick reference guide, a graphical approach to assess the wind dynamic response of bridge decks is presented. It is based on the method in the British annex to Eurocode, PD 6688-1-4:2009 (BSI, 2009), which in contrast to Eurocode 1:4, SS-EN 1991-1-4:2005 (SIS, 2005), gives a course of action specifically tailored for bridge decks. The purpose of this guide is to provide a quick reference for the wind dynamic response of a bridge, especially useful in preliminary design. A background document containing information on assumptions, choices made and detailed derivations is presented in Appendix B.

The wind dynamic phenomena relevant for medium-span bridges, i.e., bridges with longest spans in the range 50 to 200 metres, are vortex shedding, galloping, torsional flutter and classical flutter. Vortex shedding is the formation of vortices with alternating rotational direction in the wake of a bluff body, creating lift forces with alternating sign. When the frequency of the fluctuations is close to a fundamental frequency of the bridge, vibrations of large amplitude may ensue. Galloping, torsional flutter and classical flutter are pure bending, pure torsion and coupled bending and torsion motion, respectively. They are induced by self-excited forces in the structure and should be avoided as they are divergent phenomena.

The structure of the procedure is presented as a flowchart in Figure A.1. The most critical phenomena depends on parameters such as bridge type and geometry. Therefore, the flowchart guides the user to the relevant checks based on the users inputs. It is important to keep in mind that failure to meet the requirements of this guide is not always equivalent to an unsatisfactory wind dynamic response, especially if the checks are almost satisfied. If one or more checks are not fulfilled, detailed calculations according to the procedure in the British Annex is recommended. Some guidance for such calculations can be found in the background document in Appendix B. Specifically for vortex shedding, estimation of the vibration amplitudes according to Section A.1.5.4 in the British Annex may be sufficient.

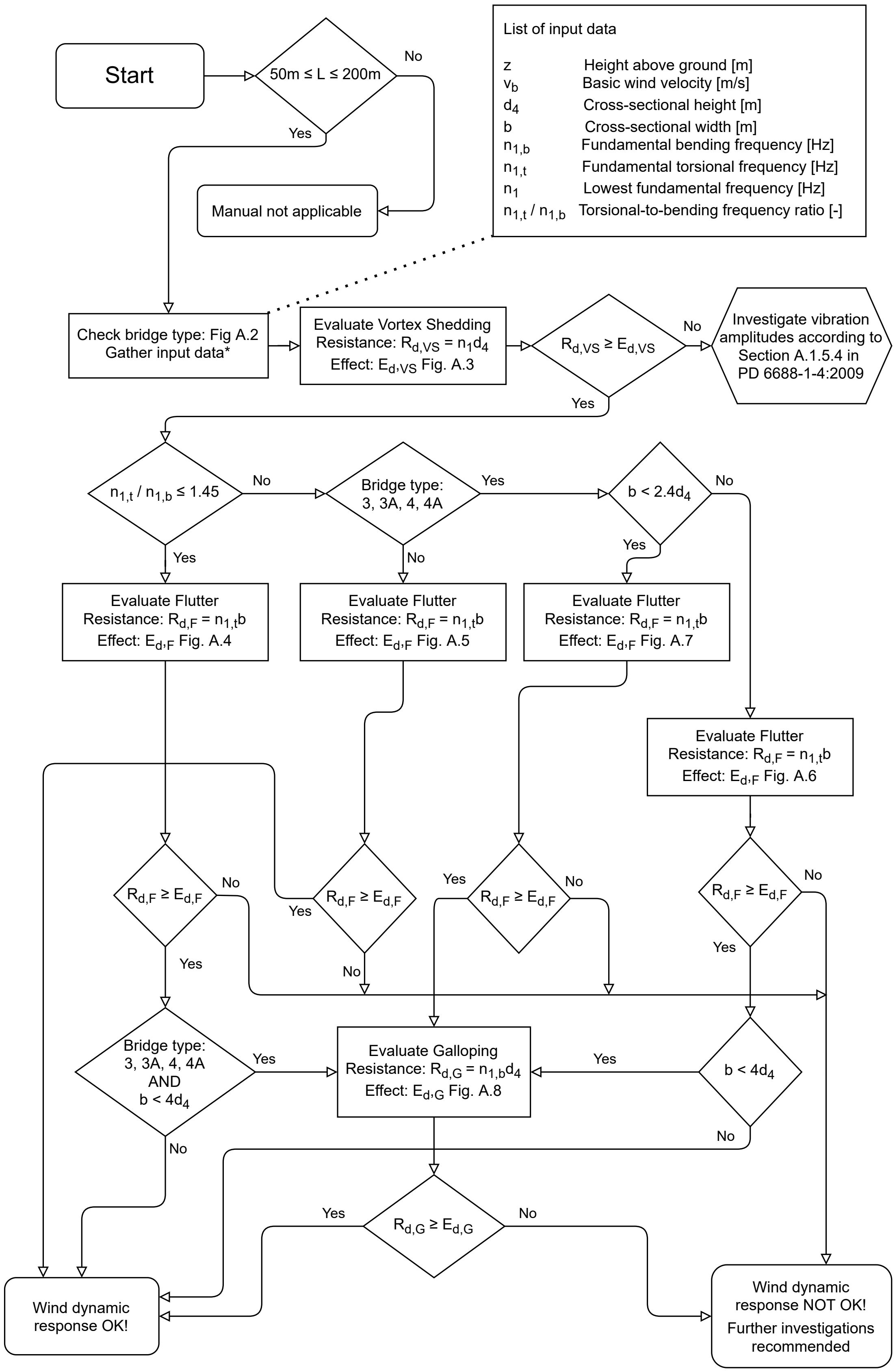
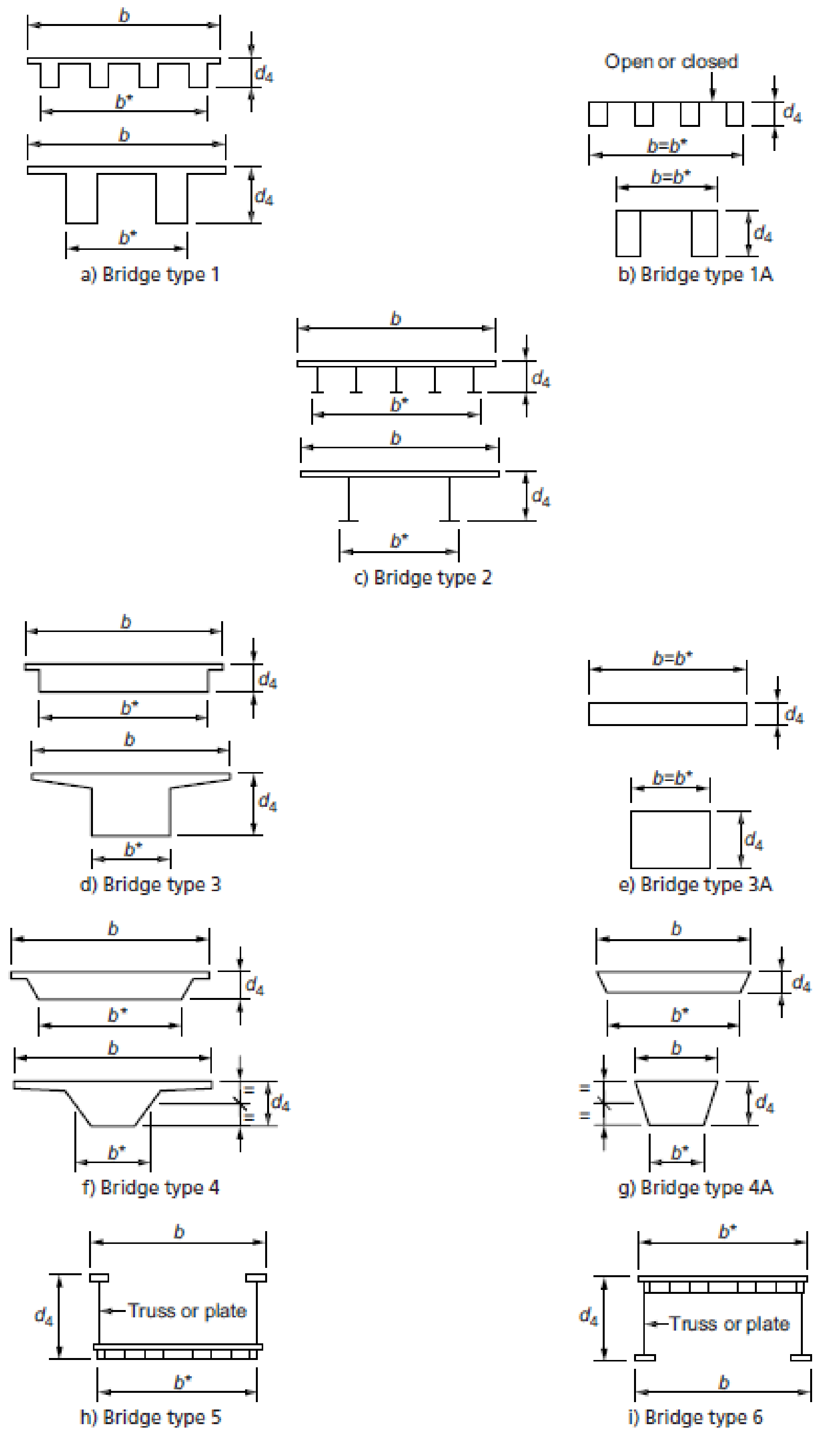


Figure A.1: Flowchart for wind dynamic assessment of bridge decks.



NOTE 1 For truss bridges of type 5 or 6, d_4 is taken as φd_4, where φ is the truss solidity.

NOTE 2 Trusses with $\varphi \geqslant 0.5$ may be treated conservatively as plate girders but taking the depth d_4 as φd_4.

Figure A.2: Bridge types. Reproduced from Figure A.3 in PD 6688-1-4:2009.

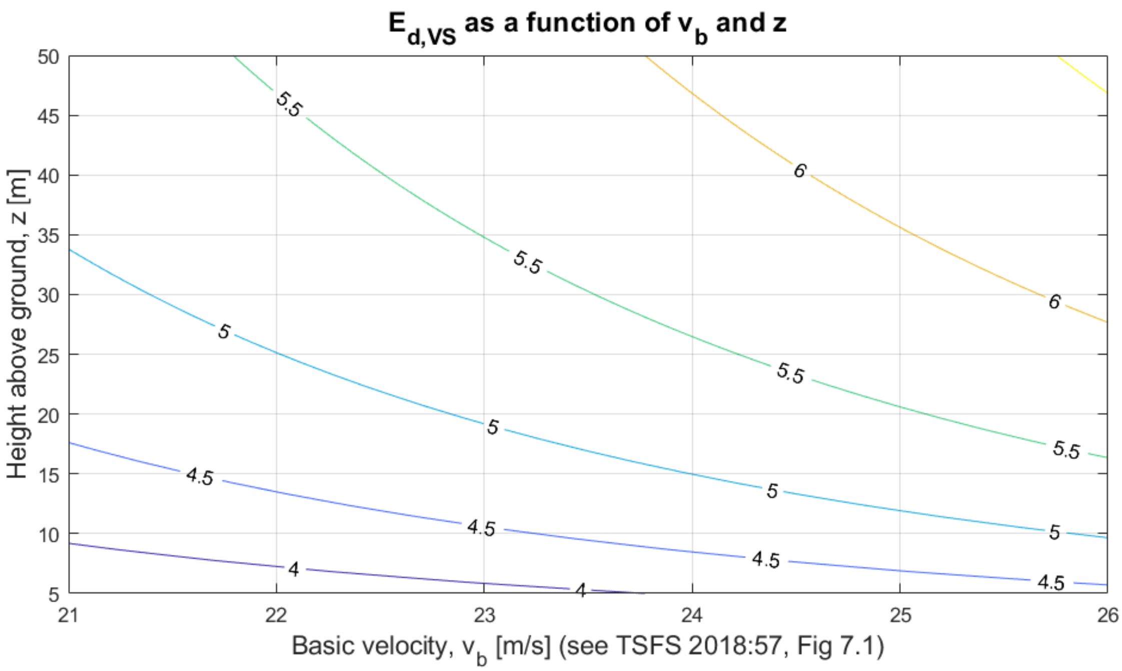


Figure A.3: Design curves for $E_{d,VS}$ to be compared with $R_{d,VS} = n_1 d_4$. Valid for all bridge types. Linear interpolation is allowed.

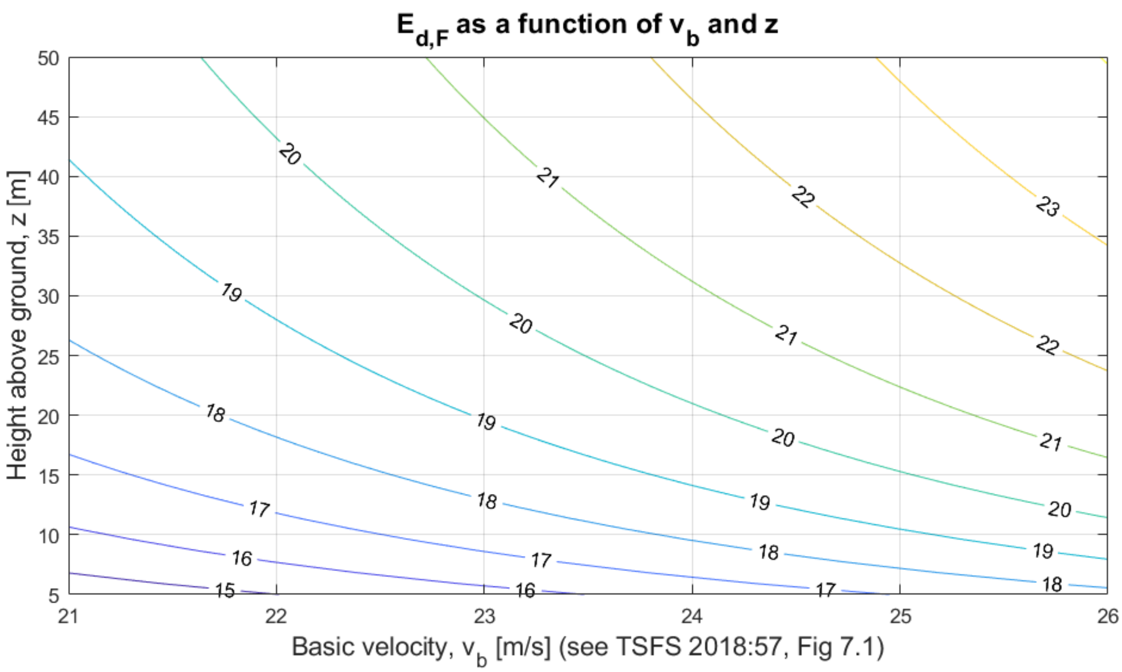


Figure A.4: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t} b$. Valid for bridge types 1, 1A, 2, 5 & 6, with $n_{1,t}/n_{1,b} < 1.1$, conservative for all bridge types with ratios above 1.1, up to 1.45. Linear interpolation is allowed.

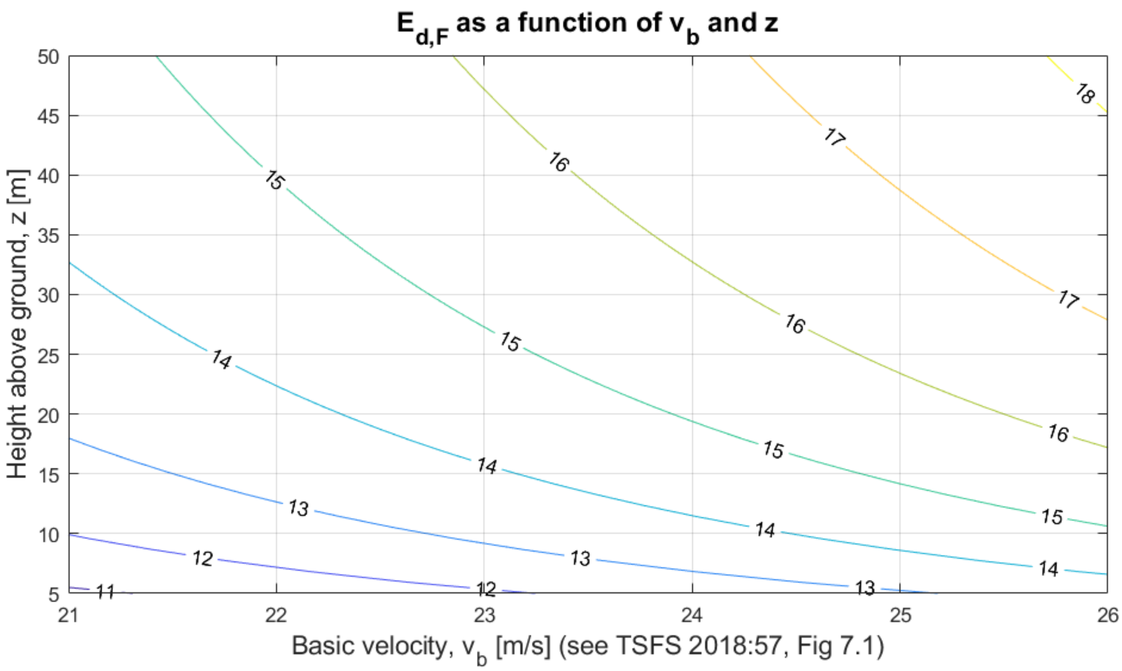


Figure A.5: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 1, 1A, 2, 5 & 6, with $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

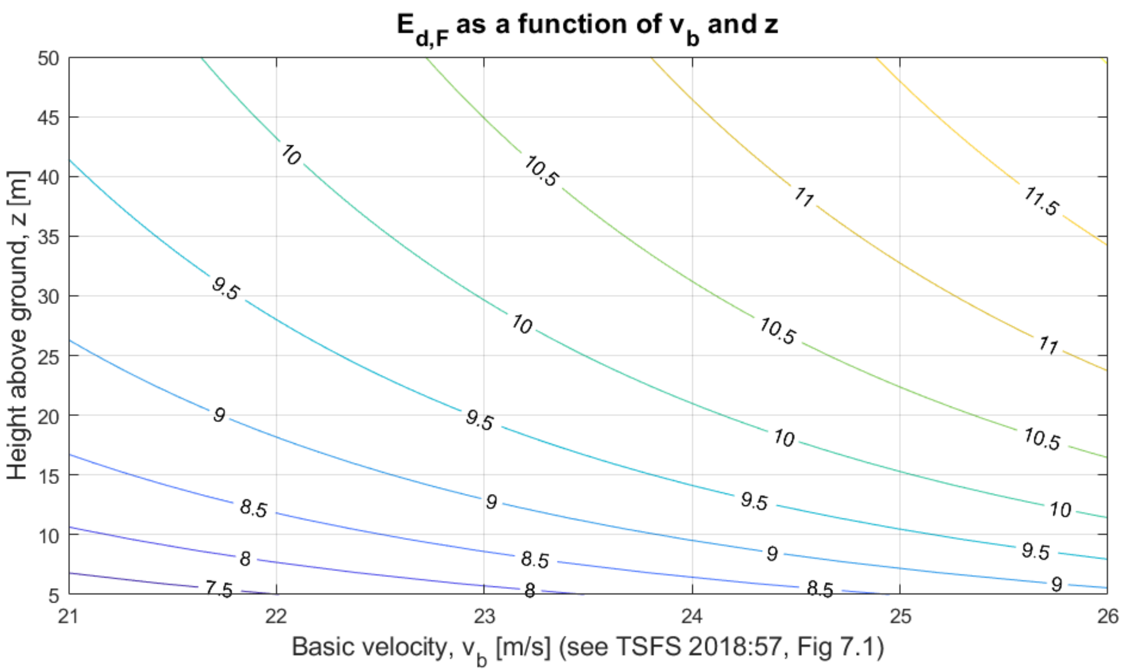


Figure A.6: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 3, 3A, 4, & 4A, with $b \geq 2.4d_4$ AND $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

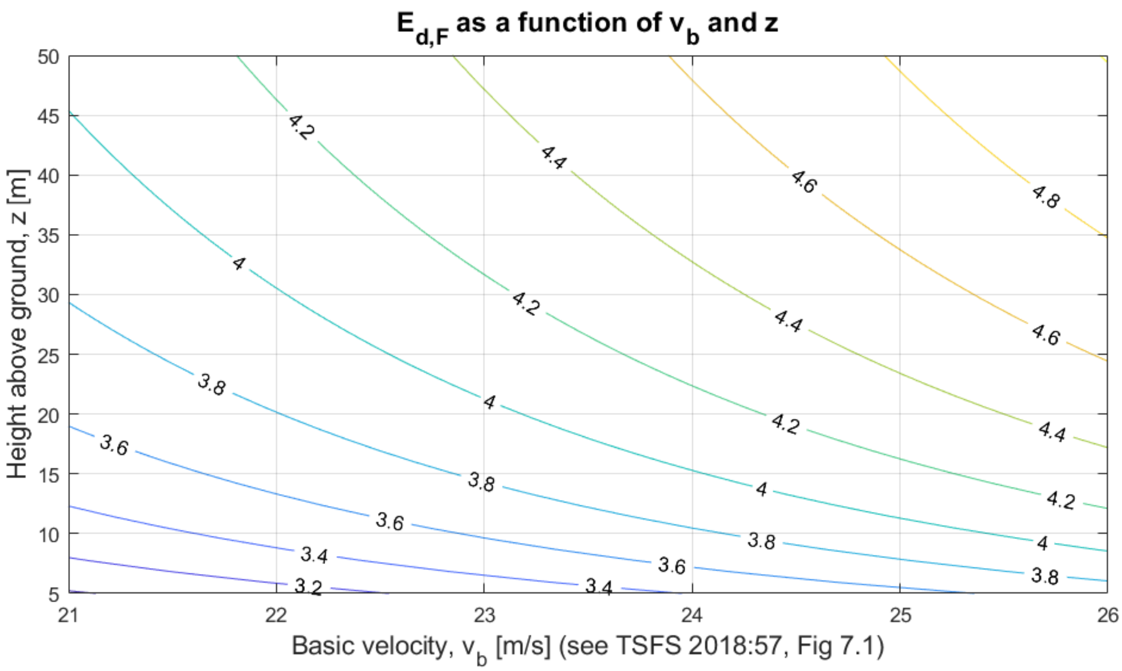


Figure A.7: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 3, 3A, 4, & 4A, with $b < 2.4d_4$ AND $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

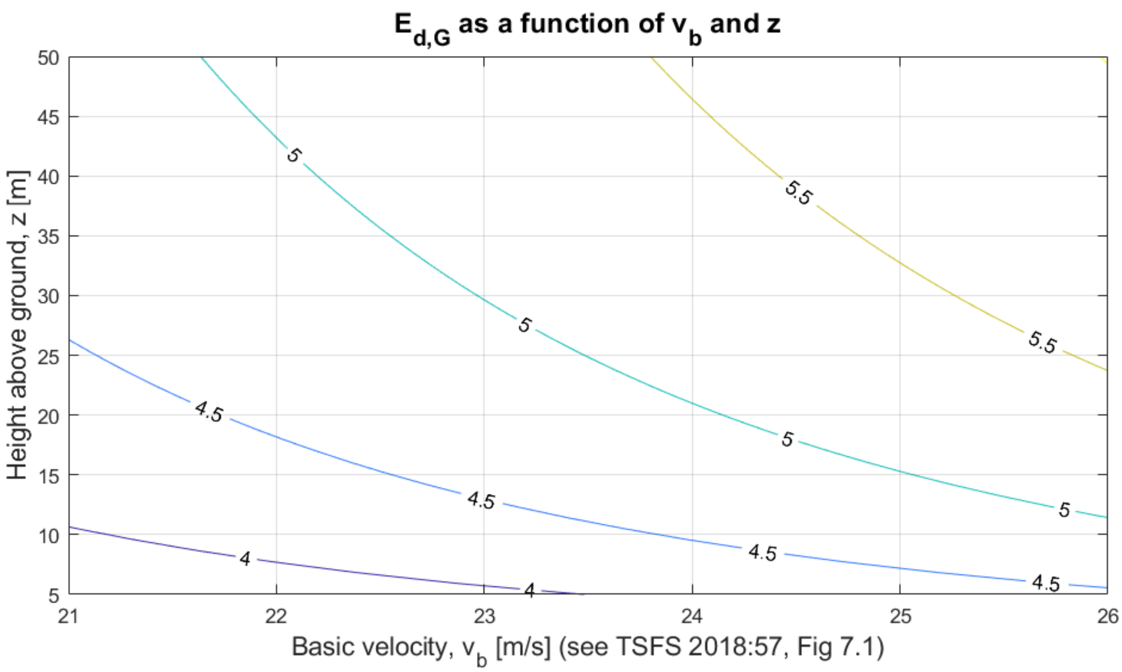


Figure A.8: Design curves for $E_{d,G}$ to be compared with $R_{d,G} = n_{1,b}d_4$, if, and ONLY if, $b < 4d_4$ for bridge types 3, 3A, 4 and 4A. Linear interpolation is allowed.

B

Background Document for Quick Reference Guide

The procedure to assess the wind dynamic response of bridge decks in the British annex to Eurocode, PD 6688-1-4:2009, is constructed such that, for each phenomenon, an onset wind velocity needs to be larger than a criteria velocity.

Nine bridge types, with span lengths in the range of 50 to 200 metres, are distinguished in the procedure. The bridge types, and their variations, are presented in Figure B.1. It is a condensed version of Figure A.3 in PD 6688-1-4:2009.

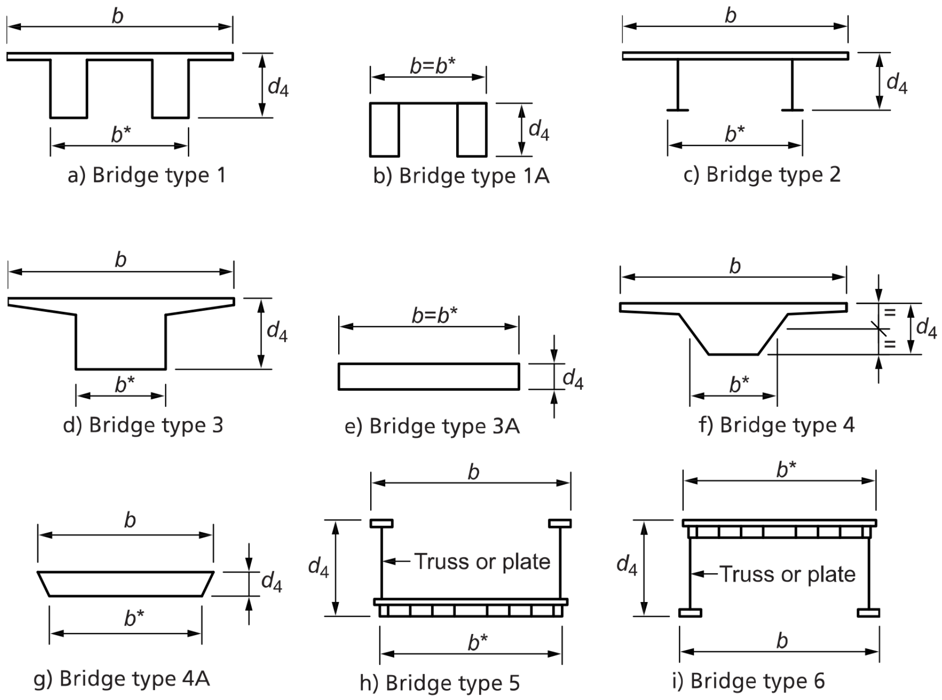


Figure B.1: Condensed figure of bridge types. Reproduced from Figure A.3 in PD 6688-1-4:2009.

The bridge types can be divided into two groups based on their torsional stiffness in relation to bending stiffness. Bridge types 1, 1A, 2, 5 and 6 generally have low torsional stiffness and types 3, 3A, 4 and 4A high torsional stiffness. This distinction is important as the fundamental torsional frequency of bridges with low torsional stiffness roughly coincides with the fundamental bending frequency, influencing the wind dynamic assessment procedure.

B.1 Vortex Shedding

According to Section A.1 in PD 6688-1-4:2009, if the largest-to-smallest crosswind dimension exceeds six, the effect of vortex shedding should be investigated. For a bridge with a span length of 50 metres, a height of 8.3 metres is required to omit the vortex shedding check. This check was likely developed for other structures as this is evidently not feasible for bridges. Hence, all bridges are evaluated for vortex shedding. The effect of vortex shedding can be investigated according to Section A.1.5.4 in PD 6688-1-4:2009. However, the effect of vortex shedding does not need to be investigated if the following requirement is fulfilled

$$v_{crit} > 1.25 v_m \tag{B.1}$$

where v_m is the mean wind velocity. The critical vortex shedding velocity, v_{crit}, is calculated as

$$v_{crit} = \frac{n_1 d_4}{St} \tag{B.2}$$

where n_1 is the lowest fundamental frequency, d_4 is the cross-sectional height of the bridge according to Figure B.1 and St is the Strouhal number defined in Section A.1.3.2 (PD 6688-1-4:2009). For all bridge decks with a ratio $b^*/d_4 \leq 5$, the Strouhal number is defined as the fraction 1/6.5, with b^* defined according to Figure B.1. Ratios above yield lower numbers, depending on bridge type. As it is inversely proportional to the critical velocity, the Strouhal number is locked as 1/6.5, yielding a conservative constraint.

The mean wind velocity, v_m, is calculated according to section 4.3.1 of SS-EN 1991-1-4:2005 as

$$v_m = c_r \cdot c_o \cdot v_b \tag{B.3}$$

The orography factor, c_o, is set as 1.0 as the topography in the vicinity of most bridge decks is negligible. For other cases, special consideration should be taken and the design curves may not be accurate. The basic wind velocity, v_b, is in Sweden determined using Figure 7.1 in TSFS 2018:57. The roughness factor, c_r, is calculated as

$$c_r = k_r \cdot \ln\left(\frac{z}{z_0}\right) \tag{B.4}$$

where z is the height above ground, and z_0 is the roughness length, defined as 0.05 in the caption of Figure 7.1 in TSFS 2018:57. The terrain factor, k_r, is defined as

$$k_r = 0.19 \cdot \left(\frac{z_0}{z_{0,II}}\right)^{0.07} \tag{B.5}$$

where $z_{0,II}$ is the roughness length for category II, defined as 0.05 in Section 4.3.2 of SS-EN 1991-1-4:2005. Given that $z_0 = z_{0,II}$, the terrain factor always equates to 0.19.

Setting the critical vortex shedding velocity equal to the requirement in Equation B.1 and rewriting gives that

$$n_1 d_4 = 1.25 k_r St \cdot \ln\left(\frac{z}{z_0}\right) v_b \tag{B.6}$$

Inserting the presented parameters, the design effect for vortex shedding, $E_{d,VS}$, is introduced as

$$E_{d,VS} := 1.25 \cdot 0.19 \cdot \frac{1}{6.5} \ln\left(\frac{z}{0.05}\right) v_b = 0.037 \ln\left(\frac{z}{0.05}\right) v_b \tag{B.7}$$

to be compared with the design resistance for vortex shedding, $R_{d,VS} := n_1 d_4$. Using the relationship in Equation B.7, the design curves in Figure B.2 is produced in MATLAB.

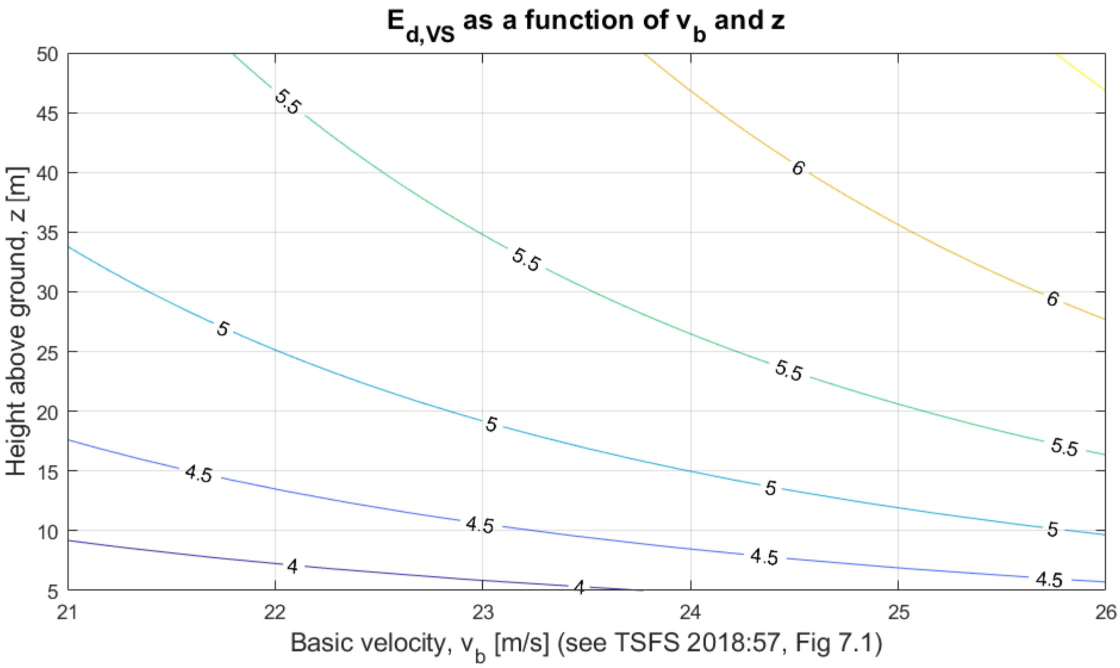


Figure B.2: Design curves for $E_{d,VS}$ to be compared with $R_{d,VS} = n_1 d_4$. Valid for all bridge types. Linear interpolation is allowed.

B.2 Flutter, Galloping and Stall Flutter

According to Sections A.2 and A.4 in PD 6688-1-4:2009, a bridge deck is aeroelastically stable if the following requirements are fulfilled

$$v_g > v_{WO} \tag{B.8a}$$
$$v_f > v_{WO} \tag{B.8b}$$

where v_g and v_f are the onset velocities for galloping or stall (torsional) flutter, and (classical) flutter, respectively, and v_{WO} is the requirement velocity in wind storms.

B.2.1 Wind Storm Velocity

The wind storm velocity, v_{WO}, according to Section A.2.4.2 in PD 6688-1-4:2009, is calculated as

$$v_{WO} = K_{1U} K_{1A} v_m \Big(1 + 2 I_v \sqrt{B^2}\Big) \tag{B.9}$$

where K_{1U} and K_{1A} are national parameters set to 1.1 and 1.25, respectively. The turbulence intensity, I_v, defined in section 4.4 in SS-EN 1991-1-4:2005 as

$$I_v = \frac{k_l}{c_o \cdot \ln\left(\frac{z}{z_0}\right)} \tag{B.10}$$

where z is the height above ground, and z_0 is the roughness length, defined as 0.05 in the caption of Figure 7.1 in TSFS 2018:57. The turbulence factor, k_l, is set as 1.0 as it is the recommended value and no national recommendation for Sweden exists.

The background factor is defined in TSFS 2018:57 as

$$B^2 = exp\Bigg[- 0.05 \cdot \left(\frac{d_4}{z}\right) + \left(1 - \frac{b}{d_4}\right)\left(0.04 + 0.01\left(\frac{d_4}{z}\right)\right)\Bigg] \tag{B.11}$$

where d_4 is the cross-sectional height of the bridge deck according to Figure B.1, b is the total width of the bridge and z is the height above ground. In Equation B.11, d_4 is set as 2 metres as it is a reasonable approximation for bridges with spans exceeding 50 metres. The choice has an almost negligible influence on the wind storm velocity, where an increase of 0.5 metres affects it by 0.1 percent. With increasing width-to-height ratios, b/d_4, the background factor B decreases. Hence, the ratio is set to the markedly low value of 2.5 to yield conservative estimates. Note that the difference in wind storm velocity between a more realistic ratio of 5 and 2.5 is approximately one percent.

So, the wind storm velocity, v_{WO}, can be expressed as a function of height above ground, z, and basic wind velocity, v_b, as

$$\begin{aligned} v_{WO}(v_b, z) = 0.261 \cdot \ln\left(\frac{z}{0.05}\right) \cdot v_b \Bigg(1 + 2\frac{1}{\ln\left(\frac{z}{0.05}\right)} \cdot \\ \sqrt{exp\Bigg[- 0.05\left(\frac{2}{z}\right) + \Big(1 - 2.5\Big)\left(0.04 + 0.01\left(\frac{2}{z}\right)\right)\Bigg]}\Bigg) \end{aligned} \tag{B.12}$$

B.2.2 Flutter

In Section A.4.4 (PD 6688-1-4:2009), the onset velocity of (classical) flutter for bridge decks is determined as

$$v_f = v_{Rf} n_{1,t} b \tag{B.13}$$

where $n_{1,t}$ is the fundamental torsional frequency and b is the width of the bridge deck. The non-dimensional reduced flutter velocity, v_{Rf}, is defined as

$$v_{Rf} = max\left(1.8\left[1 - 1.1\left(\frac{n_{1,b}}{n_{1,t}}\right)^2\right]^{1/2}\left(\frac{mr}{\rho b^3}\right)^{1/2}, 2.5\right) \tag{B.14}$$

where $n_{1,b}$ is the fundamental bending frequency, ρ is the air density defined as 1.25 kg/m^3, m is the mass per unit length and r is the radius of gyration of the cross-section. Note that v_{Rf} cannot be less than 2.5.

Studying Equation B.14, it is beneficial to break it down into smaller components. Aside from the constant 1.8, two decisive components remain:

1) $$\left[1 - 1.1\left(\frac{n_{1,b}}{n_{1,t}}\right)^2\right]^{1/2}$$

2) $$\left(\frac{mr}{\rho b^3}\right)^{1/2}$$

For torsional-to-bending frequency ratios[1], $n_{1,t}/n_{1,b}$, around 1, the first component becomes the square root of a negative quantity, i.e. an imaginary number. Hence, for bridge types 1, 1A, 2, 5, and 6, v_{Rf} is equal to 2.5 regardless of the value of the second component, if the fundamental frequencies coincide. The equation for the onset wind velocity of flutter for those bridge types becomes

$$v_f = 2.5 n_{1,t} b \tag{B.15}$$

For larger ratios $n_{1,t}/n_{1,b}$, typical for bridges of types 3, 3A, 4, and 4A, the first component grows quickly, thus enabling v_{Rf}'s exceeding 2.5. Then, the second component is of interest. As it is beneficial to have large v_{Rf}'s, a small second component is conservative. This is produced by lightweight bridge decks with low radii of gyration, giving values in the range of 8 to 5. Choosing to set the second component to 4 gives that, for ratios $n_{1,t}/n_{1,b} > 1.45$, v_{Rf} is larger than 5. Hence, the following expression can be derived which will be useful for comparison with galloping later.

$$v_f \geq 5 n_{1,t} b \tag{B.16}$$

[1]For clarity, this study uses torsional-to-bending frequency ratios rather than bending-to-torsion frequency ratios used in Equation B.14. This choice is made as the fundamental torsional frequency is larger than the fundamental bending frequency for the lion's share of the relevant bridges, giving ratios larger than one.

B.2.3 Bridge Types 1, 1A, 2, 5 and 6

Aside from classical flutter, PD 6688-1-4:2009 stipulate that galloping and stall flutter needs to be checked. The procedure separates galloping and stall flutter, i.e., vertical and torsional motion. For bridges of type 1, 1A, 2, 5 and 6 only torsional motion is checked, and the equation to evaluate onset wind velocity for stall flutter, v_g, is defined in section A.2.4.1 of PD 6688-1-4:2009, as

$$v_g = 3.3 n_{1,t} b \tag{B.17}$$

where $n_{1,t}$ is the fundamental torsional frequency and b is the width of the bridge deck. Comparing to Equation B.13, it is observed that both equations are dependent on the product $n_{1,t}b$ multiplied by a constant. Given the limits of v_{Rf} (see Equation B.14) in relation to the torsional-to-bending frequency ratio $n_{1,t}/n_{1,b}$, some conclusions can be drawn comparing the two phenomena. For ratios $n_{1,t}/n_{1,b} < 1.1$, the constant v_{Rf} is 2.5, and therefore, classical flutter always will be the determining phenomena. Inserting this into B.8 and rewriting yields that

$$n_{1,t} b = \frac{1}{2.5} v_{WO}(v_b, z) \tag{B.18}$$

with $v_{WO}(v_b, z)$ from Equation B.12. The design effect for galloping, stall flutter and flutter, $E_{d,F}$, is introduced as

$$E_{d,F}(v_b, z) := \frac{1}{2.5} v_{WO}(v_b, z) = 0.4 v_{WO}(v_b, z) \tag{B.19}$$

to be compared with the design resistance, $R_{d,F} := n_{1,t}b$. By plotting the effect $E_{d,F}$ in MATLAB, the design curves in Figure B.3 are produced.

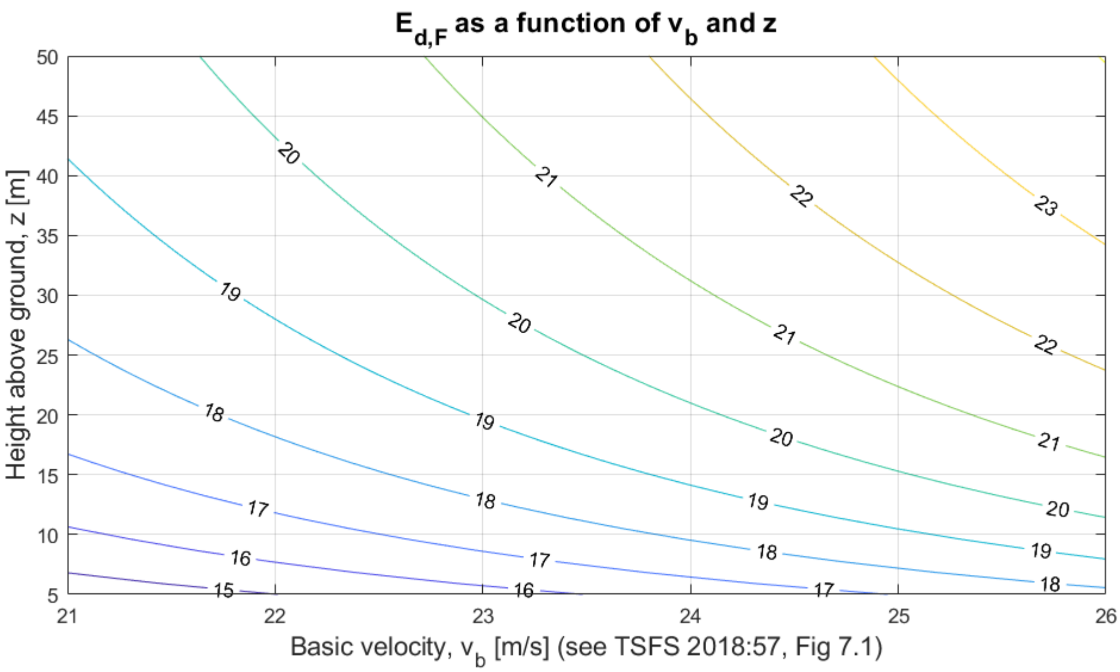


Figure B.3: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 1, 1A, 2, 5 & 6, with $n_{1,t}/n_{1,b} < 1.1$, conservative for ratios above 1.1, up to 1.45. Linear interpolation is allowed.

Bridges of types 1, 1A, 2, 5 and 6 with connected main girders will have increased torsional stiffness. Thus, the assumption of coinciding vertical and torsional fundamental frequencies does not hold. Ratios $1.1 < n_{1,t}/n_{1,b} < 1.45$ lie in a transition region where the dependency on locked parameters impede the ability to give generally applicable design recommendations. For such ratios, Figure B.3 can be used, but results are conservative.

For torsional-to-bending frequency ratios $n_{1,t}/n_{1,b} > 1.45$, v_{Rf} of Equation B.14 is larger than 5 and therefore stall flutter, according to Equation B.17, will be the decisive phenomenon. The following design effect for galloping, stall flutter and flutter, $E_{d,F}$, is derived

$$E_{d,F}(v_b, z) := \frac{1}{3.3} v_{WO}(v_b, z) = 0.303 v_{WO}(v_b, z) \tag{B.20}$$

to be compared with the design resistance, $R_{d,F} := n_{1,t}b$. By plotting the effect $E_{d,F}(v_b, z)$ in MATLAB, the design curves in Figure B.4 are produced.

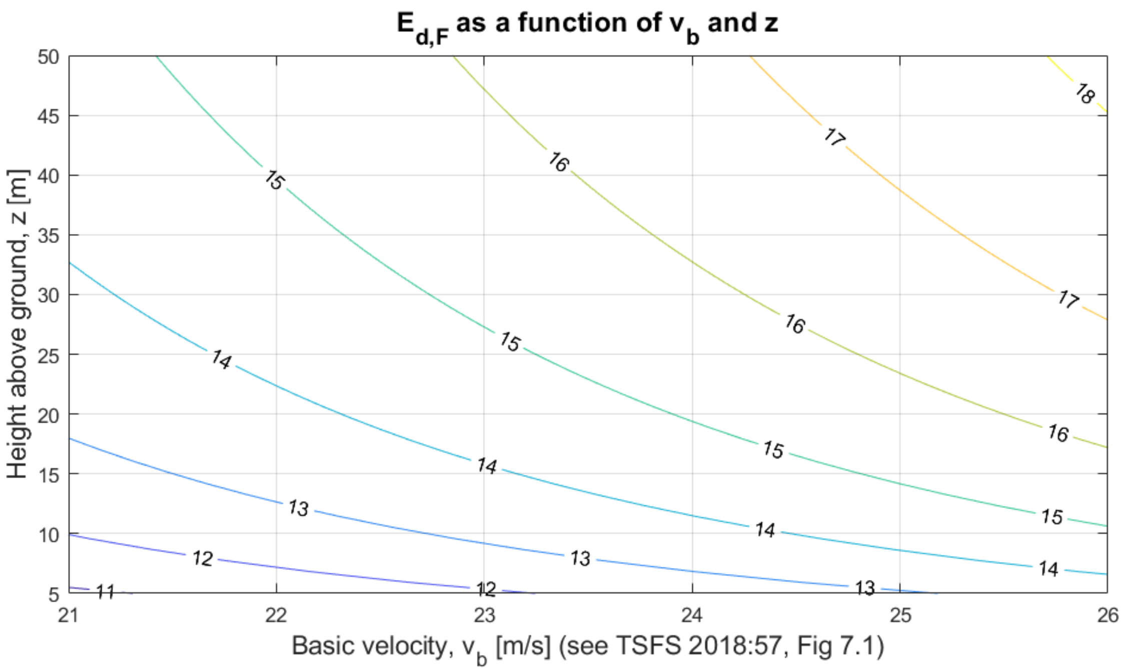


Figure B.4: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 1, 1A, 2, 5 & 6, with $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

B.2.4 Bridge Types 3, 3A, 4 and 4A

As bridges of types 3, 3A, 4 and 4A are torsionally stiff, it generally holds that $n_{1,t}/n_{1,b} > 1.45$ [2], for which v_{Rf} is at least 5 (See Equation B.16). Subsequently, stall flutter is more critical than classic flutter. The equation to evaluate the onset velocity for torsional motion is

[2]For frequency ratios $n_{1,t}/n_{1,b} < 1.45$, a markedly conservative estimate of the design effect for stall flutter and flutter, $E_{d,F}$, can be gathered from Figure B.3.

$$v_g = 5n_{1,t}b \tag{B.21}$$

where $n_{1,t}$ is the fundamental torsional frequency and b is the width of the bridge deck. However, for ratios $b < 4d_4$, the onset velocity equation varies dependent on width-to-height ratio, and is determined as the lesser of

$$v_g = 5n_{1,t}b \tag{B.22a}$$

$$v_g = 12n_{1,t}d_4 \tag{B.22b}$$

By studying the equations, the need to evaluate them separately can be eliminated by further dividing the width-to-height ratio range as the velocities are equal when $b = \frac{5}{12}d_4$. Therefore, for ratios $2.4d_4 \leq b < 4d_4$, the design effect for stall flutter and flutter, $E_{d,F}$, is defined as

$$E_{d,F}(v_b, z) := \frac{1}{5}v_{WO}(v_b, z) = 0.2v_{WO}(v_b, z) \tag{B.23}$$

to be compared with the design resistance, $R_{d,F} := n_{1,t}b$. Furthermore, as the effect is the same for ratios $b > 4d_4$, the relationship holds true for all ratios $b \geq 2.4d_4$. By plotting the effect, $E_{d,F}(v_b, z)$, in MATLAB, the design curves in Figure B.5 are produced.

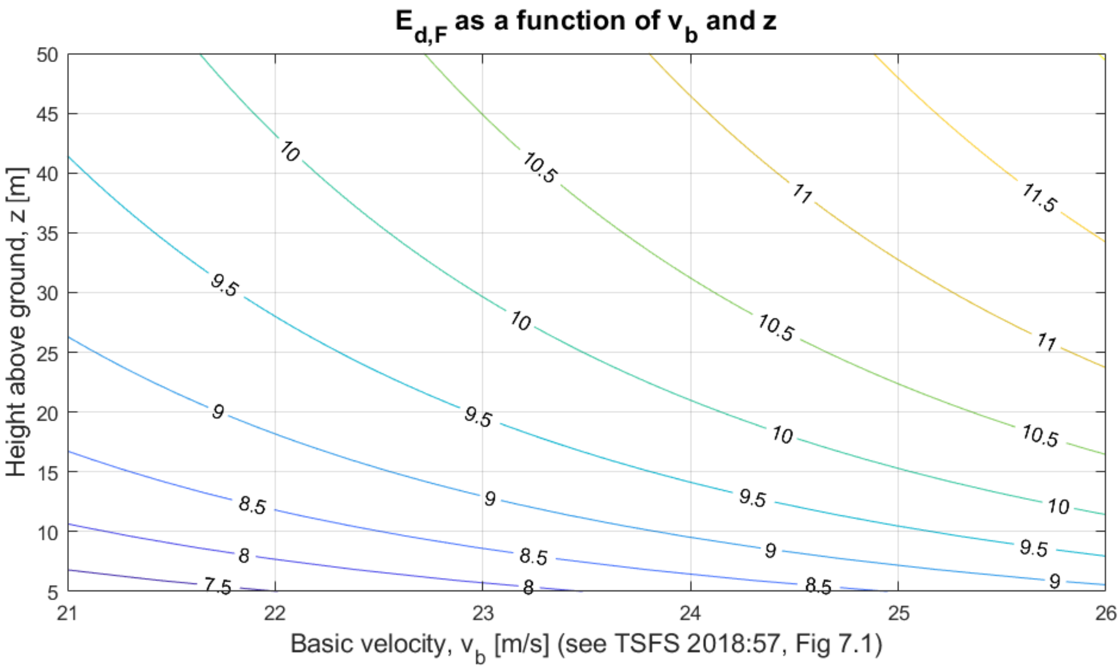


Figure B.5: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 3, 3A, 4, & 4A, with $b \geq 2.4d_4$ AND $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

For ratios $b < 2.4d_4$, the design effect for stall flutter and flutter, $E_{d,F}$, is determined as

$$E_{d,F}(v_b, z) := \frac{1}{12} v_{WO}(v_b, z) = 0.083 v_{WO}(v_b, z) \tag{B.24}$$

to be compared with the design resistance, $R_{d,F} := n_{1,t}d_4$. By plotting the effect $E_{d,F}(v_b, z)$ in MATLAB, the design curves in Figure B.6 are produced.

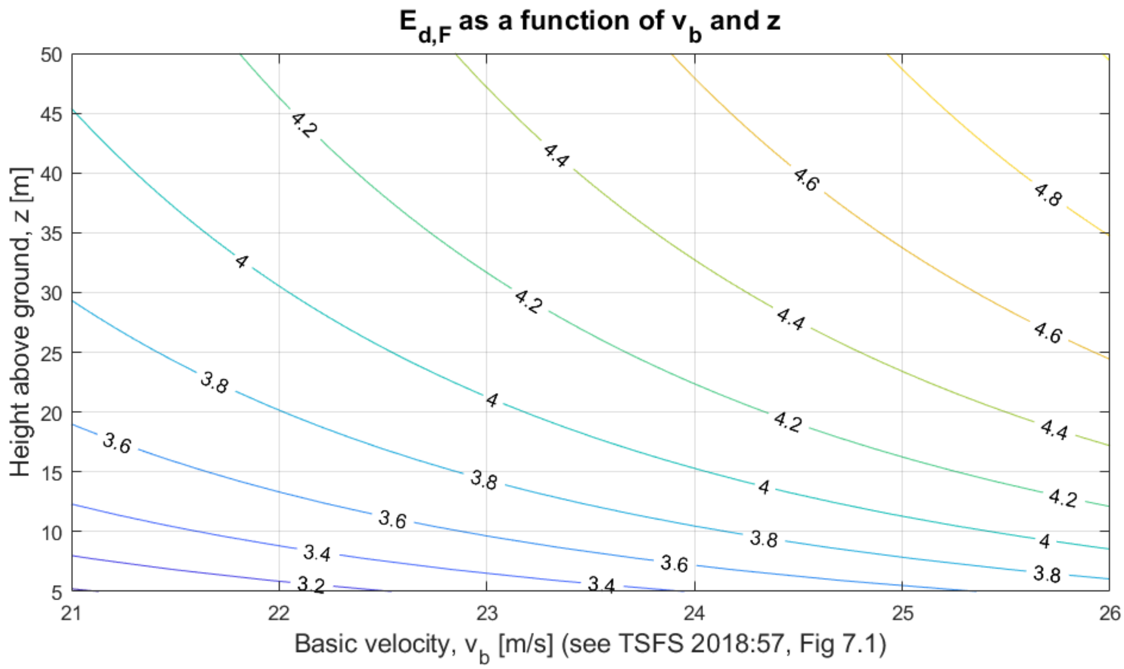


Figure B.6: Design curves for $E_{d,F}$ to be compared with $R_{d,F} = n_{1,t}b$. Valid for bridge types 3, 3A, 4, & 4A, with $b < 2.4d_4$ AND $n_{1,t}/n_{1,b} > 1.45$. Linear interpolation is allowed.

For bridges of type 3, 3A, 4 and 4A, PD 6688-1-4:2009 stipulate that, aside from classical flutter, both vertical and torsional motion needs to be checked. Note that vertical motion only needs to be checked for width-to-height ratios $b < 4d_4$. The vertical motion, i.e., galloping, is evaluated with

$$v_g = v_{Rg} n_{1,b} d_4 \tag{B.25}$$

where d_4 is the cross-sectional height of the bridge and $n_{1,b}$ is the fundamental bending frequency. The non-dimensional reduced galloping velocity, v_{Rg}, is defined as

$$v_{Rg} = \frac{C_g(m\delta_s)}{\rho d_4^2} \tag{B.26}$$

where m is the mass per unit length, ρ is the density of air and δ_s is the logarithmic decrement of structural damping, approximately determined according to Section F.5 in SS-EN 1991-1-4:2005. For pure concrete and composite concrete-steel bridges, δ_s is 4 %, and for pure steel bridges it is 2 %. C_g is a factor, defined as either 1 or 2, based on bridge type and geometry. The lowest v_{Rg}'s are generated by lightweight steel bridges, giving values in the range of 15 to 30. Hence, v_{Rg} is set to 10, giving a conservative relationship

$$v_g = 10 n_{1,b} d_4 \tag{B.27}$$

The requirement to exceed is the wind storm velocity, giving the design effect for galloping, $E_{d,G}$, as

$$E_{d,G}(v_b, z) := \frac{1}{10} v_{WO}(v_b, z) = 0.1 v_{WO}(v_b, z) \tag{B.28}$$

to be compared with the design resistance, $R_{d,G} := n_{1,b} d_4$. By plotting the effect $E_{d,G}(v_b, z)$ in MATLAB, the design curves in Figure B.7 are produced.

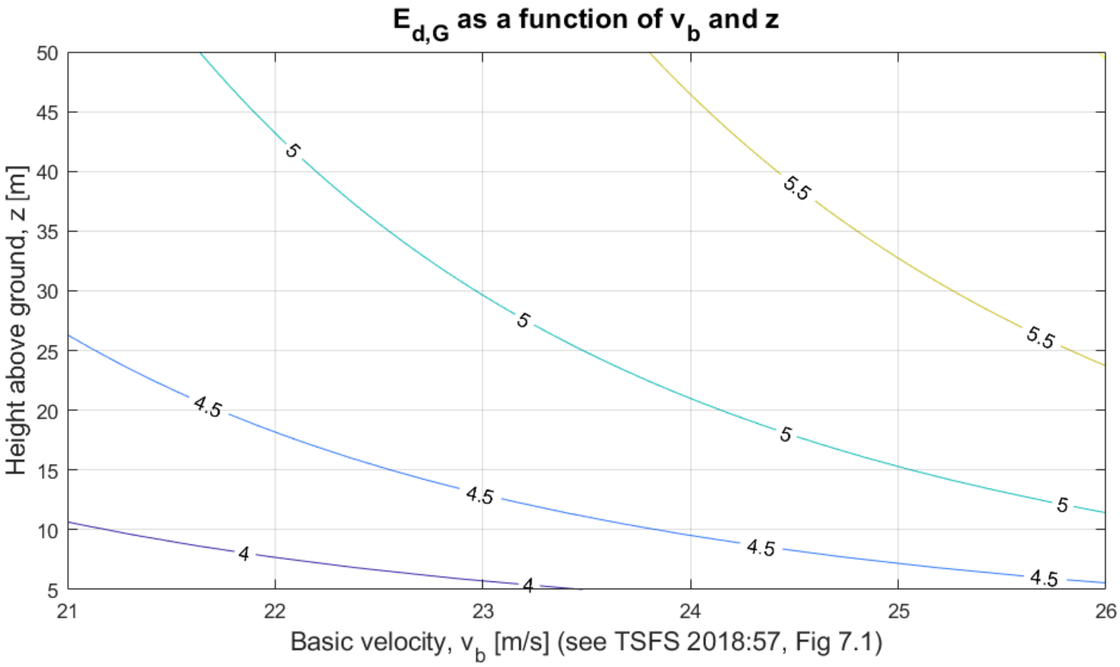


Figure B.7: Design curves for $E_{d,G}$ to be compared with $R_{d,G} = n_{1,b} d_4$, if, and ONLY if, $b < 4d_4$ for bridge types 3, 3A, 4 and 4A. Linear interpolation is allowed.

C

Verification of Strouhal Numbers from OpenFOAM on Rectangles

2D simulations of vortex shedding on a set of rectangles are performed in order to verify the accuracy of vortex shedding simulations with CFD. The wind flow around rectangles of various width-to-height ratios are simulated and their respective Strouhal numbers are extracted from the gathered data. The Strouhal number from the simulation is compared to tabulated data from Section 7.6 of SS-EN 1991-1-4:2005.

The methodology of the simulations is to run a steady-state simulation with the simpleFoam solver for 2000 iteration steps, using the RANS turbulence model K-omega SST. The flow field of the steady-state solution is mapped using the OpenFOAM utility mapFields and used as initial conditions for the transient simulation using the pimpleFoam solver, employing the hybrid LES-RANS turbulence model K-Omega SST DES. The transient simulation is run for approximately 100 seconds, with a maximum time step length of 0.01 seconds and maximum Courant number set to 5. The kinematic viscosity of air is set to 15e-6 m^2/s.

C.1 Geometry, Boundary Conditions and Mesh

In Figure C.1, the geometrical notations of the studied rectangles are illustrated.

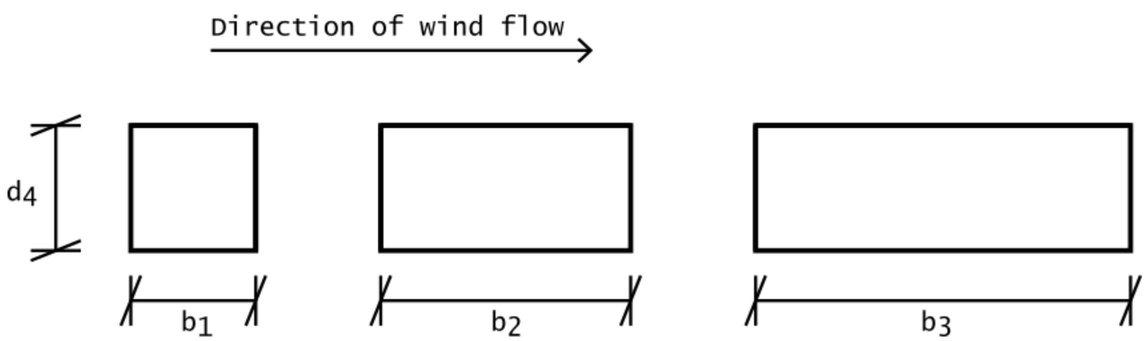


Figure C.1: Geometrical notations of the studied rectangles.

In order to make the verification relevant for future application on bridge cross-sections, the height, d_4 is set to 2 meters as bridges with longest spans between 50 and 200 metres have similar heights. The widths, b_1, b_2 and b_3 are set to 2, 4 and 10 meters respectively, yielding rectangles with width-to-height ratios of 1, 2 and 5. The wind tunnel dimensions used in the simulations are based on the cross-sectional dimensions of the respective rectangles, as presented in Figure C.2.

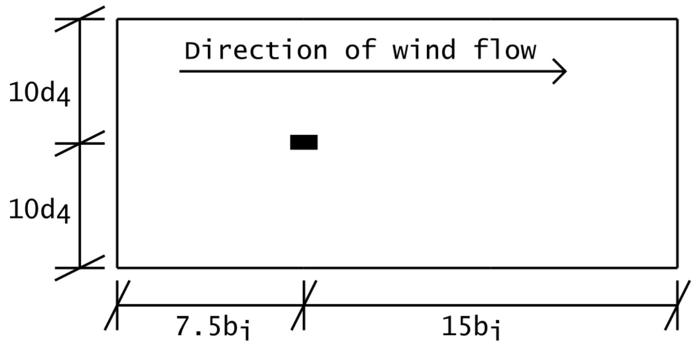


Figure C.2: Dimensions of wind tunnel.

As seen in Figure C.2, the dimensions in the direction of the wind flow are determined by the width, b_i, of the studied rectangle and the height is determined by the constant height, d_4. Therefore, the total height of the wind tunnel is 40 metres for all rectangles, while the total length is 45, 90 and 225 metres, respectively. The boundary conditions are identical for all cases, and are compiled in Table C.1.

Table C.1: Boundary conditions of the wind tunnels.

Boundary	**Parameter and Value**
Inlet (air velocity)	$v = 10$ m/s
Outlet (pressure)	$p = 0$ Pa
Slip (air velocity)	$v_n = 0$ m/s (normal), and $v_t \neq 0$ m/s (tangential)
Wall (air velocity)	$v = 0$ m/s (Wall functions are used for k, ω, and $\tilde{\nu}$)

In Figure C.3, the positions of the boundary conditions given in Table C.1 are presented. As this is a 2D simulation, the front and back faces are set as empty to indicate that the direction is not solved.

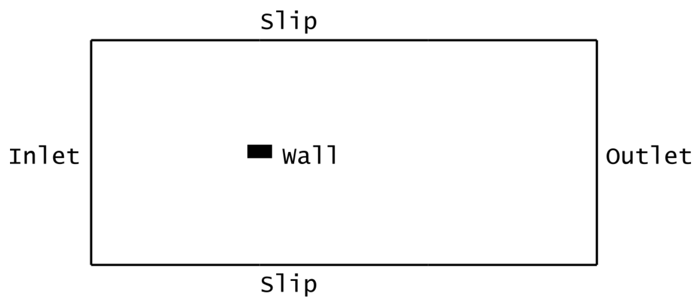
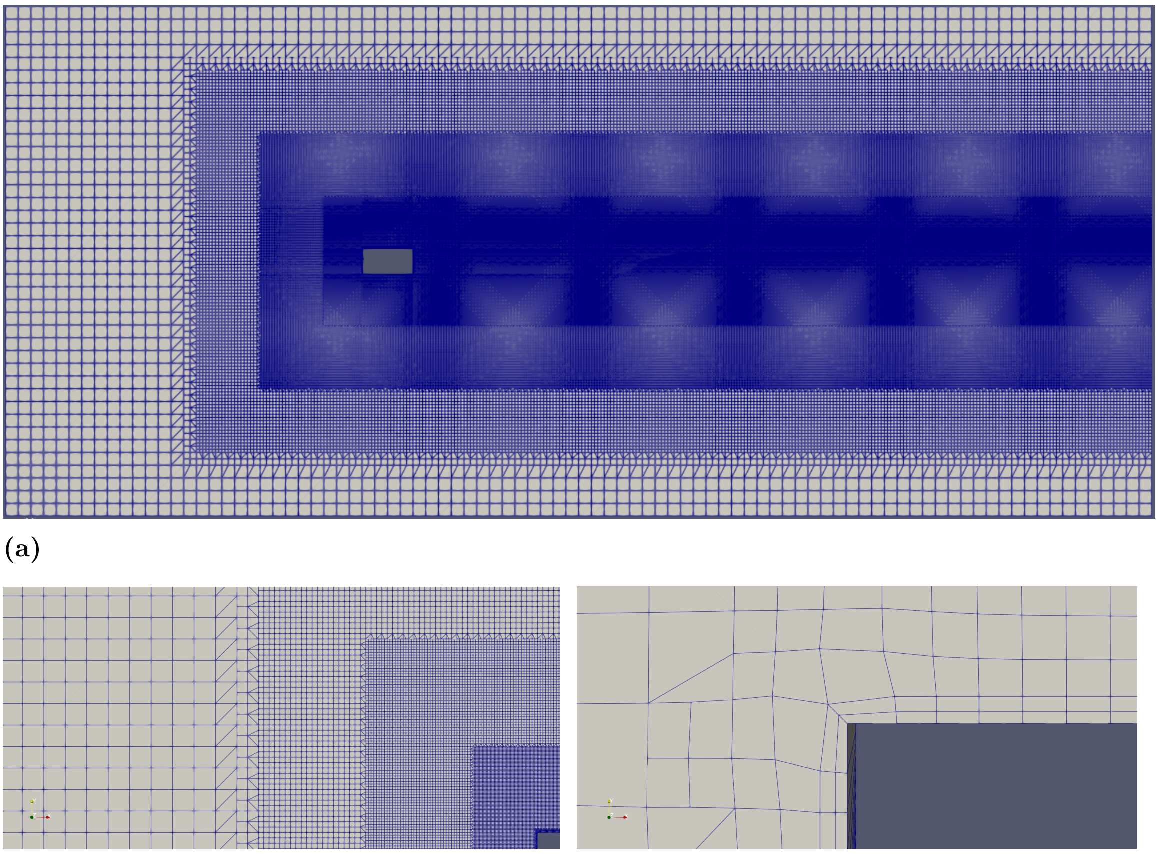


Figure C.3: Boundary conditions of the wind tunnels.

In Figure C.4, the mesh for the rectangle with a width-to-height ratio of 2 is presented. The mesh is very similar for the other rectangles with small adjustments to fit the variations in dimensions.



(a)

(b)

(c)

Figure C.4: Snapshots of the mesh of the wind tunnel, illustrating the refinement arrangement in the regions of the mesh. The mesh is generated using blockMesh and snappyHexMesh in OpenFOAM.

As seen in Figure C.4 (a), the mesh is refined in boxes with increasing refinement closer to the rectangle. The wake is highly refined as good resolution of this area is imperative to accurately simulate vortex shedding. In (b), the gradual refinement of the mesh is highlighted. The size of the largest elements furthest from the rectangle are 1.0 x 1.0 metres. The mesh is then refined in three stages, with a refinement level of 2, 3 and 4 respectively. The element sizes in the most refined area, excluding the boundary layer, are 2^4, i.e., 16, times smaller than the largest, yielding elements of sizes 0.0625 x 0.0625 metres. In (c), the boundary layer elements are visible, where the number of elements is chosen as 2. The height of element closest to rectangle surface is 0.004 metres, normal to the surface.

C.2 Initial Conditions

In Table C.2, the turbulence conditions are presented. The turbulence intensity is chosen as 15 % to represent typical wind flow. The characteristic turbulence length is chosen equal to the height of the rectangles as this is approximately the size of the largest vortices.

Table C.2: Turbulence Conditions.

Condition	**Parameter and Value**
Turbulence intensity	$I = 15\ \%$
Inlet turbulence energy	$k = \frac{3}{2}(v \cdot I)^2 = \frac{3}{2}(10 \cdot 0.15)^2 = 3.375$ J/kg
Characteristic turbulence length	$l = d_4 = 2$ m
Specific dissipation rate	$\omega = \frac{\sqrt{k}}{l} = \frac{\sqrt{3.375}}{2} = 0.91856$ 1/s

In Table C.3, the y^+ values for the final iteration step of the steady state RANS simulations used as initial conditions in the transient RANS-LES hybrid simulation, for each respective rectangle are presented.

Table C.3: Minimum, maximum and average y^+ values from the results of the steady state RANS simulations.

Width-to-height ratio	min	max	average
$b/d_4 = 1$	31.8	111.2	48.3
$b/d_4 = 2$	32.1	112.2	47.7
$b/d_4 = 5$	10.3	105.6	44.5

A criteria of good mesh resolution at the surface of an object is to have y^+ values between 30 and 300. The averages of all rectangles satisfy this, but the min value for the rectangle with a width-to-height ratio of 5 does not. This occurs as the velocity behind the rectangle, just below and above the corners, is very low. However, the amount of cells affected are few and therefore the mesh quality is deemed sufficient.

C.3 Results

The lift coefficient is extracted for all time steps and plotted in MATLAB. Furthermore, a Fast Fourier transform of the lift coefficient is performed and the resulting frequency spectra is converted to corresponding Strouhal numbers with the formula:

$$St = \frac{v}{d_4 \cdot f_{FFT}}$$

In subsequent sections, the simulation results of the studied rectangles are presented.

C.3.1 Rectangle with $b/d_4 = 1$

In Figure C.5, the velocity flow field of the rectangle is visualized with Paraview.

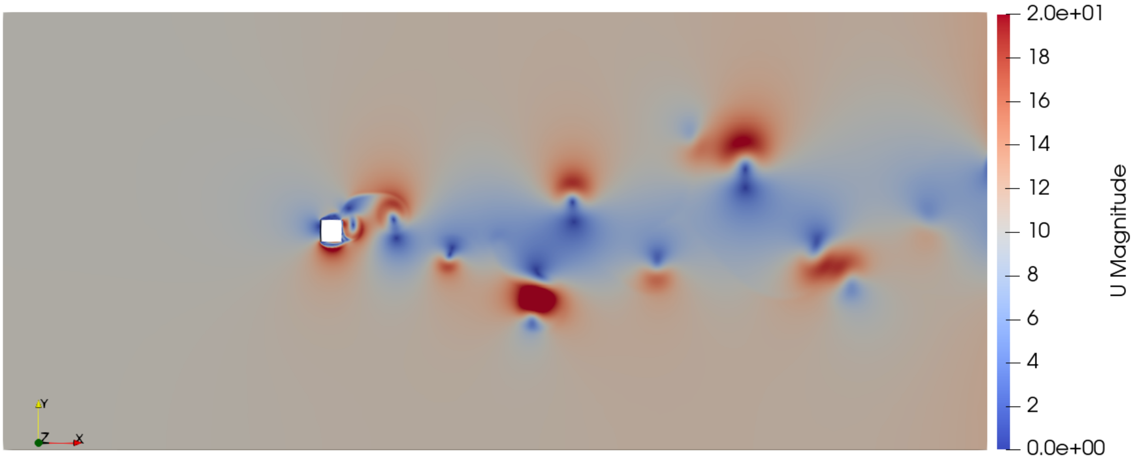


Figure C.5: Velocity flow field in metres per second around rectangle with width-to-height ratio of 1. Snapshot corresponds to fully developed vortex shedding conditions, captured at the simulation time 60.0 seconds.

In Figure C.6, the upper graph depicts lift coefficient as a function of time for the rectangle. The lower graph depicts Strouhal numbers, determined through an FFT of the lift coefficient from the times 20 to 99 seconds.

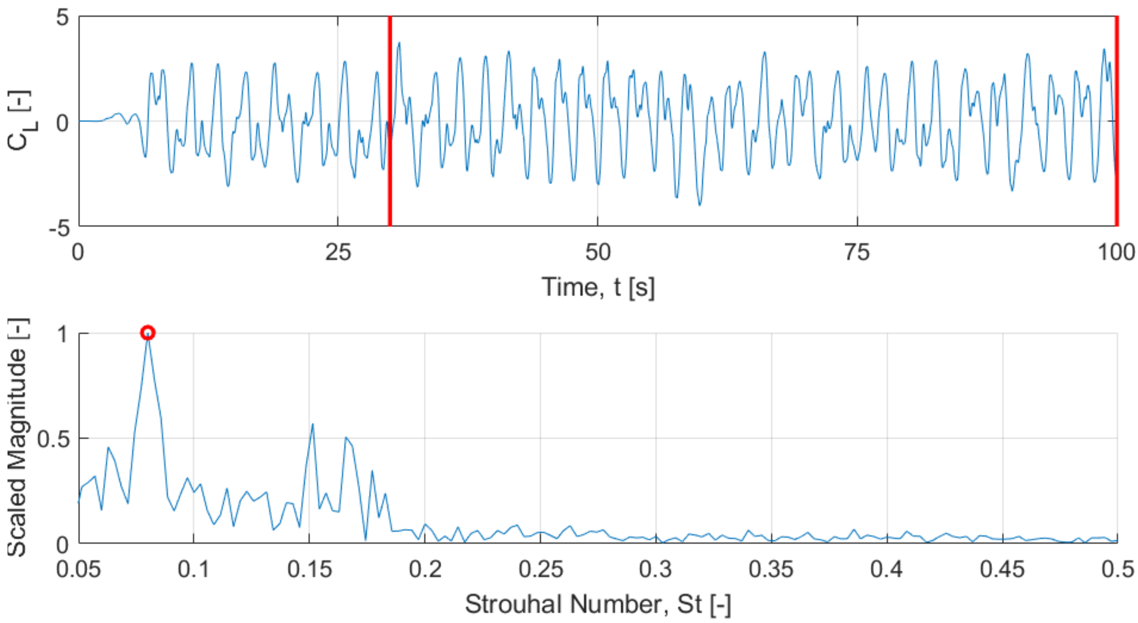


Figure C.6: Lift coefficient as a function of time and Strouhal numbers corresponding to the frequencies of an FFT on the lift coefficient between times 20 and 99 seconds, for a rectangle with width-to-height ratio of 1.

As seen in Figure C.6, the Strouhal number is extracted at the tallest peak, corresponding to 0.084. The cluster further right are neglected as the corresponding vibration amplitude will be low.

C.3.2 Rectangle with $b/d_4 = 2$

In Figure C.7, the velocity flow field of the rectangle is visualized with Paraview.

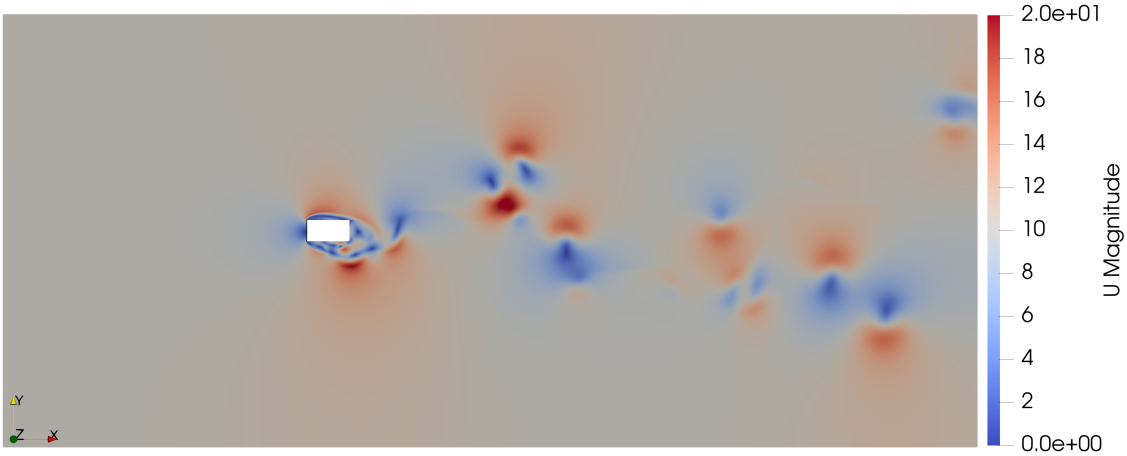


Figure C.7: Velocity flow field in metres per second around rectangle with width-to-height ratio of 2. Snapshot corresponds to fully developed vortex shedding conditions, captured at the simulation time 60.5 seconds.

In Figure C.8, the upper graph depicts lift coefficient as a function of time for the rectangle. The lower graph depicts Strouhal numbers, determined through an FFT of the lift coefficient from the times 20 to 99 seconds.

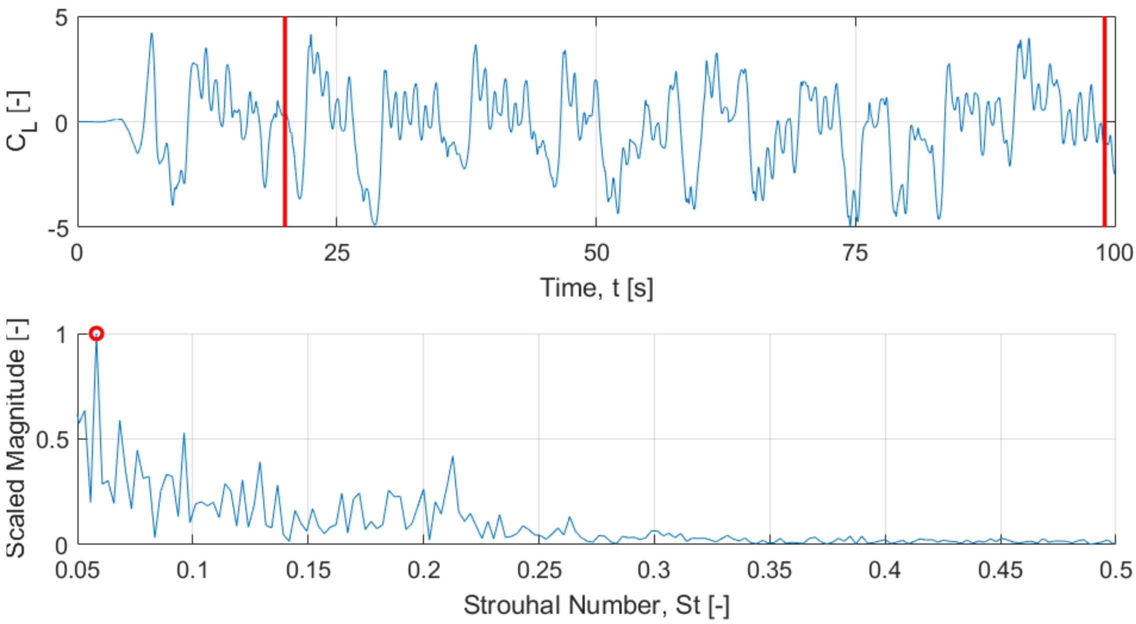


Figure C.8: Lift coefficient as a function of time and Strouhal numbers corresponding to the frequencies of an FFT on the lift coefficient between times 20 and 99 seconds, for a rectangle with width-to-height ratio of 2.

As seen in Figure C.8, the Strouhal number is extracted at the 0.058 as it corresponds to the tallest peak above 0.05.

C.3.3 Rectangle with $b/d_4 = 5$

In Figure C.5, the velocity flow field of the rectangle is visualized with Paraview.

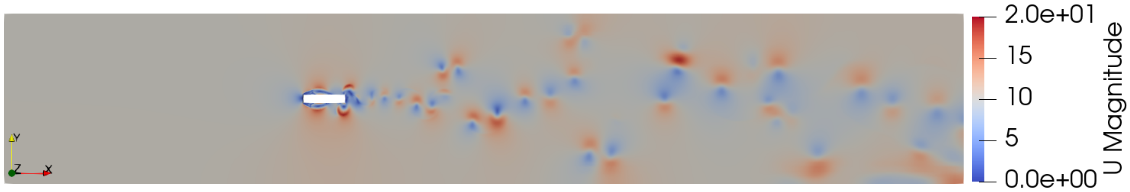


Figure C.9: Velocity flow field in metres per second around rectangle with width-to-height ratio of 5. Snapshot corresponds to fully developed vortex shedding conditions, captured at the simulation time 65.3 seconds.

In Figure C.10, the upper graph depicts lift coefficient as a function of time for the rectangle. The lower graph depicts Strouhal numbers, determined through an FFT of the lift coefficient from the times 30 to 124 seconds.

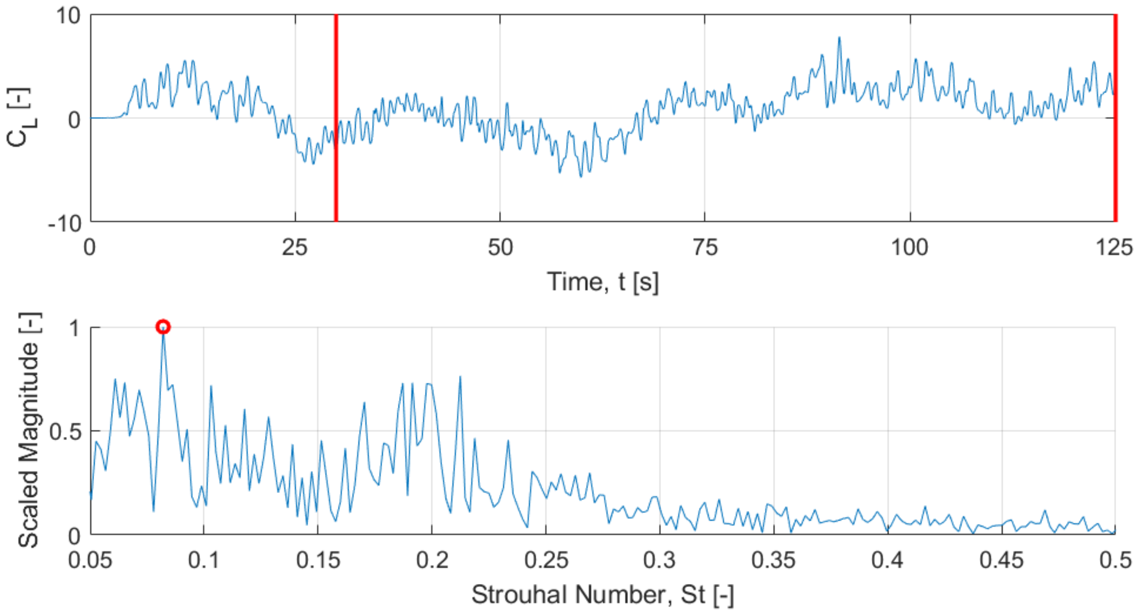


Figure C.10: Lift coefficient as a function of time and Strouhal numbers corresponding to the frequencies of an FFT on the lift coefficient between times 30 and 124 seconds, for a rectangle with width-to-height ratio of 5.

As seen in Figure C.6, the Strouhal number is extracted at the highest peak corresponding to 0.082. The cluster further right is neglected as the corresponding vibration amplitudes will be low.

C.4 Verification

Figure C.11 shows the relationship between Strouhals number, St, and the width-to-height ratio, b/d_4, for rectangular sections.

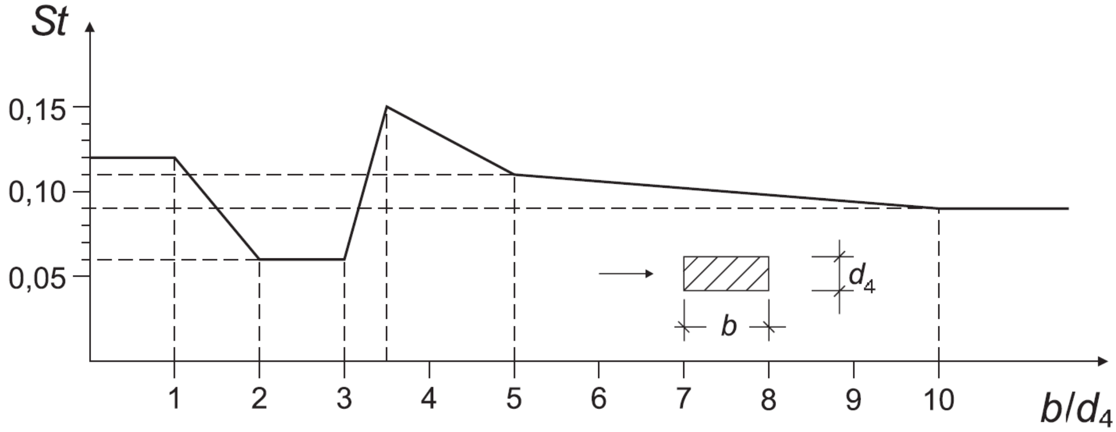


Figure C.11: Strouhal Number, St, for rectangular sections with sharp corners. Reproduced from SS-EN 1991-1-4:2005.

In Table C.4, simulation results and corresponding tabulated data for the studied rectangles is presented. The drag coefficient from the norm is extracted from Section 7.6 of SS-EN 1991-1-4:2005. Mean lift and drag coefficients from simulations are calculated with MATLAB. Tabulated Strouhal numbers are extracted from Figure C.11 for respective rectangles. Simulated Strouhal numbers are extracted from Figures C.6, C.8 and C.10 for the respective rectangles. The difference in Strouhal number to the norm, ΔSt, is calculated as $(St_{sim} - St_{norm})/St_{norm}$.

Table C.4: Simulation results of Strouhal numbers and mean drag and lift coefficients, simulated at a wind velocity of 10 m/s, and tabulated data, for rectangles of various width-to-height ratios, b/d_4.

b/d_4	$C_{D,norm}$	$C_{D,sim}$	$C_{L,sim}$	St_{norm}	St_{sim}	ΔSt
1	2.10	2.66	0.031	0.12	0.084	-30.4 %
2	1.65	1.54	-0.359	0.06	0.058	-3.3 %
5	1.00	1.40	-0.429	0.11	0.082	-25.4 %

Note that for symmetric objects such as rectangles with wind flow normal to the cross-wind dimension, the mean lift force should be zero. Hence, deviations from this is an error.

D

Matlab Code for Data Interpretation

```
function [Iter, c_l, scaledP1, posmax, St, t1, t2] = ...
    strouhalPlot(forceCoeffs, U, d4, time_1, time_2)
%function [Iter, c_l, scaledP1, posmax, St, t1, t2] = ...
%    strouhalPlot(forceCoeffs, U, d4, time_1, time_2)
%
%   Function for post-processing of data from simulations in OpenFOAM.
%
%   INPUTS:
%       forceCoeffs (.dat file loaded from OpenFOAM)
%       U        - free stream velocity [m/s]
%       d4       - height of bridge (cross-wind dimension) [m]
%       time_1   - sampling time of lift coefficient, lower limit [s]
%       time_2   - sampling time of lift coefficient, upper limit [s]
%
%   OUTPUTS:
%       Iter     - time steps [s]
%       c_l      - variation of lift coefficient in time
%       scaledP1 - scaled magnitude from FFT, correspionding to St
%       St       - strouhal values
%       t1       - lower limit of the sampling time (for plotting)
%       t2       - upper limit of the sampling time (for plotting)
%
%   Written by: Lukas Ehn & Sven Lundell

% LOAD INPUTS
    % Force Coefficients
    Iter = forceCoeffs(:,1);
    c_l  = forceCoeffs(:,4);
    c_d  = forceCoeffs(:,2);

    % Sample Times
    num1 = find(forceCoeffs(:,1)>time_1,1);
    if nargin == 4
        time_2 = forceCoeffs(end - 1,1);
    elseif nargin == 5
```

```
    else
        error('Wrong number of inputs')
    end
num2 = find(forceCoeffs(:,1)>time_2,1);
    t1 = time_1; t2 = time_2;

% FFT OF LIFT COEFFICIENT
    L  = num2 - num1;
    Y  = fft(c_l(num1:num2));
    P2 = abs(Y/L);
    P1 = P2(1:L/2+1);
    P1(2:end - 1) = 2*P1(2:end-1);
    Fs = L/(time_2 - time_1);
    F  = Fs*(0:(L/2))/L;

% STROUHAL NUMBERS
    St = F * d4/U;

% SCALING
    [~,posSt1] = find(St>0.05,1);
    [~,posSt2] = find(St>0.154,1);
    P1_Man = P1(posSt1:posSt2);
    scaledP1 = P1 / max(P1_Man);

% POSITION OF MAXIMUM STROUHAL NUMBER
    [~,posSt1] = find(St>0.05,1);
    [~,posSt2] = find(St>0.154,1);
    P1_Man = P1(posSt1:posSt2);
    [~,posmax] = max(P1_Man); posmax = posmax + (posSt1 - 1);

    disp(['Mean c_l = ', num2str(mean(c_l(num1:num2)))]);
    disp(['Mean c_d = ', num2str(mean(c_d(num1:num2)))]);
    disp(' ')
end
```

E

Directory Structure and Settings in OpenFOAM

In this appendix, the directory structure and the files needed to run a transient simulation using K-omega SST DES is presented. The files are from the simulation on a rectangle with a width-to-height ratio of 5 with a wind velocity of 10 m/s. All commands needed to run a case are included in the the text file *Commands*. Cleaning of all folders prior to running a case is recommended and it is performed with the executable *Allclean*. Furthermore, mesh generation and copying of the *0.orig* folder is performed with the executable *Prerun*.

```
Case Folder
├─ crossSection_snappyHexMesh
├─ crossSection_simpleFoam
├─ crossSection_pimpleFoam
├─ Allclean
├─ Prerun
└─ Commands.txt
```

```
crossSection_snappyHexMesh
├─ constant
│  └─ triSurface
│     ├─ crossSection.msh
│     ├─ crossSection.stl
│     └─ crossSection.stp
└─ system
   ├─ controlDict
   ├─ blockMeshDict
   ├─ surfaceFeatureExtractDict
   ├─ snappyHexMeshDict
   ├─ fvSchemes
   └─ fvSolution
```

```
crossSection_simpleFoam
├── 0.orig
│   ├── include
│   │   ├── fixedInlet
│   │   ├── frontBackTopBottomPatches
│   │   └── intitalConditions
│   ├── k
│   ├── nut
│   ├── omega
│   ├── p
│   └── U
├── constant
│   ├── transportProperties
│   └── turbulenceProperties
└── system
    ├── controlDict
    ├── forceCoeffs
    ├── createPatchDict
    ├── extrudeMeshDict
    ├── fvSchemes
    └── fvSolution
```

```
crossSection_pimpleFoam
├── constant
│   ├── transportProperties
│   └── turbulenceProperties
└── system
    ├── controlDict
    ├── forceCoeffs
    ├── renumberMeshDict
    ├── fvSchemes
    └── fvSolution
```

E.1 Allclean

```
#!/bin/sh
cd "${0%/*}" || exit                                # Run from this directory
. ${WM_PROJECT_DIR:?}/bin/tools/CleanFunctions      # Tutorial clean functions
#------------------------------------------------------------------------------

( cd crossSection_snappyHexMesh && cleanCase )
( cd crossSection_simpleFoam && cleanCase0 )
( cd crossSection_pimpleFoam && cleanCase0)
(
  cd crossSection_simpleFoam || exit
  rm -r logs
)
(
  cd crossSection_pimpleFoam || exit
  rm -r logs
)

#------------------------------------------------------------------------------
```

E.2 Prerun

```
#!/bin/sh
cd "${0%/*}" || exit                                # Run from this directory
. ${WM_PROJECT_DIR:?}/bin/tools/RunFunctions        # Tutorial run functions
#------------------------------------------------------------------------------

# Make 3D mesh in slab of cells.
(
    cd crossSection_snappyHexMesh || exit

    runApplication blockMesh
    runApplication surfaceFeatureExtract
    runApplication foamJob snappyHexMesh -overwrite
)

# Make a 2D mesh by extruding a patch and solve to steady state.
(
    cd crossSection_simpleFoam || exit
    runApplication extrudeMesh
    runApplication createPatch -overwrite
    restore0Dir
)
#------------------------------------------------------------------------------
```

E.3 Commands.txt

```
./Allclean
./Prerun
cd crossSection_simpleFoam
touch solution.foam
foamJob -s simpleFoam
foamLog log
simpleFoam -postProcess -func yPlus -latestTime
cd ..
cd crossSection_pimpleFoam
cp -r ../crossSection_simpleFoam/constant/polyMesh/ ./constant/.
mapFields ../crossSection_simpleFoam -sourceTime latestTime -consistent
renumberMesh -overwrite
touch solution.foam
foamJob -s pimpleFoam
```

E.4 crossSection_snappyHexMesh

E.4.1 constant

E.4.1.1 triSurface

```
  crossSection.msh  (from GMSH)
  crossSection.stl  (from CAD-program, here Autodesk Inventor Professional)
  crossSection.stp  (from CAD-program, here Autodesk Inventor Professional)
```

E.4.2 system

E.4.2.1 controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     snappyHexMesh;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         100;

deltaT          1;

writeControl    runTime;

writeInterval   1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  7;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

// ************************************************************************* //
```

E.4.2.2 blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
```

```
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

scale   1;

vertices
(
    (-75  -20  0)
    ( 150 -20  0)
    ( 150  20  0)
    (-75   20  0)
    (-75  -20  0.5)
    ( 150 -20  0.5)
    ( 150  20  0.5)
    (-75   20  0.5)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (225 40 1) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
    topAndBottom
    {
        type patch;
        faces
        (
            (3 7 6 2)
            (1 5 4 0)
        );
    }

    inlet
    {
        type patch;
        faces
```

```
        (
            (0 4 7 3)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }

    symFront
    {
        type symmetryPlane;
        faces
        (
            (4 5 6 7)
        );
    }

    symBack
    {
        type symmetryPlane;
        faces
        (
            (0 3 2 1)
        );
    }
);

mergePatchPairs
(
);

// ************************************************************************* //
```

E.4.2.3 surfaceFeatureExtractDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
```

```
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      surfaceFeatureExtractDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

crossSection.stl
{
    // How to obtain raw features (extractFromFile || extractFromSurface)
    extractionMethod    extractFromSurface;

    extractFromSurfaceCoeffs
    {
        // Mark edges whose adjacent surface normals are at an angle less
        // than includedAngle as features
        // - 0  : selects no edges
        // - 180: selects all edges
        includedAngle   150;
    }

    subsetFeatures
    {
        // Keep nonManifold edges (edges with >2 connected faces)
        nonManifoldEdges       no;

        // Keep open edges (edges with 1 connected face)
        openEdges       yes;
    }


    // Write options

        // Write features to obj format for postprocessing
        writeObj                yes;
}


// ************************************************************************* //
```

E.4.2.4 snappyHexMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Which of the steps to run
castellatedMesh true;
snap            true;
addLayers       true;


// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
    crossSection.stl
    {
        type    triSurfaceMesh;
        name    crossSection;
    }

    refinementBox1
    {
        type searchableBox;
        min   (-9   -5  0);
        max   ( 150  5  0.5);
    }

    refinementBox2
    {
```

```
        type searchableBox;
        min   (-14  -10  0);
        max   (150   10  0.5);
    }

    refinementBox3
    {
        type searchableBox;
        min   (-19   -15   0);
        max   (150    15   0.5);
    }

}


// Settings for the castellatedMesh generation.
castellatedMeshControls
{

    // Refinement parameters
    // ~~~~~~~~~~~~~~~~~~~~~

    // If local number of cells is >= maxLocalCells on any processor
    // switches from from refinement followed by balancing
    // (current method) to (weighted) balancing before refinement.
    maxLocalCells 10000000;

    // Overall cell limit (approximately). Refinement will stop immediately
    // upon reaching this number so a refinement level might not complete.
    // Note that this is the number of cells before removing the part which
    // is not 'visible' from the keepPoint. The final number of cells might
    // actually be a lot less.
    maxGlobalCells 200000000;

    // The surface refinement loop might spend lots of iterations refining just
    // a few cells. This setting will cause refinement to stop if <=
    // minimumRefine are selected for refinement. Note: it will at least do one
    // iteration (unless the number of cells to refine is 0)
    minRefinementCells 100;

    // Number of buffer layers between different levels.
    // 1 means normal 2:1 refinement restriction, larger means slower
    // refinement.
    nCellsBetweenLevels 6;
```

```
// Explicit feature edge refinement
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies a level for any cell intersected by its edges.
// This is a featureEdgeMesh, read from constant/triSurface for now.
features
(
    {
        file "crossSection.eMesh";
        level 6;
    }
);



// Surface based refinement
// ~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies two levels for every surface. The first is the minimum level,
// every cell intersecting a surface gets refined up to the minimum level.
// The second level is the maximum level. Cells that 'see' multiple
// intersections where the intersections make an
// angle > resolveFeatureAngle get refined up to the maximum level.

refinementSurfaces
{
    crossSection
    {
        // Surface-wise min and max refinement level
        level (5 6);
    }
}

// Resolve sharp angles on fridges
resolveFeatureAngle 5;


// Region-wise refinement
// ~~~~~~~~~~~~~~~~~~~~~~

// Specifies refinement level for cells in relation to a surface. One of
// three modes
// - distance. 'levels' specifies per distance to the surface the
//   wanted refinement level. The distances need to be specified in
//   descending order.
```

```
    // - inside. 'levels' is only one entry and only the level is used. All
    //   cells inside the surface get refined up to the level. The surface
    //   needs to be closed for this to be possible.
    // - outside. Same but cells outside.

    refinementRegions
    {

        refinementBox1
        {
          mode inside;
          levels ((1e15 4));
        }

        refinementBox2
        {
          mode inside;
          levels ((1e15 3));
        }

        refinementBox3
        {
          mode inside;
          levels ((1e15 2));
        }

    }


    // Mesh selection
    // ~~~~~~~~~~~~~~

    // After refinement patches get added for all refinementSurfaces and
    // all cells intersecting the surfaces get put into these patches. The
    // section reachable from the locationInMesh is kept.
    // NOTE: This point should never be on a face, always inside a cell, even
    // after refinement.
    locationInMesh (-74.9 19.9 0.002);


    // Whether any faceZones (as specified in the refinementSurfaces)
    // are only on the boundary of corresponding cellZones or also allow
    // free-standing zone faces. Not used if there are no faceZones.
    allowFreeStandingZoneFaces true;
}
```

```
// Settings for the snapping.
snapControls
{
    // Number of feature edge snapping iterations (disabled if omitted)
    nFeatureSnapIter 10;

    //- Number of patch smoothing iterations before finding correspondence
    //  to surface
    nSmoothPatch 5;

    //- Relative distance for points to be attracted by surface feature point
    //  or edge. True distance is this factor times local
    //  maximum edge length.
    tolerance 4.0;

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 40;

    //- Maximum number of snapping relaxation iterations. Should stop
    //  before upon reaching a correct mesh.
    nRelaxIter 10;
}



// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the undistorted
    // size of the refined cell outside layer (true) or absolute sizes (false).
    relativeSizes true;

    // Per final patch (so not geometry!) the layer information
    layers
    {
        "(crossSection).*"
        {
            nSurfaceLayers 2;
        }
    }

    // Expansion factor for layer mesh
    expansionRatio 1.3;
```

```
// Wanted thickness of final added cell layer. If multiple layers
// is the thickness of the layer furthest away from the wall.
// Relative to undistorted size of cell outside layer.
// See relativeSizes parameter.
finalLayerThickness 0.7;

// Minimum thickness of cell layer. If for any reason layer
// cannot be above minThickness do not add layer.
// Relative to undistorted size of cell outside layer.
// See relativeSizes parameter.
minThickness 0.25;

// If points get not extruded do nGrow layers of connected faces that are
// also not grown. This helps convergence of the layer addition process
// close to features.
// Note: changed(corrected) w.r.t 1.7.x! (didn't do anything in 1.7.x)
nGrow 0;

// Advanced settings

// When not to extrude surface. 0 is flat surface, 90 is when two faces
// are perpendicular
featureAngle 110;

// Maximum number of snapping relaxation iterations. Should stop
// before upon reaching a correct mesh.
nRelaxIter 5;

// Number of smoothing iterations of surface normals
nSmoothSurfaceNormals 1;

// Number of smoothing iterations of interior mesh movement direction
nSmoothNormals 3;

// Smooth layer thickness over surface patches
nSmoothThickness 10;

// Stop layer growth on highly warped cells
maxFaceThicknessRatio 0.5;

// Reduce layer growth where ratio thickness to medial
// distance is large
maxThicknessToMedialRatio 0.3;

// Angle used to pick up medial axis points
// Note: changed(corrected) w.r.t 16x! 90 degrees corresponds to 130 in 16x.
```

```
    minMedialAxisAngle 90;

    // Create buffer region for new layer terminations
    nBufferCellsNoExtrude 0;


    // Overall max number of layer addition iterations. The mesher will exit
    // if it reaches this number of iterations; possibly with an illegal
    // mesh.
    nLayerIter 100;
}



// Generic mesh quality settings. At any undoable phase these determine
// where to undo.
meshQualityControls
{
    //- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 65;

    //- Max skewness allowed. Set to <0 to disable.
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //- Max concaveness allowed. Is angle (in degrees) below which concavity
    //  is allowed. 0 is straight face, <0 would be convex face.
    //  Set to 180 to disable.
    maxConcave 80;

    //- Minimum pyramid volume. Is absolute volume of cell pyramid.
    //  Set to a sensible fraction of the smallest cell volume expected.
    //  Set to very negative number (e.g. -1E30) to disable.
    minVol 1e-13;

    //- Minimum quality of the tet formed by the face-centre
    //  and variable base point minimum decomposition triangles and
    //  the cell centre.  Set to very negative number (e.g. -1E30) to
    //  disable.
    //     <0 = inside out tet,
    //      0 = flat tet
    //      1 = regular tet
    minTetQuality 1e-30;

    //- Minimum face area. Set to <0 to disable.
    minArea -1;
```

```
    //- Minimum face twist. Set to <-1 to disable. dot product of face normal
    //  and face centre triangles normal
    minTwist 0.05;

    //- Minimum normalised cell determinant
    //  1 = hex, <= 0 = folded or flattened illegal cell
    minDeterminant 0.001;

    //- minFaceWeight (0 -> 0.5)
    minFaceWeight 0.05;

    //- minVolRatio (0 -> 1)
    minVolRatio 0.01;

    //must be >0 for Fluent compatibility
    minTriangleTwist -1;


    // Advanced

    //- Number of error distribution iterations
    nSmoothScale 10;
    //- Amount to scale back displacement at error points
    errorReduction 0.75;
}


// Advanced

// Merge tolerance. Is fraction of overall bounding box of initial mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1e-6;


// ************************************************************************* //
```

E.4.2.5 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
```

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss limitedLinearV 1;
    div(phi,k)      Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)      Gauss upwind;
    div(R)          Gauss linear;
    div(phid,p)     Gauss limitedLinear 1;
    div(phi,K)      Gauss limitedLinear 1;
    div(phi,e)      Gauss limitedLinear 1;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear limited corrected 0.5;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
```

```
    default         corrected;
}


// ************************************************************************* //
```

E.4.2.6 fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-12;
        relTol          0;
    }

    rho
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0;
    }

    "(U|e|k|epsilon|R)"
    {
        $p;
        tolerance       1e-08;
```

```
        relTol          0;
    }
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 2;
}


// ************************************************************************* //
```

E.5 crossSection_simpleFoam

E.5.1 0.orig

E.5.1.1 include

fixedInlet

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

inlet
{
    type  fixedValue;
    value $internalField;
}

// ************************************************************************* //
```

frontBackTopBottomPatches

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

topAndBottom
```

```
{
    type slip;
}

front
{
    type empty;
}

back
{
    type empty;
}

// ************************************************************************* //
```

A.2.1.1.1 fixedInlet

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

inlet
{
    type  fixedValue;
    value $internalField;
}

// ************************************************************************* //
```

A.2.1.1.2 frontBackTopBottomPatches

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

topAndBottom
{
    type slip;
}
```

```
front
{
    type empty;
}

back
{
    type empty;
}

// ************************************************************************* //
```

A.2.1.1.3 initialConditions

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

flowVelocity         (10 0 0);
pressure             0;
turbulentKE          3.375;
turbulentOmega       0.91856;

// ************************************************************************* //
```

E.5.1.2 k

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
#include        "include/initialConditions"

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform $turbulentKE;

boundaryField
{
    #include "include/fixedInlet"

    outlet
    {
        type            inletOutlet;
        inletValue      $internalField;
        value           $internalField;
    }

    crossSection
    {
        type            kqRWallFunction;
        value           $internalField;
    }

    #include "include/frontBackTopBottomPatches"
}


// ************************************************************************* //
```

E.5.1.3 nut

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      nut;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    crossSection
    {
        type            nutkWallFunction;
        value           uniform 0;
    }

    "(front|back|topAndBottom|inlet|outlet)"
    {
        type            calculated;
        value           uniform 0;
    }
}


// ************************************************************************* //
```

E.5.1.4 omega

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      omega;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"

dimensions      [0 0 -1 0 0 0 0];
```

```
internalField   uniform $turbulentOmega;

boundaryField
{
    #include "include/fixedInlet"

    outlet
    {
        type            inletOutlet;
        inletValue      $internalField;
        value           $internalField;
    }

    crossSection
    {
        type            omegaWallFunction;
        value           $internalField;
    }

    #include "include/frontBackTopBottomPatches"
}


// ************************************************************************* //
```

E.5.1.5 p

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"

dimensions      [0 2 -2 0 0 0 0];
```

```
internalField   uniform $pressure;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            fixedValue;
        value           $internalField;
    }

    crossSection
    {
        type            zeroGradient;
    }

    #include "include/frontBackTopBottomPatches"
}

// ************************************************************************* //
```

E.5.1.6 U

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include        "include/initialConditions"
```

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform $flowVelocity;

boundaryField
{
    #include "include/fixedInlet"

    outlet
    {
        type            inletOutlet;
        inletValue      uniform (0 0 0);
        value           $internalField;
    }

    crossSection
    {
        type            noSlip;
    }

    #include "include/frontBackTopBottomPatches"
}


// ************************************************************************* //
```

E.5.2 constant

E.5.2.1 transportProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      transportProperties;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

transportModel  Newtonian;

nu              15e-06;

// ************************************************************************* //
```

E.5.2.2 turbulenceProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType RAS;

RAS
{
    RASModel        kOmegaSST;

    turbulence      on;

    printCoeffs     on;
}

// ************************************************************************* //
```

E.5.3 system

E.5.3.1 controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
```

```
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     simpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         2000;

deltaT          1;

writeControl    runTime;

writeInterval   100;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

functions
```

```
{
    #include "forceCoeffs"
}

// ************************************************************************* //
```

E.5.3.2 forceCoeffs

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

forceCoeffs1
{
    type            forceCoeffs;

    libs            ("libforces.so");

    writeControl    timeStep;
    timeInterval    1;

    log             yes;

    patches         (crossSection);
    rho             rhoInf;      // Indicates incompressible
    rhoInf          1;           // Redundant for incompressible
    liftDir         (0 1 0);
    dragDir         (1 0 0);
    CofR            (0 0 0);     // Centre of rotation for moment calculations
    pitchAxis       (0 0 1);
    magUInf         10;          // Freestream velocity magnitude
    lRef            10;           // Reference length
    Aref            1;           // Reference area
    /*
    binData
    {
        nBin        20;          // output data into 20 bins
        direction   (1 0 0);     // bin direction
        cumulative  yes;
    }
    */
}
```

```
// ************************************************************************* //
```

E.5.3.3 extrudeMesh

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      extrudeMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// What to extrude:
//      patch   : from patch of another case ('sourceCase')
//      mesh    : as above but with original case included
//      surface : from externally read surface

constructFrom patch;
sourceCase "../crossSection_snappyHexMesh";
sourcePatches (symFront);

// If construct from patch: patch to use for back (can be same as sourcePatch)
exposedPatchName symBack;

// Flip surface normals before usage. Valid only for extrude from surface or
// patch.
flipNormals false;

//- Linear extrusion in point-normal direction
extrudeModel        linearNormal;

nLayers             1;

expansionRatio      1.0;

linearNormalCoeffs
```

```
{
    thickness       0.5;
}

// Do front and back need to be merged? Usually only makes sense for 360
// degree wedges.
mergeFaces false;   //true;

// Merge small edges. Fraction of bounding box.
mergeTol 0;


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

E.5.3.4 createPatch

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      createPatchDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
pointSync false;

patches
(
    {
        // Name of new patch
        name front;

        // Type of new patch
        patchInfo
        {
            type empty;
        }
```

```
        // How to construct: either from 'patches' or 'set'
        constructFrom patches;

        // If constructFrom = patches : names of patches. Wildcards allowed.
        patches (symFront);
    }
    {
        // Name of new patch
        name back;

        // Type of new patch
        patchInfo
        {
            type empty;
        }

        // How to construct: either from 'patches' or 'set'
        constructFrom patches;

        // If constructFrom = patches : names of patches. Wildcards allowed.
        patches (symBack);
    }
);

// ************************************************************************* //
```

E.5.3.5 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
```

```
    default steadyState;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss linearUpwind grad(U);
    div(phi,k)      bounded Gauss upwind;
    div(phi,omega)  bounded Gauss upwind;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

wallDist
{
    method meshWave;
}


// ************************************************************************* //
```

E.5.3.6 fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
```

```
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        smoother        GaussSeidel;
        tolerance       1e-7;
        relTol          0.01;
    }

    Phi
    {
        $p;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }

    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }
```

```
    omega
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 2;
    consistent yes;
}

potentialFlow
{
    nNonOrthogonalCorrectors 10;
}

relaxationFactors
{
    equations
    {
        U               0.9;
        k               0.7;
        omega           0.7;
    }
}

cache
{
    grad(U);
}

// ************************************************************************* //
```

E.6 crossSection_pimpleFoam

E.6.1 constant

E.6.1.1 transportProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
```

```
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      transportProperties;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

transportModel  Newtonian;

nu              15e-06;

// ************************************************************************* //
```

E.6.1.2 turbulenceProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType LES;

LES
{
    LESModel            kOmegaSSTDES;
```

```
    delta               maxDeltaxyz;

    printCoeffs         on;

    turbulence          on;

    maxDeltaxyzCoeffs
    {
    deltaCoeff          1;
    }
}

// ************************************************************************* //
```

E.6.2 system

E.6.2.1 controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     pimpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         125.1;
```

```
deltaT          0.0001; //1e-5;

writeControl    adjustableRunTime;

writeInterval   0.1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  8;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep  yes;

maxCo           5;
maxDeltaT       0.01;

functions
{
    #include "forceCoeffs"
}

// ************************************************************************* //
```

E.6.2.2 forceCoeffs

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  plus                                  |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

forceCoeffs1
{
    type            forceCoeffs;
```

```
    libs            ("libforces.so");

    writeControl    timeStep;
    timeInterval    1;

    log             yes;

    patches         (crossSection);
    rho             rhoInf;      // Indicates incompressible
    rhoInf          1;           // Redundant for incompressible
    liftDir         (0 1 0);
    dragDir         (1 0 0);
    CofR            (0 0 0);     // Centre of rotation for moment calculations
    pitchAxis       (0 0 1);
    magUInf         10;          // Freestream velocity magnitude
    lRef            10;           // Reference length
    Aref            1;           // Reference area
    /*
    binData
    {
        nBin        20;          // output data into 20 bins
        direction   (1 0 0);     // bin direction
        cumulative  yes;
    }
    */
}


// ************************************************************************* //
```

E.6.2.3 renumberMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  6                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      renumberMeshDict;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Write maps from renumbered back to original mesh
writeMaps true;

// Optional entry: sort cells on coupled boundaries to last for use with
// e.g. nonBlockingGaussSeidel.
sortCoupledFaceCells false;

// Optional entry: renumber on a block-by-block basis. It uses a
// blockCoeffs dictionary to construct a decompositionMethod to do
// a block subdivision) and then applies the renumberMethod to each
// block in turn. This can be used in large cases to keep the blocks
// fitting in cache with all the the cache misses bunched at the end.
// This number is the approximate size of the blocks - this gets converted
// to a number of blocks that is the input to the decomposition method.
//blockSize 1000;

// Optional entry: sort points into internal and boundary points
//orderPoints false;

method          CuthillMcKee;
//method          Sloan;
//method          manual;
//method          random;
//method          structured;
//method          spring;
//method          zoltan;             // only if compiled with zoltan support

CuthillMcKeeCoeffs
{
    // Reverse CuthillMcKee (RCM) or plain
    reverse true;
}

manualCoeffs
{
    // In system directory: new-to-original (i.e. order) labelIOList
    dataFile "cellMap";
}

// For extruded (i.e. structured in one direction) meshes
structuredCoeffs
{
    // Patches that mesh was extruded from. These determine the starting
    // layer of cells
```

```
    patches (front); //(movingWall);
    // Method to renumber the starting layer of cells
    method  random;

    // Renumber in columns (depthFirst) or in layers
    depthFirst true;

    // Reverse ordering
    reverse false;
}

springCoeffs
{
    // Maximum jump of cell indices. Is fraction of number of cells
    maxCo 0.01;

    // Limit the amount of movement; the fraction maxCo gets decreased
    // with every iteration
    freezeFraction 0.999;

    // Maximum number of iterations
    maxIter 1000;
}

blockCoeffs
{
    method          scotch;
    //method hierarchical;
    //hierarchicalCoeffs
    //{
    //    n           (1 2 1);
    //    delta       0.001;
    //    order       xyz;
    //}
}

zoltanCoeffs
{
    ORDER_METHOD    LOCAL_HSFC;
}

// ************************************************************************* //
```

E.6.2.4 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
```

```
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default       CrankNicolson 0.5;
}

gradSchemes
{
    //default         Gauss linear;
    //grad(p)         Gauss linear;
    //grad(U)         Gauss linear;

    default       cellLimited leastSquares 1;

    grad(U)       cellLimited Gauss linear 1;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss linearUpwind grad(U);
    div(phi,k)      Gauss linearUpwind default; // limitedLinear 1;
    div(phi,omega)  Gauss linearUpwind default; // limitedLinear 1;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    //default         Gauss linear limited uncorrected;//corrected 0.5;
    default         Gauss linear limited 1;
}
```

```
interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited 1; //uncorrected; //corrected
}

wallDist
{
    method meshWave;
}


// ************************************************************************* //
```

E.6.2.5 fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{

    p
    {
        solver          GAMG;
        tolerance       1e-6;
        relTol          0.01;
        smoother        GaussSeidel;
        nPreSweeps      0;
```

```
        nPostSweeps      2;
        cacheAgglomeration on;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 200;
        mergeLevels      1;
    minIter       2;
    }

    pFinal
    {
      solver           GAMG;
      tolerance        1e-6;
      relTol           0.0;
      smoother         GaussSeidel;
      nPreSweeps       0;
      nPostSweeps      2;
      cacheAgglomeration on;
      agglomerator     faceAreaPair;
      nCellsInCoarsestLevel 200;
      mergeLevels      1;
    minIter     3;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
      minIter     3;
    }

    UFinal
    {
        solver          PBiCGStab;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
      minIter     3;
    }

    omega
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
```

```
        relTol          0;
      minIter     3;
    }

    omegaFinal
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
      minIter     3;
    }

    k
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
      minIter     3;
    }

    kFinal
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
      minIter     3;
    }
}

PIMPLE
{
    turbOnFinalIterOnly      false;
    momentumPredictor        yes;
    nOuterCorrectors         50;
    nCorrectors              4;
    nNonOrthogonalCorrectors 1;
      pRefCell      0;
      pRefValue     0;

    residualControl
    {
        U
        {
```

```
                tolerance  1e-5;
                relTol      0;
        }
        p
        {
                tolerance  1e-5;
                relTol      0;
        }
     }

}

// ************************************************************************* //
```

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY