



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

On the Stability of a Fluid–Beam Interaction in a Wind Tunnel

TESI DI LAUREA MAGISTRALE IN
MATHEMATICAL ENGINEERING - INGEGNERIA MATEMATICA

Author: **Pietro Di Giustino**

Student ID: 276042

Advisor: Prof. Filippo Gazzola

Co-advisors: Dr. Edoardo Bocchi

Academic Year: 2025-26

Abstract

This work investigates a fluid-structure interaction problem set in a wind tunnel, around a three-dimensional beam placed transversely to the flow. On a larger scale, this models the action of the wind on a bridge. Several cross-sectional shapes of the bridge are considered. The beam is taken as rigid and held in a prescribed position during each simulation. The flow is governed by the stationary Navier-Stokes equations for a viscous, incompressible and homogeneous fluid, with a prescribed inflow profile and no-slip conditions on the tunnel walls and on the beam surface.

The analysis addresses two questions. The first concerns the stability of the flow as inflow conditions and beam cross-section vary. The transition from a stationary to an unsteady regime is detected through the convergence of the lift force: as long as the lift converges to a constant value the flow is regarded as steady, whereas the onset of sustained oscillations translates into loss of stationarity. The velocity of the inflow is measured by a parameter λ , which is proportional to the Reynolds number, and the critical value λ_c is recorded for each configuration. The second question concerns the equilibrium positions of the beam, defined as those vertical positions z^* of its centre at which the lift vanishes; the stability of each one is inferred from the sign of the lift in a neighbourhood of z^* , classifying as stable those configurations for which the lift acts as a restoring force.

The numerical experiments complement the theoretical results proved in [6]. For small values of λ a single equilibrium is detected at the centre of the channel, stable by symmetry. As λ increases uniqueness of the equilibrium is lost, and further positions appear which move progressively away from the centre. The influence of the cross-sectional geometry on the critical threshold is then assessed through two parametric families of elliptical sections: in the first the perimeter of each ellipse equals the circumference of a reference circle, in the second it is the cross-sectional area that is fixed to that of the same circle. The constant-perimeter sections are found to be more stable than the reference circle, and the two families are finally compared at equal aspect ratio.

Keywords: fluid-structure interaction, Navier-Stokes equations, wind tunnel, beam aerodynamics, equilibrium configurations, flow stability

Abstract in lingua italiana

Il presente lavoro studia un problema di interazione fluido-struttura in una galleria del vento, attorno a una trave tridimensionale disposta trasversalmente al flusso. Questo modello, in larga scala, rappresenta l'azione del vento su un ponte. Vengono prese in esame diverse forme della sezione del ponte. La trave è assunta rigida e mantenuta in una posizione prescritta durante ciascuna simulazione. Il flusso è governato dalle equazioni di Navier-Stokes stazionarie per un fluido viscoso, incomprimibile e omogeneo, con un profilo di velocità imposto in ingresso e condizione di aderenza sulle pareti della galleria e sulla superficie della trave.

L'analisi affronta due questioni. La prima riguarda la stabilità del flusso al variare delle condizioni di inflow e della sezione della trave. La transizione da un regime stazionario a uno non stazionario è individuata attraverso la convergenza della forza di portanza: finché la portanza converge a un valore costante il flusso è ritenuto stazionario, mentre l'insorgere di oscillazioni persistenti traduce la perdita di stazionarietà. La velocità in ingresso del flusso è misurata tramite il parametro λ , proporzionale al numero di Reynolds, di cui si registra per ciascuna configurazione il valore critico λ_c . La seconda questione riguarda le posizioni di equilibrio della trave, definite come quelle quote z^* del baricentro in cui la portanza si annulla; la stabilità di ciascuna è dedotta dal segno della portanza in un intorno di z^* , classificando come stabili le configurazioni in cui essa agisce come forza di richiamo.

Le simulazioni numeriche completano i risultati teorici dimostrati in [6]. Per piccoli valori di λ si individua un'unica posizione di equilibrio al centro del canale, stabile per simmetria. All'aumentare di λ si perde l'unicità dell'equilibrio e compaiono ulteriori posizioni che si allontanano progressivamente dal centro. L'influenza della geometria sulla soglia critica è poi valutata attraverso due famiglie di sezioni ellittiche: nella prima il perimetro di ogni ellisse coincide con la circonferenza di un cerchio di riferimento, nella seconda ne coincide l'area. Le sezioni a perimetro costante risultano più stabili del cerchio di riferimento, e le due famiglie sono infine confrontate a parità di rapporto tra gli assi.

Parole chiave: interazione fluido-struttura, equazioni di Navier-Stokes, galleria del vento, aerodinamica della trave, configurazioni di equilibrio, stabilità del flusso

Contents

Abstract	i
Abstract in lingua italiana	iii
Contents	v
1 Introduction	1
2 Mathematical Model	3
2.1 Physical Model	3
2.2 Theoretical Results	4
3 Dependence of Flow Stability with respect to Inlet Profile	7
3.1 Introduction	7
3.2 Geometry and Numerical Setup	9
3.3 Methodology	11
3.4 Numerical Results	13
3.5 Discussion	14
4 Equilibrium positions in a Poiseuille flow	17
4.1 Introduction	17
4.2 Geometry and Numerical Setup	17
4.3 Methodology	18
4.4 Numerical Results	19
4.4.1 Equilibrium Position for given Velocity	19
4.4.2 Comparison for Different Values of λ	20
4.5 Discussion	21
4.5.1 Comparison with the Theoretical Framework	22
4.5.2 A Similar Model: Leonardo da Vinci's Ferry	23

5	Flow Stability of Elliptical Cross-Sections	25
5.1	Introduction	25
5.2	Geometry and Numerical Setup	25
5.3	Methodology and Numerical Results	27
5.3.1	Constant Perimeter	27
5.3.2	Constant Area	28
5.3.3	Comparison	29
5.4	Discussion	30
6	Conclusions and future developments	33
	Bibliography	35
A	Geometry and Numerical Setup in OpenFOAM	37
A.1	Initial and Boundary Conditions	37
A.2	Mesh and Domain	39
A.3	Numerical Setup	42
A.4	Simulation Control and Run-time Settings	44
B	MATLAB Scripts	47
B.1	Analysis of Wake Oscillations	47
B.2	λ_c vs p	49
B.3	Inlet Profile Comparison	50
B.4	Equilibrium Position Analysis	51
B.5	λ_c vs ratio	52
	List of Figures	55
	List of Tables	57

1 | Introduction

In this work, the theoretical results stemming mainly from [6] are studied and analyzed numerically. That paper investigates the equilibrium configurations of a fluid-structure interaction problem involving a bridge-like beam $\mathcal{B}$, of various shapes, immersed in a wind tunnel $P = (-R, R) \times (-1, 1)^2$, with $R \gg 1$. The fluid domain is defined as $\Omega = P \setminus \mathcal{B}$. All simulations were carried out by imposing and studying different inlet flow profiles, a zero-gradient condition at the outlet, and a no-slip condition on both the tunnel walls and the beam surface. The flow is viscous, homogeneous, incompressible and governed by the stationary Navier–Stokes equations

$$-\eta\Delta u + u \cdot \nabla u + \nabla p = 0, \quad \nabla \cdot u = 0 \quad \text{in } \Omega, \quad (1.1)$$

where u is the fluid velocity, p the scalar pressure, and η the kinematic viscosity.

In all the configurations considered, the beam $\mathcal{B}$ is held fixed and cannot deform during a simulation, although its initial position and shape vary in different simulations. More specifically, the cross-section of $\mathcal{B}$ is circular with radius 0.2 and center at $x = 0$ for all y , while the z -coordinate of the center varies depending on the configuration under study, as in Chapters 3 and 4. In Chapter 5, the cross-section is taken elliptical.

The study of stability and equilibrium positions of bridge-like structures has broad engineering relevance [1–4, 7–15, 17–23]. Understanding how a bridge responds to wind loading is an essential component of structural safety: wind-induced forces are far from negligible and can drive the structure into resonance if not properly accounted for in the design process. What was investigated here is how the fluid behaves around the structure, with the goal of identifying, for each geometry and inlet profile, the critical threshold λ_c — a parameter proportional to the Reynolds number of the system — beyond which the flow transitions from a steady state to a regular, symmetric oscillation. This oscillation generates a fluctuating lift force L that, in extreme cases, could subject a bridge to loads severe enough to cause structural failure. The objective in bridge design is therefore to identify cross-sectional shapes that minimize the lift force.

The thesis is organized as follows. Chapter 2 introduces the fluid–structure interaction

model [6] on which this thesis builds. We recall the existence, uniqueness, and stability results obtained for its equilibrium positions, collecting the main statements and theorems. The chapter then traces the passage from this analytical formulation to the numerical model, which made possible all the simulations carried out in the following chapters. Chapter 3 considers a beam with a circular cross-section of radius 0.2, with center fixed at $(x, z) = (0, 0)$, extending along the y -direction. The inlet flow is parametrized by a family of profiles defined by the formula

$$U(y, z) = \lambda c_p (1 - |y|^p) (1 - |z|^p) \quad (1.2)$$

where λ is a parameter proportional to the flow velocity and to the Reynolds number, as detailed in Chapter 3. For each profile exponent p , the value of λ is increased gradually until the critical threshold λ_c is reached, defined as the value beyond which the flow begins to oscillate. The critical thresholds are then compared across different values of p , with the aim of understanding how flow stability depends on the shape of the incoming wind profile. The main finding is that as p increases, the bridge becomes more stable, with λ_c growing monotonically and appearing to approach an asymptotic value.

Chapter 4 addresses a different aspect of the same geometry. Starting from the same circular cross-section of Chapter 3, the lift force acting on the beam is studied for different values of λ . For each value, the z -coordinate of the beam center at which the lift — or its time-average in the oscillatory regime — vanishes is identified, defining that position as an equilibrium position. This analysis is carried out across a range of λ values to track how both the location and the number of equilibrium points evolve. The results confirm what was anticipated by [5] and [6]: at low values of λ a single stable equilibrium exists at the center, while above a critical threshold two new stable equilibria appear symmetrically on both sides of the center and the central equilibrium becomes unstable. As λ increases further, the two stable equilibria move progressively away from the center.

Chapter 5 extends the stability analysis to a family of elliptical cross-sections. The cylindrical beam, which in the previous chapters had a circular base, now has an elliptical one. Using the same methodology as in Chapter 3, λ_c is computed for several ellipses, compared in two distinct ways: first at fixed perimeter equal to that of the reference circle, then at fixed area. The results are presented as a function of the aspect ratio a/b , where a and b denote the semi-axes along the x - and z -directions, respectively. As expected, increasing a/b — that is, making the ellipse more elongated — raises λ_c in both cases, ensuring more flow stability. For equal aspect ratio, the constant-perimeter family consistently yields higher critical thresholds than the constant-area family.

2 | Mathematical Model

2.1. Physical Model

In this chapter, the mathematical and physical framework underlying the numerical simulations and results presented in this thesis is described. In [6], several results are established, which are here complemented numerically. As already anticipated in the introduction, the problem is set in a rectangular parallelepiped

$$P = (-R, R) \times (-1, 1)^2, \quad R \gg 1, \quad (2.1)$$

representing a wind tunnel, crossed by a bridge-like beam $\mathcal{B}$, attached to the lateral walls at $y = \pm 1$, with the inlet flow directed along the x -axis and the bridge oriented perpendicularly to it. The fluid domain is defined as $\Omega = P \setminus \mathcal{B}$, which is then not simply connected. The fluid is viscous, homogeneous, incompressible and governed by the stationary Navier-Stokes equations introduced in (1.1). The quantity of interest is the lift force $L = L(y, u, p)$, directed vertically upward, exerted by the fluid on the beam cross-section $\mathcal{B}_y$. In [6], the beam $\mathcal{B}$ is allowed to deform, with its vertical displacement described by a function $h = h(y)$. As a result, the beam reaches a new position $\mathcal{B}^h$, and the fluid domain is accordingly modified to

$$\Omega_h = P \setminus \mathcal{B}^h = P \setminus \{(x, y, z + h(y)) \text{ with } (x, y, z) \in \mathcal{B}\}. \quad (2.2)$$

In the absence of flow, $h \equiv 0$.

The problem under study then consists of the following coupled system:

$$\begin{cases} -\eta\Delta u + u \cdot \nabla u + \nabla p = 0, & \nabla \cdot u = 0 & \text{in } \Omega_h, \\ u = 0 & & \text{on } \partial\mathcal{B}^h, \\ u = \lambda V e_1 & & \text{on } \Gamma_y^\pm \setminus \mathcal{B}_{\pm 1} \cup \Gamma_z^\pm, \\ u = \lambda V_i e_1 & & \text{on } \Gamma_x^-, \\ u = \lambda V_o e_1 & & \text{on } \Gamma_x^+, \end{cases} \quad (2.3)$$

where $\lambda > 0$ is the same parameter as in the previous chapters, controlling the intensity of the flow, a no-slip condition is imposed on the beam surface, and the inflow and outflow profiles V_i, V_o are prescribed at the faces $\{x = \pm R\}$. The lift force is defined as the vertical component of the fluid stress acting on each cross-section $\mathcal{B}_y^h$:

$$L(y, u, p) = -e_3 \cdot \int_{\partial\mathcal{B}_y^h} \mathbb{T}(u, p) n = -e_3 \cdot \int_{\partial\mathcal{B}_y^h} [\eta (\nabla u + \nabla^T u) - p \mathbb{I}] n d\sigma, \quad (2.4)$$

where $\mathbb{I}$ is the 3×3 identity matrix and n is the outward unit normal to $\partial\mathcal{B}^h$, hence pointing inside $\mathcal{B}^h$. The vertical position of the barycenter of each cross-section $\mathcal{B}_y$ satisfies the beam equation

$$h''''(y) + f(h(y)) = L(y, u, p) = L(y, h(y)), \quad (2.5)$$

where $f(h)$ denotes the elastic restoring forces due to cables, hangers and gravity, satisfying $f(0) = 0$. The two equations are coupled through L : the Navier-Stokes equations depend on Ω_h , which in turn depends on h , while the lift force depends on u and p , the solutions of the Navier-Stokes equations.

Depending on how the beam is fixed to the walls, two boundary conditions are considered: Clamped means $h(\pm 1) = h'(\pm 1) = 0$ or Hinged $h(\pm 1) = h''(\pm 1) = 0$. To find the equilibrium configurations of the FSI problem, one must solve the coupled system formed by (2.3) and (2.5). The goal is to analyse the existence, uniqueness and stability of such equilibrium positions.

2.2. Theoretical Results

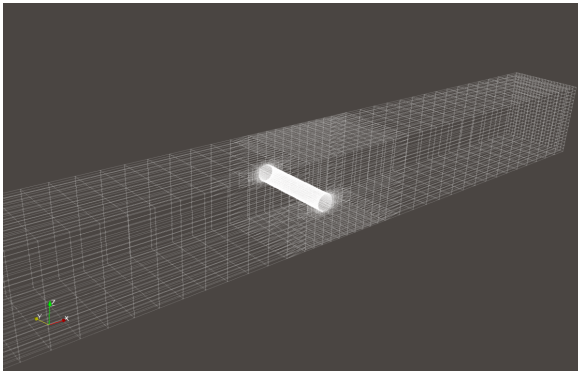
The two main results of [6] relevant to this thesis are the following.

Proposition 2.1. *For sufficiently small inflow intensity λ , the FSI problem admits a*

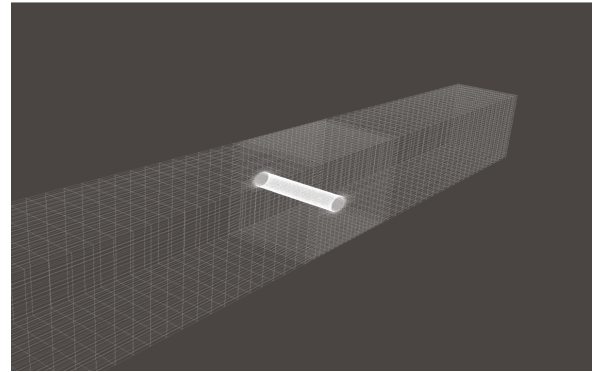
unique equilibrium solution (h, u, p) . In other words, at low flow velocities — equivalently, at low Reynolds numbers — the system settles into a single, well-defined equilibrium configuration.

Proposition 2.2. *Under symmetry assumptions on both the geometry and the inflow profile, the unique equilibrium solution is $h \equiv 0$, meaning the beam undergoes no vertical displacement and the lift force vanishes identically. The symmetric equilibrium position, with the beam centered in the tunnel, is the only possible one.*

Starting from these results, the thresholds of stability are determined numerically and, once we engaged with this class of problems, the investigation was pushed further, exploring the behaviour of the system for increasing values of λ , as already anticipated in the introduction. On the numerical side, starting from the model of [6], the computational domain was built as a three-dimensional rectangular parallelepiped $P = [-10, 10] \times [-1, 1] \times [-1, 1]$, which represents a wind tunnel. Inside this domain, the beam is geometrically represented in two different ways: in Chapters 3 and 4 as a circular cylinder of radius 0.2, centred at $x = 0$, extending along the entire y -axis, whose vertical position z varies across simulations, as discussed more specifically in Chapter 4; in Chapter 5, an elliptical cross-section is considered. The domain is discretized with a $60 \times 30 \times 30$ mesh, refined around the beam — as detailed in Chapter 3 — to ensure the robustness of the simulations. Both configurations are shown in Figure 2.1.



(a) Wind tunnel with circular cylinder.



(b) Wind tunnel with elliptical cross-section.

Figure 2.1: Rigid computational domain geometry.

The fundamental difference with respect to [6] is that in our CFD simulations, performed in OpenFOAM, the beam cannot deform. Therefore, in order to analyse equilibrium positions, existence and stability, the lift force was monitored directly, moving the beam upward between successive simulations, and an equilibrium position was defined as the

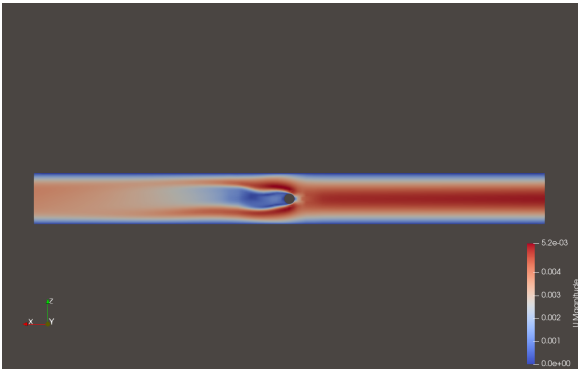
height z^* of the cylinder's center at which the lift force acting on the beam vanishes. The flow moves along the x -direction, perpendicular to the cylinder. Unlike [6], only an inlet condition is imposed at the face $\{x = -10\}$, given by the velocity profile (1.2), while at the outlet $\{x = 10\}$ a zero-gradient condition is applied. The validity of each simulation is verified by checking that the flux integral at the inlet equals that at the outlet. No-slip conditions are enforced on all lateral walls and on the beam surface. The governing equations are the stationary incompressible Navier–Stokes equations

$$-\eta\Delta u + u \cdot \nabla u + \nabla p = 0, \quad \nabla \cdot u = 0 \quad \text{in } \Omega, \quad (2.6)$$

in full agreement with system (1.1) of [6]. Another important difference between the study carried out in [6] and the present work is that throughout this thesis the analysed lift is the total force exerted by the fluid on the entire beam, whereas in [6] the quantity $L(y, u, p)$ denotes the lift acting on the single cross-section B_y . The two lift forces are related through

$$\mathcal{L} = \int_{-1}^1 L(y, u, p) dy. \quad (2.7)$$

In the figures below, the wake oscillations from the wind tunnel experiment and from the numerical simulation are compared. The colour gradients in the simulation represent the different velocity values taken by the flow inside the channel; it is precisely these velocity gradients, together with the pressure distribution, that give rise to the lift force, which is the central quantity of interest throughout this thesis.



(a) Wake oscillations in the numerical simulation.



(b) Vortex shedding observed in the Bovisa wind tunnel.

Figure 2.2: Comparison of wake oscillations.

3 | Dependence of Flow Stability with respect to Inlet Profile

3.1. Introduction

In this chapter the stability of a beam, modelled as a cylinder placed inside a wind tunnel, is investigated. The quantity of interest is the critical value λ_c , a parameter proportional to the inflow velocity at which a bifurcation occurs: for $\lambda \leq \lambda_c$ the flow around the cylinder is steady, whereas just above λ_c it loses stability and gives rise to vortex shedding. The dependence of the critical value λ_c on the inlet profile is studied, and more specifically on the exponent p that governs the shape of the velocity distribution prescribed at the entrance of the computational domain. The expression of the profile also includes a normalisation factor c_p , whose role and explicit form are discussed later in this section. The inlet velocity distribution $U(y, z)$ is then defined as:

$$U(y, z) = \left(\frac{U_{max}}{c_p} \right) c_p (1 - |y|^p) (1 - |z|^p) \quad (3.1)$$

By defining the bifurcation parameter λ as:

$$\lambda = \frac{U_{max}}{c_p} \quad (3.2)$$

the expression is simplified into its operational form:

$$U(y, z) = \lambda c_p (1 - |y|^p) (1 - |z|^p) \quad (3.3)$$

The role of c_p is to make the bifurcation parameter λ comparable across inlet profiles with different values of p . Without a suitable normalisation, two profiles sharing the same λ but a different exponent would deliver a different amount of fluid flow into the domain, and any comparison between the corresponding critical values λ_c would lose its physical

meaning. The factor c_p is therefore chosen so that, at fixed λ , the total incoming flow rate does not depend on p . This requirement is enforced by imposing that the integral of the shape function over the inlet section, which in our setting is the square $[-1, 1] \times [-1, 1]$, equals one:

$$\int_{-1}^1 \int_{-1}^1 c_p (1 - |y|^p) (1 - |z|^p) dy dz = 1 \quad (3.4)$$

Exploiting the symmetry and the separability of the function, the double integral can be decomposed into the product of two identical one-dimensional integrals:

$$c_p \left[\int_{-1}^1 (1 - |y|^p) dy \right]^2 = 1. \quad (3.5)$$

The integration of the velocity component yields:

$$\int_{-1}^1 (1 - |y|^p) dy = 2 \int_0^1 (1 - y^p) dy = 2 \left[y - \frac{y^{p+1}}{p+1} \right]_0^1 = \frac{2p}{p+1} \quad (3.6)$$

Substituting this result into the normalization condition yields the explicit value for c_p :

$$c_p = \left(\frac{p+1}{2p} \right)^2 \quad (3.7)$$

A significant inlet profile is the Poiseuille flow with $p = 2$ that can be observed in Figure 3.1.

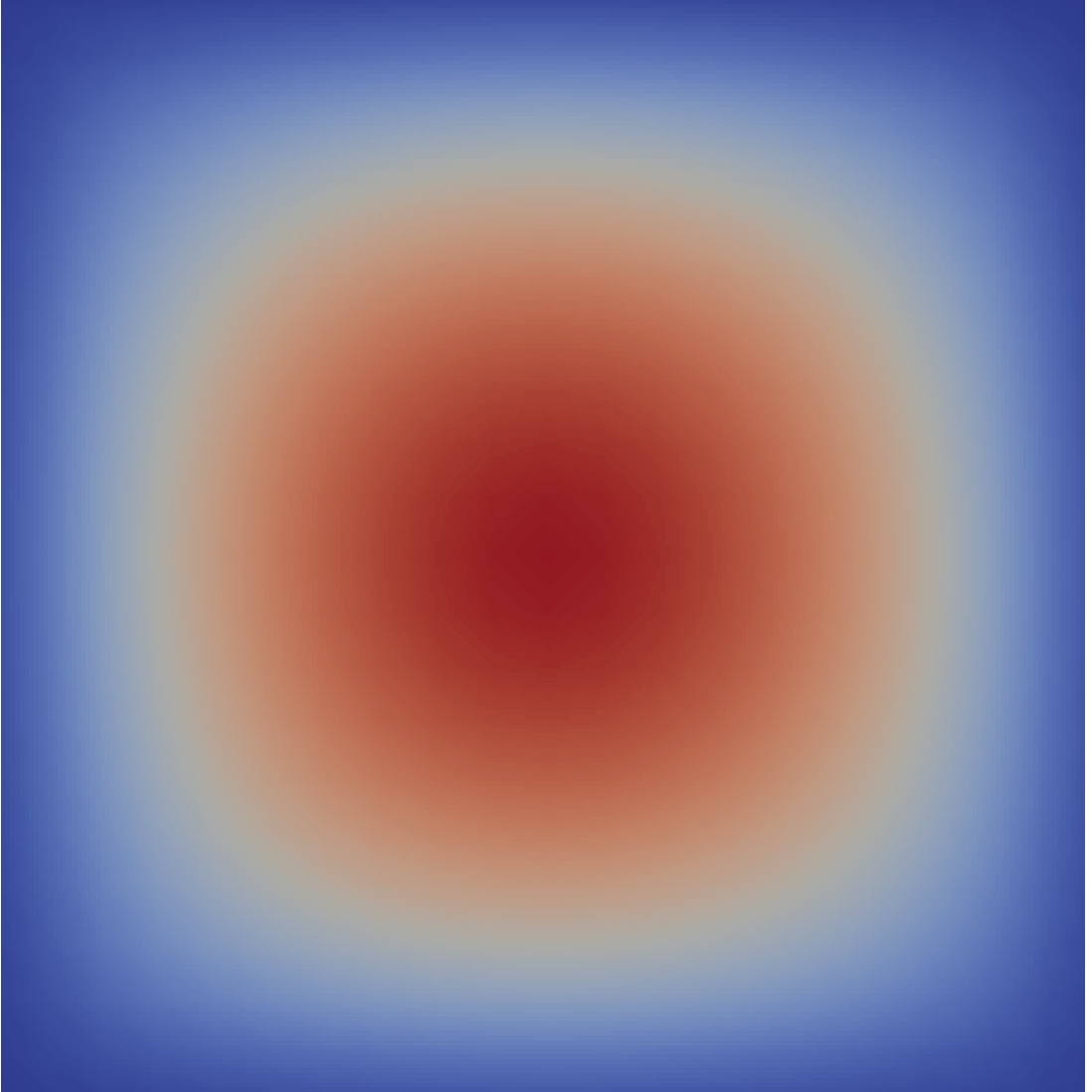


Figure 3.1: Poiseuille inlet profile

3.2. Geometry and Numerical Setup

The computational domain represents the wind tunnel in which the beam is placed. Adopting a Cartesian frame (x, y, z) in which x is the streamwise direction while y and z correspond to the spanwise and vertical directions, the domain is the parallelepiped $\Omega = [-10, 10] \times [-1, 1] \times [-1, 1]$, with overall dimensions $20 \times 2 \times 2$. The beam is modelled as a circular cylinder of radius $r = 0.2$, whose cross-section lies in the xz -plane and which extends across the entire y length of the domain. The cylinder axis is parallel to the y -axis and intersects the xz -plane at the origin, so that the geometry is symmetric with respect to the planes $x = 0$ and $z = 0$. No-slip boundary conditions for the fluid are imposed on all the walls of the domain, including the surface of the cylinder. A sketch of the resulting

configuration is shown in Figure 3.2.

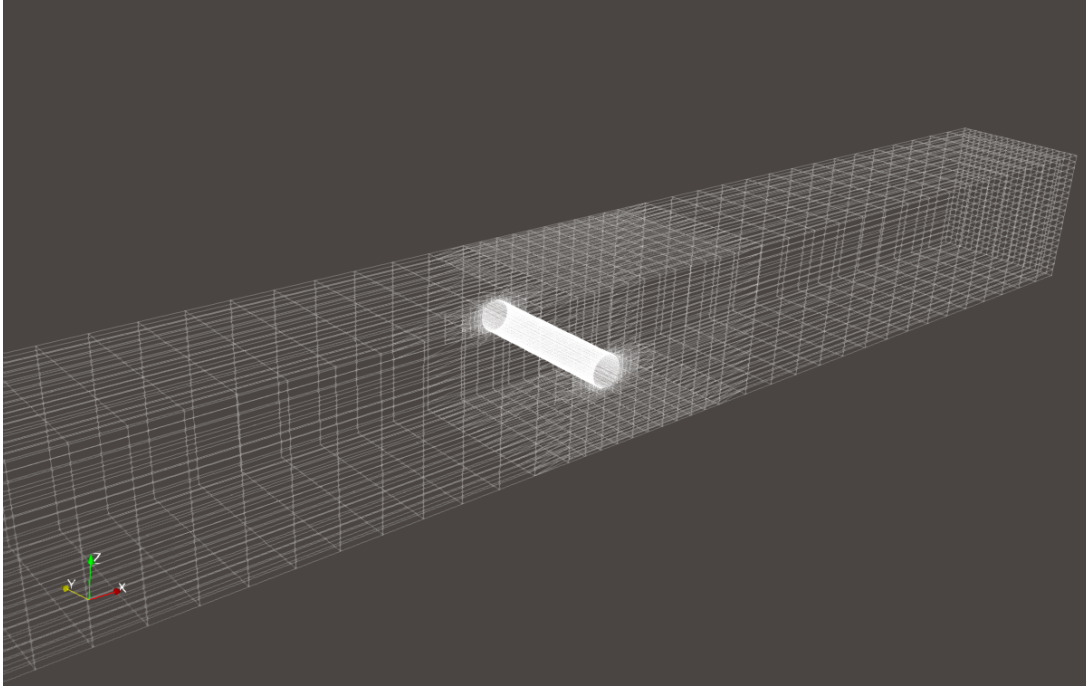


Figure 3.2: View of the computational domain.

The simulations are run with OpenFOAM, the open-source CFD library. The starting point is a structured background mesh of $60 \times 30 \times 30$ cells, uniform along the three coordinate directions, locally refined around the cylinder, where a better resolution is needed. The refinement is carried out with the `snappyHexMesh` utility, which fits the mesh to the cylindrical surface and subdivides the cells in its immediate vicinity. A close-up of the result is shown in Figure 3.3.

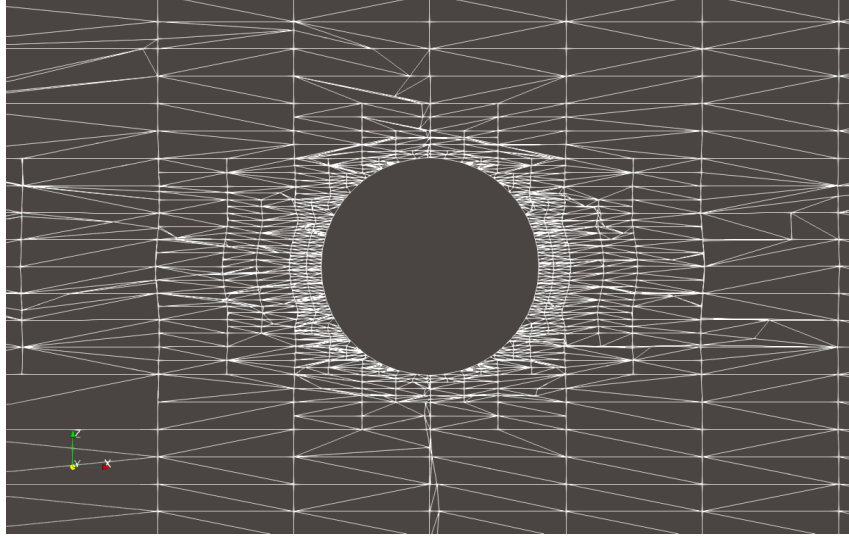


Figure 3.3: Detail of the mesh refinement around the cylinder.

The choice of the `simpleFoam` solver is grounded in both the mathematical formulation that motivates this work and in standard computational considerations. The fluid–structure interaction problem analysed in [6], which provides the theoretical starting point for the present thesis, is formulated through the *stationary* Navier–Stokes equations coupled with a static beam equation, and the existence and uniqueness of equilibrium configurations are established in a smallness regime. A steady-state solver is therefore the natural numerical counterpart of that continuous formulation, since it discretises the same class of equations on which the mathematical analysis relies. The same choice is also justified on physical and computational grounds. The objective of the simulations is to identify the critical threshold λ_c , so the cases of interest lie in the pre-critical regime, where the flow remains steady. A steady-state solver fits this regime naturally and proves substantially faster than a transient one, since it bypasses the resolution of fast time scales that are not relevant below the bifurcation. The same setting also yields a forward bifurcation indicator: for $\lambda \leq \lambda_c$ the residuals decrease monotonically and the solution converges to a stationary equilibrium, whereas for $\lambda > \lambda_c$ no stable steady solution exists, the residuals stagnate, and the lift force develops sustained oscillations that mark the transition to the unsteady regime.

3.3. Methodology

To determine the critical threshold λ_c , a large number of simulations was performed at a fixed value of the profile exponent p , varying the parameter λ and monitoring the behaviour of the lift force acting on the whole cylinder. For each value of λ , the simulation

was classified as stable or unstable depending on whether the lift signal converged to a constant value or developed sustained oscillations. The interval containing the transition was then progressively refined, narrowing the range of λ around the bifurcation point until a sufficiently accurate estimate of λ_c was obtained.

A representative example of this procedure is reported in Figure 3.4, which compares the lift signal in a stable case and in an unstable one for the Poiseuille profile ($p = 2$). In the first case, the lift converges towards a steady value, while in the second it exhibits persistent oscillations, clearly indicating that the critical threshold has been exceeded.

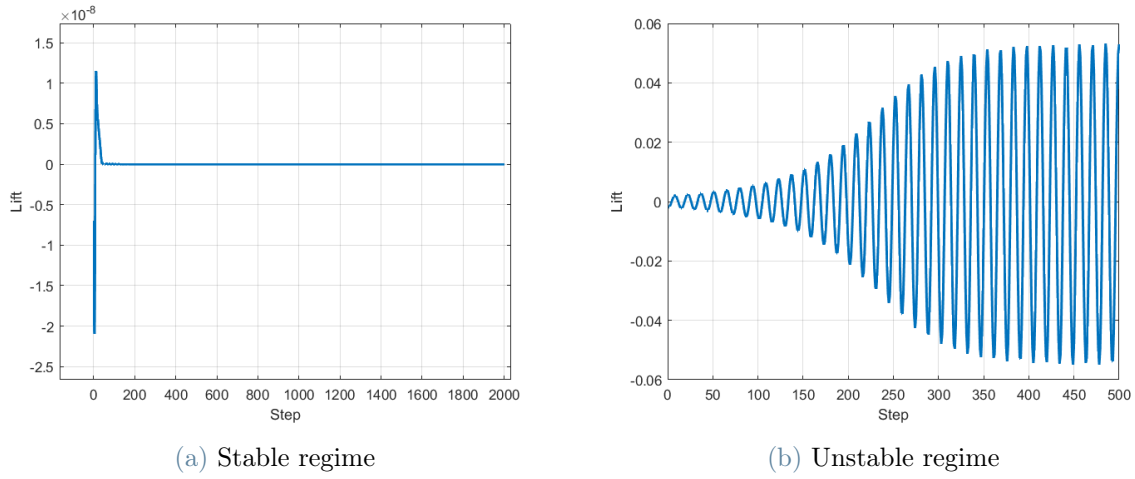


Figure 3.4: Comparison of lift force

The same procedure was also applied to the pressure field in the wake of the cylinder. The pressure was sampled at a fixed probe point downstream of the structure, and its evolution was analysed using the same criterion adopted for the lift: a constant value indicates a stable regime, while persistent fluctuations mark the transition to the unsteady one. The same comparison between a stable and an unstable case is reported in Figure 3.5. Both quantities were obtained through the post-processing tools available in OpenFOAM, which compute forces and field probes directly during the simulation.

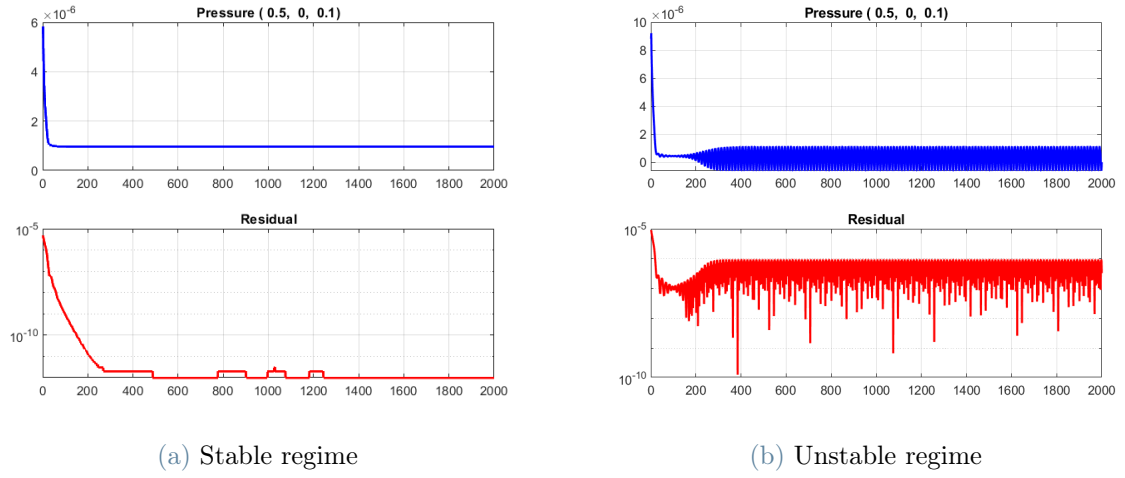


Figure 3.5: Comparison of Pressure

3.4. Numerical Results

This section presents the results of the parametric study conducted to identify the critical threshold λ_c as a function of the inlet profile exponent p . The analysis covers a wide range of exponents, specifically $p \in \{1, 2, 4, 8, 16, 30, 50\}$ in (3.3), making it possible to observe the transition from a sharp, triangular-like profile ($p = 1$) to an almost perfectly flat “plug-flow” distribution ($p = 50$) as observed in Figure 3.7. The results of this systematic search are summarized in Table 3.1.

Profile Exponent p	Critical Threshold λ_c
1	0.0040
2	0.0085
4	0.0092
8	0.0096
16	0.0097
30	0.0097
50	0.0097

Table 3.1: Critical values of the bifurcation parameter λ_c for different inlet profile exponents p in (3.3).

To better visualize the impact of the flow shape on the system stability, the relationship between the critical threshold and the exponent p is plotted in Figure 3.6.

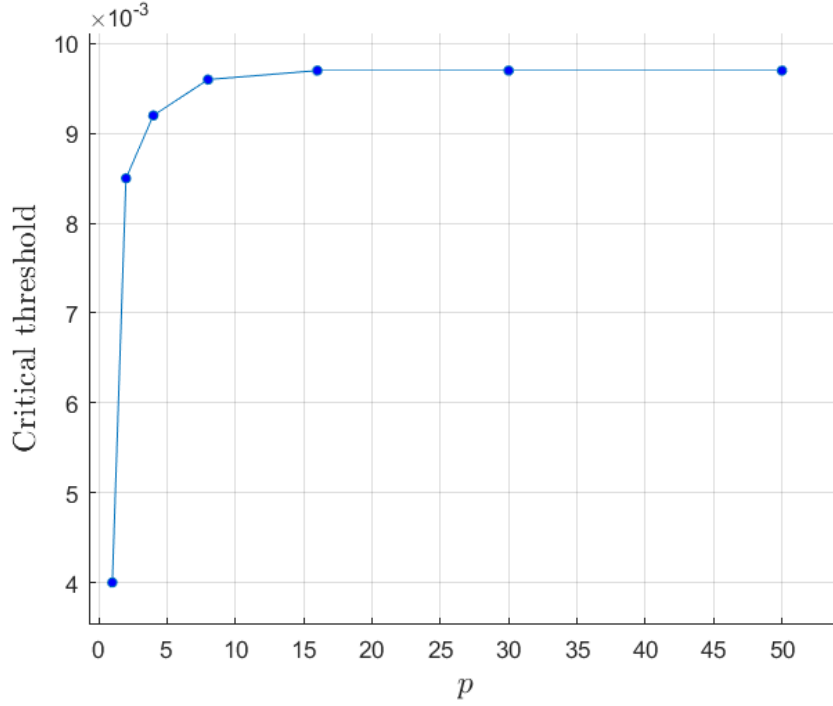
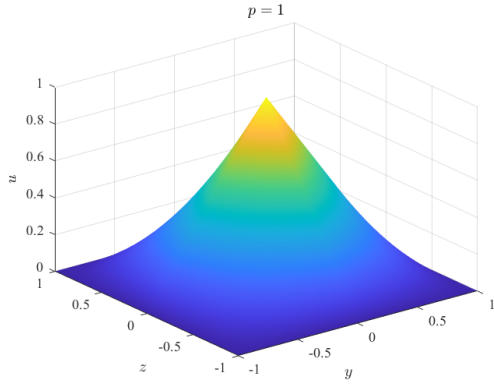


Figure 3.6: Critical threshold λ_c as a function of the inlet profile exponent p .

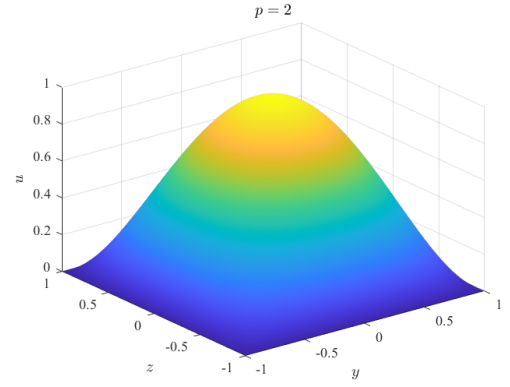
3.5. Discussion

The results show that the critical threshold λ_c increases with the exponent p and progressively seems to approach an asymptotic value. This behaviour can be explained by two concurrent effects. On the one hand, since the integral of the velocity profile over the inlet section is normalized to unity for any value of p , an increment in p produces a flatter distribution and a lower peak velocity at fixed λ : the same total momentum is spread more uniformly across the section, reducing the local intensity of the flow that impinges on the cylinder. On the other hand, increasing p improves the regularity of the inlet profile, reducing the magnitude of the velocity gradients in the transverse directions and yielding a more uniform momentum distribution, which tends to stabilise the stationary flow. The largest variation of λ_c is observed in the transition from $p = 1$ to $p = 2$. For $p = 1$ the inlet profile is piecewise linear and exhibits a sharp ridge along the centrelines $y = 0$ and $z = 0$, where the function is only C^0 and its first derivatives are discontinuous. The resulting strong, localised gradients introduce a destabilising mechanism that lowers the critical threshold considerably. Already for $p > 1$ the profile becomes smooth, the gradients are bounded and continuous, and the critical threshold rises sharply. For larger values of p , the inlet distribution progressively flattens towards a plug-flow shape, and λ_c tends to a finite asymptotic value, as the further reduction of both the peak velocity and

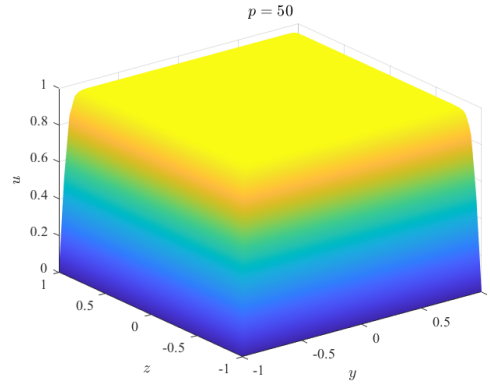
the gradients no longer produces a significant stabilising effect. A visual comparison of the inlet profiles for $p = 1$, $p = 2$ and $p = 50$ is reported in Figure 3.7, which clearly illustrates the transition from the sharp pyramidal shape at $p = 1$ to the smooth Poiseuille-like profile at $p = 2$, and finally to the nearly flat plug-flow distribution at $p = 50$.



(a) Pyramidal profile, $p = 1$.



(b) Poiseuille profile, $p = 2$.



(c) Plug-flow profile, $p = 50$.

Figure 3.7: Inlet velocity profiles for different values of the exponent p .

4 | Equilibrium positions in a Poiseuille flow

4.1. Introduction

As anticipated by its title, this chapter is devoted to the search for the equilibrium positions of the system, namely the configurations at which the lift force exerted by the fluid on the immersed beam vanishes. The analysis is carried out within a flow governed by the stationary Navier–Stokes equations, following the theoretical framework developed by Berchio, Bonheure, Galdi, Gazzola and Perotto [5]. In a planar channel with a symmetric body modelling the cross-section of a bridge deck, the authors prove that a single equilibrium position exists for sufficiently small flow velocities, whereas *multiple* equilibria appear once the velocity exceeds a suitable threshold. The same paper also discusses a remarkable historical application: Leonardo da Vinci’s ferry on the river Adda, which crosses the river without any propulsion system by exploiting the lift generated by the asymmetric flow around its hull. The mechanism that allows the ferry to drift across the stream is, in mathematical terms, the very same lift force whose zeros define the equilibrium positions investigated here. The primary goal of the present chapter is to trace numerically the evolution of the equilibrium positions as the flow parameter λ varies. At sufficiently low values of λ , a single equilibrium position exists, consistent with the theoretical predictions of [5]. As λ increases beyond the critical threshold, multiple equilibria emerge, and their positions shift progressively within the channel. Tracking this transition — from a unique equilibrium to a richer configuration — and characterising how the equilibrium positions depend on λ constitutes the central objective of the analysis carried out in this chapter.

4.2. Geometry and Numerical Setup

The geometric domain and the numerical strategy employed throughout this chapter are inherited from Chapter 3, to which the reader is referred for a detailed description. The

inlet velocity distribution is prescribed according to Equation (3.1) in the particular case $p = 2$, which corresponds to the classical parabolic Poiseuille profile in three dimensions, see Figure 3.7b. The associated normalisation constant c_p takes the value already computed in (3.7). The equilibrium positions have been computed for seven distinct values of the parameter λ , namely $\lambda = 0.000853$, 0.002631 , 0.003556 , 0.004409 , 0.008533 , 0.025600 , and 0.043378 , covering a range that spans from well below the critical threshold $\lambda_c = 0.0085$ up to values significantly exceeding it. Within this fixed flow configuration, the equilibrium positions are sought by translating the cylinder upward along the z -direction.

4.3. Methodology

The strategy adopted to detect the equilibrium positions of the system relies on a direct numerical investigation of the lift force, acting on the whole beam, as a function of the vertical position of the cylinder. Denoting by h the vertical displacement of the cylinder centre with respect to the centreline of the channel, the lift produced by the fluid on the beam becomes a function $L = L(h)$, and the equilibrium positions are identified as the values of h for which $L(h) = 0$. As mentioned in Section 4.1, the search is performed at the seven values of λ previously introduced. The function $L(h)$ has been reconstructed point by point through a sequence of independent steady-state simulations. Starting from the symmetric configuration $h = 0$, the cylinder has been translated upward in increments of $\Delta h = 0.1$ along the z -direction across the simulations. By symmetry of the channel and the inflow profile, the behaviour of $L(h)$ for negative displacements is the mirror image of the one reported here, and it is therefore sufficient to restrict the analysis to upward translations. At each step, the geometry of the immersed beam has been generated by applying a rigid translation to the surface mesh contained in the file `cylinder.stl` through the OpenFOAM utility `surfaceTransformPoints`, invoked with the option `"translate=(0 0 0.2)"`. The resulting displaced geometries are illustrated in Figure 4.1.



Figure 4.1: Cylinder at 0.2.

For each prescribed position of the cylinder, the simulation has been run and the corresponding value of the lift force has been recorded. The collection of pairs $(h, L(h))$

obtained in this way provides a discrete sampling of the lift curve from which the equilibrium positions are extracted. This procedure has been repeated for each of the seven values of λ considered in the present analysis.

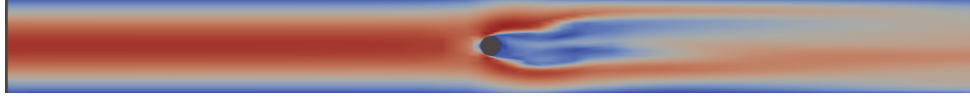


Figure 4.2: U profile around cylinder at 0.2.

4.4. Numerical Results

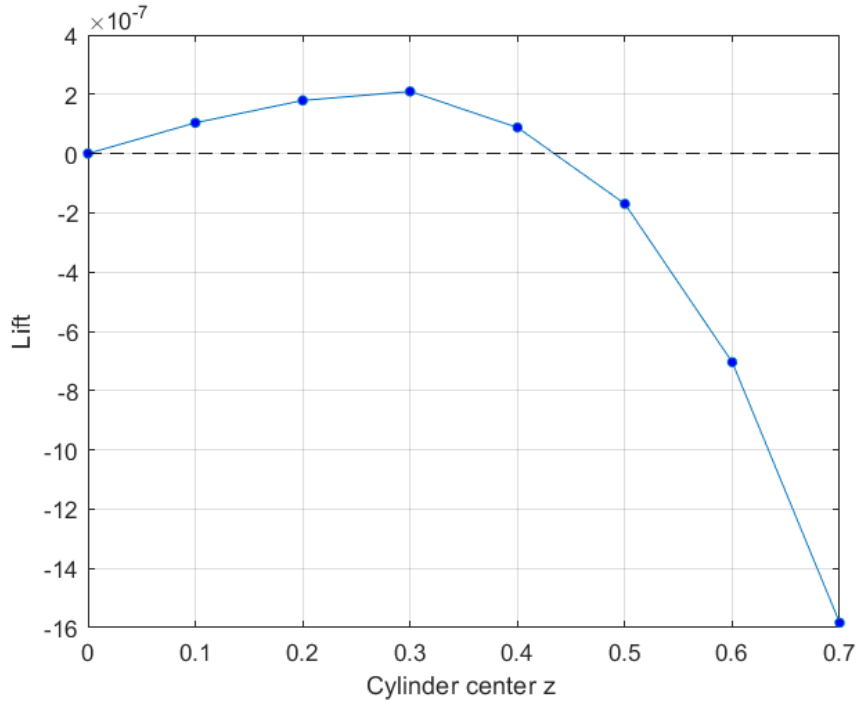
4.4.1. Equilibrium Position for given Velocity

The present section is devoted to the outcome of the numerical campaign, whose interpretation is postponed to Section 4.5. The cylinder has been displaced upward in increments of $\Delta h = 0.1$. For each prescribed position, the asymptotic value of the lift force exerted by the fluid on the cylinder has been recorded. The resulting data for $\lambda = 0.0085$ are summarised in Table 4.1.

Vertical Displacement h	Lift Force $L(h)$
0.0	-6.05×10^{-13}
0.1	1.04×10^{-7}
0.2	1.79×10^{-7}
0.3	2.10×10^{-7}
0.4	8.80×10^{-8}
0.5	-1.70×10^{-7}
0.6	-7.04×10^{-7}
0.7	-1.58×10^{-6}

Table 4.1: Displacement h and corresponding Lift

A more immediate visualisation of the lift behaviour is provided by plotting the values listed in Table 4.1 against the vertical displacement of the cylinder centre. The resulting curve $L(h)$ is reported in Figure 4.3.

Figure 4.3: Lift force L as a function of h .

4.4.2. Comparison for Different Values of λ

As anticipated, the equilibrium search has been carried out for seven distinct values of the bifurcation parameter λ . The results are summarised in Table 4.2 and illustrated in Figure 4.4.

λ	Equilibrium Position h^*	Number of Equilibria
0.000853	0	1
0.002631	0	1
0.003556	$0.1 < h^* < 0.2$	3
0.004409	$0.2 < h^* < 0.3$	3
0.008533	$0.4 < h^* < 0.5$	3
0.025600	$0.5 < h^* < 0.6$	3
0.043378	$0.6 < h^* < 0.7$	3

Table 4.2: Equilibrium positions and number of equilibria for each value of λ .

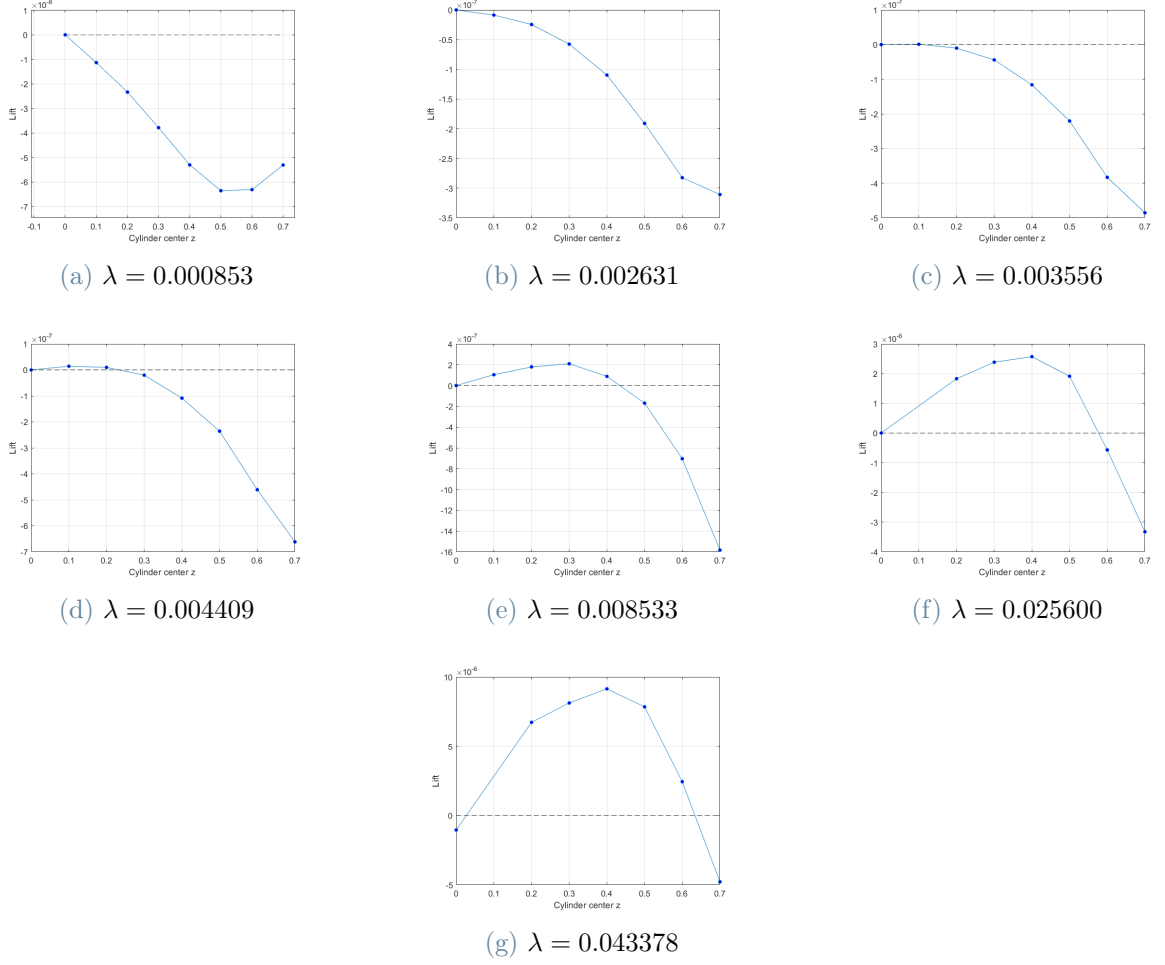


Figure 4.4: Lift force $L(h)$ as a function of the vertical displacement h for each value of λ .

4.5. Discussion

As can be seen from Figure 4.3, the lift curve $L(h)$ crosses the horizontal axis at two distinct points, thus revealing the existence of three equilibrium configurations of the system: the trivial one at $h = 0$, a non-trivial equilibrium located at $h \approx 0.45$, and, by symmetry of the geometry and of the inlet profile, a third equilibrium at $h \approx -0.45$. The behaviour of $L(h)$ in the neighbourhood of these points provides a direct insight into their stability. For values of h slightly greater than zero, the lift force takes positive values and therefore pushes the cylinder further upward, away from $h = 0$ and towards the equilibrium located at $h \approx 0.45$. Beyond this point, the lift changes sign: it becomes negative and acts as a restoring force, bringing the cylinder back to $h \approx 0.45$. The trivial equilibrium $h = 0$ is therefore unstable, whereas the two non-trivial equilibria at

$h \approx \pm 0.45$ are stable. For very low values of λ , a single equilibrium point exists at the centre of the channel, and any displacement from it produces a negative lift force that pushes the cylinder back towards it. As λ increases, two equilibrium positions appear: the central one becomes unstable, while a second stable equilibrium emerges, whose structure is analogous to the one discussed above for $\lambda = 0.0085$, the only difference being its position along z , which shifts upward as λ grows, as observed in Figure 4.4.

4.5.1. Comparison with the Theoretical Framework

The numerical evidence reported above is consistent with the theoretical results that motivate the present analysis. Bianca, Bocchi and Gazzola [6] prove, in their Theorem 4.1, the existence of $\beta_0 > 0$ and $\lambda_0 > 0$ such that, for every inflow magnitude $\lambda \in [0, \lambda_0)$, the three-dimensional fluid–structure interaction problem governed by the stationary Navier–Stokes equations admits a *unique* equilibrium configuration in a neighbourhood of the symmetric position, provided that the inflow magnitude λ remains below a suitable threshold λ_0 . As can be seen from Figure 4.4 and Table 4.2, for the two lowest values of λ considered, namely $\lambda = 0.000853$ and $\lambda = 0.002631$, a single equilibrium point is found at the centre of the channel, indicating that the system is still operating in the regime $\lambda < \lambda_0$ where uniqueness is guaranteed. As λ increases further, a second equilibrium point appears, signalling that the threshold λ_0 has been exceeded and the system has left the uniqueness regime. This transition is in full agreement with the theoretical predictions of [6].

The same phenomenon admits a precise interpretation in light of the work of Berchio, Bonheure, Galdi, Gazzola and Perotto [5], who study the analogous two-dimensional problem in a planar channel. They prove that, beyond a critical value of the inflow magnitude, the symmetric equilibrium loses uniqueness and a pair of asymmetric equilibrium configurations emerges. The three-equilibrium structure observed in the present simulations is the natural three-dimensional counterpart of the same multiplicity phenomenon described in [5]. Their Theorem 4.3 proves the existence and uniqueness of the symmetric equilibrium configuration for sufficiently small Reynolds numbers, and their numerical analysis in Section 6 (Numerical Result 6.2) shows that, beyond a critical value of the inflow magnitude, the symmetric equilibrium loses its uniqueness and a pair of symmetry-breaking equilibrium configurations emerges, with the trivial equilibrium becoming unstable and the two non-trivial equilibria being stable. What is observed in the present three-dimensional simulations is precisely the three-dimensional translation of the multiplicity phenomenon described in the two-dimensional framework of [5]: the loss of stability of the central equilibrium and the emergence of a pair of stable non-trivial configurations mirror exactly the theoretical picture established therein.

4.5.2. A Similar Model: Leonardo da Vinci's Ferry

A remarkable physical realisation of the same mechanism is provided by Leonardo da Vinci's ferry on the river Adda, to which Section 7 of [5] is devoted. The ferry crosses the river without any propulsion system: a steel cable is positioned orthogonally to the flow, the hull is hooked to the cable, and the operator orients the ferry through the helm so that the keel forms a non-zero angle of attack with the current. The asymmetric flow around the inclined hull then generates a transverse lift force that drives the ferry across the river. In this case, however, the target is to maximise the Lift as for airfoils .

The link with the present work lies in the existence of non-trivial configurations at which the fluid exerts a non-zero lift on the body. The same phenomenon that produces the lateral equilibria in the beam model studied here is what allows the ferry to move sideways: a body placed asymmetrically in a channel flow experiences a transverse force. The cable prevents the ferry from being dragged downstream and constrains its motion to the direction orthogonal to the current; the lift generated by the inclined hull then acts along this allowed direction and pushes the ferry from one bank to the other. By varying the angle of attack through the helm, the operator controls the magnitude and the sign of the lift, thereby selecting the direction of crossing and adjusting the speed of the ferry along its trajectory. In the bridge, this transverse force represents an undesired loss of symmetry; in Leonardo's design, it is the very working principle of the device.



Figure 4.5: The Leonardo da Vinci ferry on the river Adda.

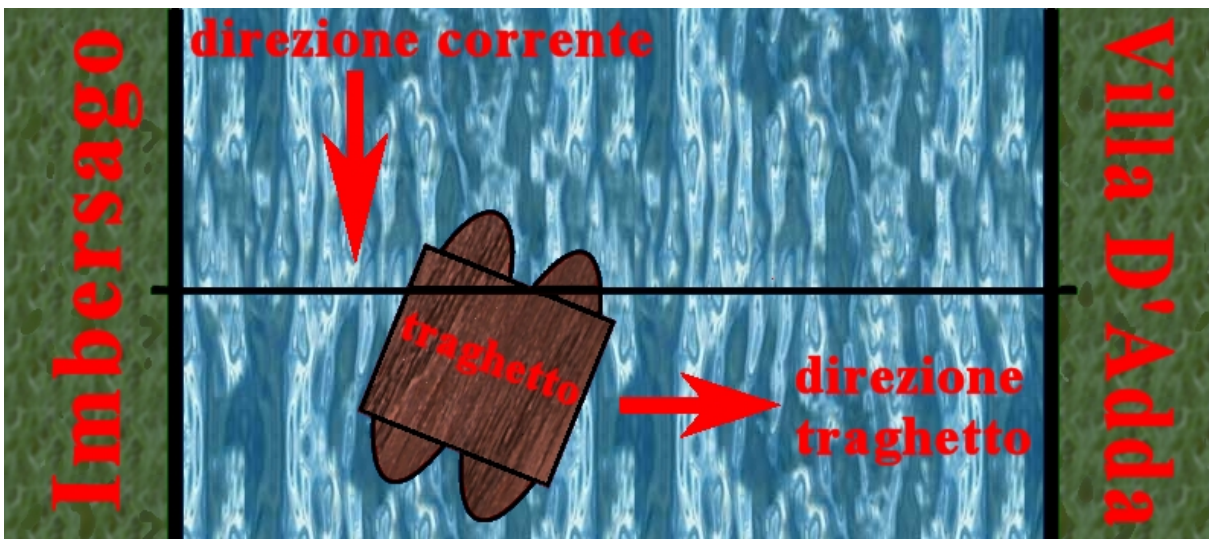


Figure 4.6: The Leonardo da Vinci ferry mechanism.

5 | Flow Stability of Elliptical Cross-Sections

5.1. Introduction

In this chapter, the focus shifts with respect to what has been done so far. Up to this point, the analysis always concerned the stability of the flow around a beam with a fixed geometry, namely a circular cylinder. The objective here is slightly different: the aim is to study how the critical threshold λ_c , beyond which the flow loses its steady character and begins to oscillate, changes as the cross-section of the beam varies. Several elliptical cross-sections are tested. Two separate parametric studies are carried out, each one imposing a different geometric constraint on the family of ellipses. In the first, the perimeter of each ellipse equals the circumference of the reference circle of radius $R = 0.2$. In the second, the cross-sectional area is kept equal to that of the same reference circle, $A_0 = \pi R^2$. This study on flow stability has an important engineering application. In bridge construction, cost constraints are often tied to the amount of concrete and materials available. The constant-perimeter case can be seen as a hollow bridge, where the perimeter represents the quantity of material used for the outer shell. The constant-area case corresponds instead to a solid bridge, where the area is proportional to the volume of concrete per unit length. The two families are then compared against each other.

5.2. Geometry and Numerical Setup

The only element that changes with respect to the previous chapter is the cross-sectional geometry of the obstacle. The elliptical cross-sections are defined by two semi-axes: a , parallel to the streamwise direction (x), and b , parallel to the transverse direction (z). Their perimeter is approximated by the formula proposed by Ramanujan [16]

$$P \approx \pi \left[(a + b) + \frac{3(a - b)^2}{10(a + b) + \sqrt{4(a + b)^2 - 3(a - b)^2}} \right]. \quad (5.1)$$

Using Equation (5.1) and varying a and b at fixed aspect ratio $k = a/b$, the semi-axes are determined by solving $P(a, b) = P_0$, where $P_0 = 2\pi R \approx 1.2566$ is the circumference of the reference circle of radius $R = 0.2$. The resulting geometries are listed in Table 5.1.

$k = a/b$	a	b
0.24	0.07061	0.29423
0.62	0.15100	0.24355
1.00	0.20000	0.20000
1.46	0.23534	0.16119
2.00	0.25941	0.12970
2.25	0.26696	0.11865
2.50	0.27303	0.10921
3.17	0.28447	0.08974
4.61	0.29705	0.06444

Table 5.1: Ellipses at constant perimeter.

Once the aspect ratios were established, the second family was built by keeping the cross-sectional area equal to that of the reference circle, $A_0 = \pi R^2 \approx 0.1257$. Imposing $\pi ab = \pi R^2$ together with $a = kb$ gives $b = R/\sqrt{k}$ and $a = R\sqrt{k}$. The resulting geometries are listed in Table 5.2.

$k = a/b$	a	b
0.24	0.09798	0.40825
0.62	0.15748	0.25400
1.00	0.20000	0.20000
1.46	0.24166	0.16552
2.00	0.28284	0.14142
2.25	0.30000	0.13333
2.50	0.31623	0.12649
3.17	0.35609	0.11233
4.61	0.42942	0.09315

Table 5.2: Ellipses at constant area.

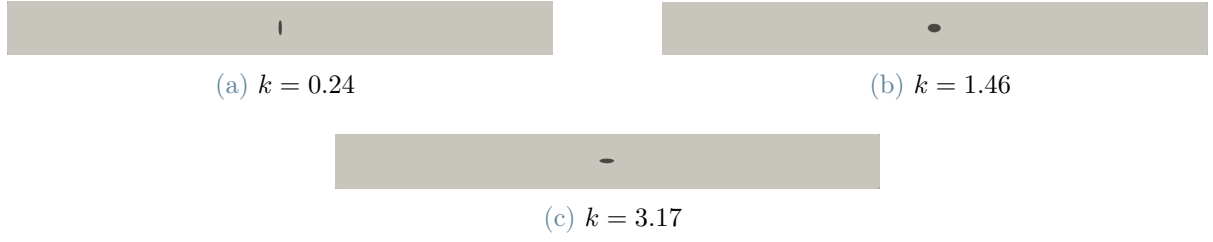


Figure 5.1: Comparison of three elliptical cross-sections with the same perimeter at aspect ratios $k = 0.24$, $k = 1.46$, and $k = 3.17$.

In particular, the inlet velocity profile remains the same as in the previous chapter, namely the Poiseuille inlet flow. For each elliptical shape, the maximum velocity is varied in order to identify the critical threshold λ_c . All other settings remain unchanged and can be found in Section 3.2.

5.3. Methodology and Numerical Results

As in the previous chapter, for each elliptical shape λ is increased and the largest value beyond which the oscillation starts, called λ_c , is sought. The process is identical to what was done in Section 3.3 and will not be repeated here.

In this section the results obtained from the simulations are presented. In particular, the critical threshold λ_c is shown as a function of the aspect ratio $k = a/b$, both for the constant-perimeter and the constant-area families.

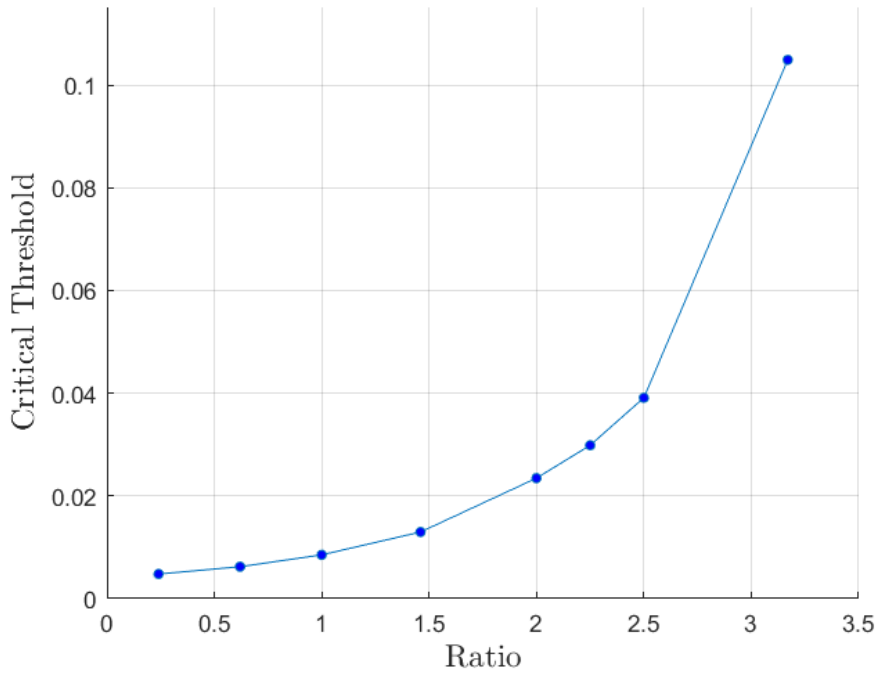
5.3.1. Constant Perimeter

The critical thresholds for the constant-perimeter case are summarized in Table 5.3.

Aspect Ratio $k = a/b$	Critical Threshold λ_c
0.24	0.0048
0.62	0.0062
1.00	0.0085
1.46	0.0130
2.00	0.0235
2.25	0.0299
2.50	0.0391
3.17	0.1049

Table 5.3: Critical values of λ_c .

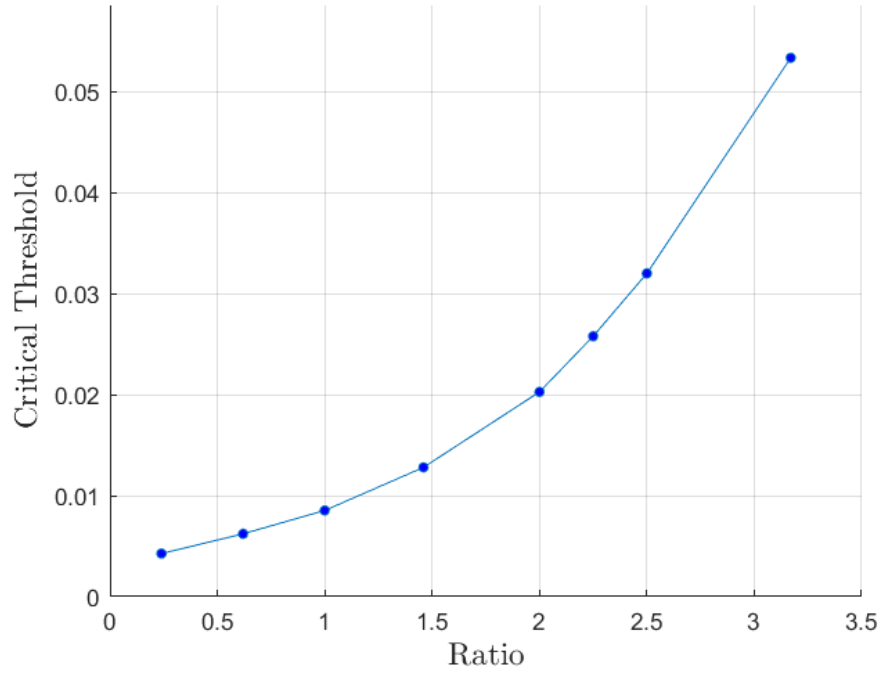
For a better visualization, the values are also reported in Figure 5.2.

Figure 5.2: Critical threshold λ_c as a function of the aspect ratio $k = a/b$.

5.3.2. Constant Area

As in the previous subsection, the same analysis is carried out for the constant-area family. The critical thresholds are listed in Table 5.4 and plotted in Figure 5.3.

Aspect Ratio $k = a/b$	Critical Threshold λ_c
0.24	0.0043
0.62	0.0062
1.00	0.0085
1.46	0.0128
2.00	0.0203
2.25	0.0258
2.50	0.0320
3.17	0.0533

Table 5.4: Critical values of λ_c .Figure 5.3: Critical threshold λ_c as a function of the aspect ratio $k = a/b$.

5.3.3. Comparison

In this subsection it is compared how λ_c evolves differently in the two cases. Starting from the circular cylinder, the critical threshold λ_c is compared when the perimeter is kept constant and when the area is kept constant. The comparison is shown in Figure 5.4.

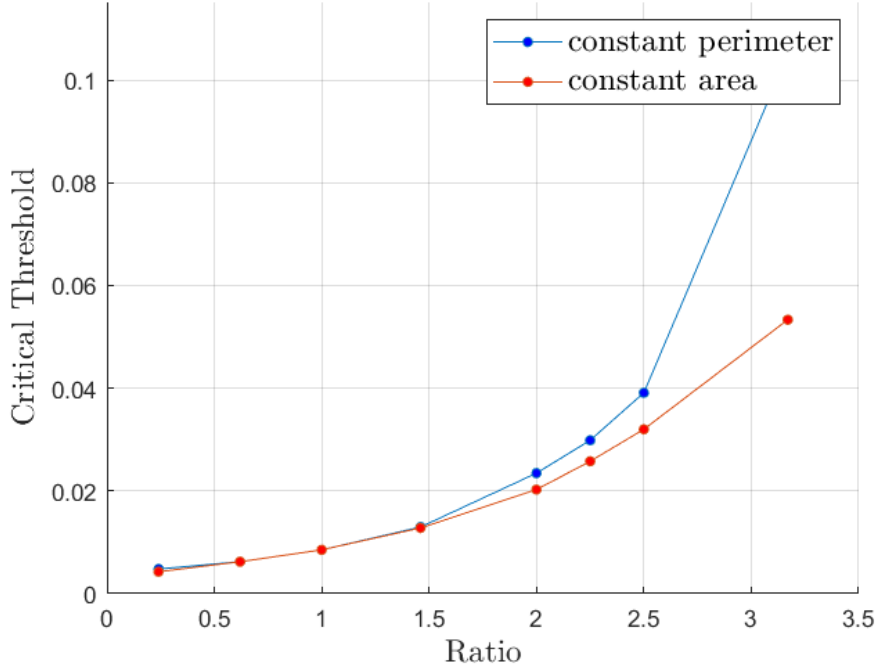


Figure 5.4: Comparison of λ_c vs ratio for constant perimeter and constant area ellipses.

5.4. Discussion

As observed in the tables and figures, λ_c increases monotonically with the aspect ratio $k = a/b$ for both families. This trend is physically consistent: as k increases, the semi-axis a , aligned with the streamwise direction, grows relative to b , which correspondingly decreases, producing a progressively more streamlined cross-section. A more elongated shape presents a smaller frontal area to the incoming flow, delaying the onset of vortex shedding, as illustrated in Figure 5.1. While this result meets physical expectations, it provides a useful confirmation of the robustness and consistency of the numerical setup. The more relevant comparison concerns the evolution of λ_c between the two families at equal aspect ratio. Starting from the reference circular cylinder at $k = 1$, where the two families coincide by construction, moving along the aspect ratio axis under either constraint leads to two distinct stability curves. From an engineering standpoint, the two constraints carry a concrete physical interpretation: constant perimeter corresponds to a hollow bridge deck built with a fixed amount of material for the outer shell, whereas constant area corresponds to a solid cross-section with a fixed volume of concrete per unit length. In both cases, the gap between the two curves grows with k , remaining small at low aspect ratios and becoming increasingly pronounced at higher ones, as reported in Table 5.3 and Table 5.4. The constant-perimeter family consistently yields higher

values of λ_c than the constant-area family at the same aspect ratio. The underlying reason follows directly from the isoperimetric inequality: among all planar figures with a given perimeter, the circle encloses the maximum area. It follows that, for any $k \neq 1$, the constant-perimeter ellipse has a strictly smaller cross-sectional area than πR^2 , which is precisely the area of the constant-area ellipse. At equal aspect ratio, the constant-perimeter beam therefore occupies less of the channel cross-section, and the corresponding reduction in blockage consistently produces a higher critical threshold λ_c . Nevertheless, the geometric argument provides a rigorous and self-contained explanation for why the constant-perimeter family is systematically more stable.

From an engineering perspective, this result suggests that a hollow bridge deck, in addition to requiring less material and being lighter than a solid one, also benefits from greater aerodynamic stability. While this observation remains qualitative and would need to be verified against more realistic structural geometries, it points to a potentially favorable combination of structural and aerodynamic properties worth exploring in future work.

6 | Conclusions and future developments

To complement the theoretical results established in [6], a beam of several shapes and in several positions was studied; unlike the configuration analysed in [6], it remains rigid throughout each simulation. The beam is placed inside a wind tunnel and the flow is governed by the stationary Navier–Stokes equations, a setting that, in a broad sense, can model the response of a bridge to wind loading. The quantity that made the stability analysis of the flow around the beam possible is the lift force acting on the whole beam. Tracking its behaviour made it possible to identify the critical value λ_c , which is proportional to the Reynolds number since it scales with the maximum inlet velocity, together with the corresponding stability threshold for each beam shape and each inlet profile.

For the cylinder with circular cross-section, stability, and therefore λ_c , was found to increase with the exponent p in (3.3), as shown in Figure 3.6. Two mechanisms explain this trend: as p grows the inlet profile becomes fuller and flatter across the section, so that the velocity gradients are smaller, and, at fixed λ , the maximum velocity decreases as p increases. For the Poiseuille inlet profile it was also observed that, as λ increases beyond a certain threshold, the system departs from the single symmetric equilibrium and develops several equilibrium positions, defined as the heights h of the cylinder at which the lift force vanishes. As anticipated, these results clearly and directly complement the propositions 2.1, 2.2 in [6] and theoretical results in [5], which state that at low λ a single stable equilibrium exists, whereas for increasing λ the number of equilibria grows: two new stable equilibria appear and the previous one loses its stability, in what amounts to a pitchfork bifurcation. After the circular cylinder, the analysis turned to a cylinder with elliptical cross-section. Starting from the circular case, two families of ellipses were built: one in which each ellipse has the same area as the cylinder, and one in which each ellipse has the same perimeter. With the bridge application in mind, where, on a much larger scale, the two families may be regarded as solid and hollow bridge decks, the stability of the two families was compared, and the equal-perimeter family was found to be the more stable at equal ratio between the semi-axes, as can be seen in Figure 5.4.

A natural way to continue this work is to extend the stability analysis beyond the circular and elliptical cross-sections, testing new and more aerodynamic shapes, such as an airfoil, as well as less aerodynamic ones, such as a square or a rectangle, while moving progressively closer to the shape of a real bridge and validating the numerical results obtained with OpenFOAM against experimental measurements from actual wind-tunnel tests. A further and particularly interesting direction is to simulate numerically, again with experimental validation in the wind tunnel, beams whose shape is increasingly close to that of a bridge, in order to model reality as faithfully as possible. This would mean, above all, coupling the Navier–Stokes equations with a genuine beam equation, so that the beam can oscillate and deform in response to the surrounding flow, and then studying the equilibria and the stability of this fluid–structure interaction problem, which would come ever closer both to the model of [6] and to reality.

Bibliography

- [1] B. Akesson. *Understanding bridge collapses*. CRC Press, Taylor & Francis Group, London, 2008.
- [2] A. Arena and W. Lacarbonara. Nonlinear parametric modeling of suspension bridges under aeroelastic forces: torsional divergence and flutter. *Nonlinear Dynamics*, 70: 2487–2510, 2012.
- [3] G. Arioli and F. Gazzola. A new mathematical explanation of what triggered the catastrophic torsional mode of the Tacoma Narrows Bridge collapse. *Appl. Math. Modelling*, 39:901–912, 2015.
- [4] E. Berchio, A. Ferrero, and F. Gazzola. Structural instability of nonlinear plates modelling suspension bridges: mathematical answers to some long-standing questions. *Nonlinear Analysis: Real World Applications*, 28:91–125, 2015.
- [5] E. Berchio, D. Bonheure, G. P. Galdi, F. Gazzola, and S. Perotto. Equilibrium configurations of a symmetric body immersed in a stationary Navier–Stokes flow in a planar channel. *SIAM Journal on Mathematical Analysis*, 56(3):3759–3801, 2024. doi: 10.1137/23M1568752.
- [6] V. Bianca, E. Bocchi, and F. Gazzola. Equilibrium configurations of a 3D fluid–beam interaction problem. *Applied Mathematics & Optimization*, 92(50), 2025. doi: 10.1007/s00245-025-10317-8.
- [7] F. Bleich. Dynamic instability of truss-stiffened suspension bridges under wind action. *Proceedings ASCE*, 74:1269–1314, 1948.
- [8] F. Bleich, C. McCullough, R. Rosecrans, and G. Vincent. *The mathematical theory of vibration in suspension bridges*. U.S. Dept. of Commerce, Bureau of Public Roads, Washington D.C., 1950.
- [9] M. Como. *Stabilità aerodinamica dei ponti di grande luce. Introduzione all’ingegneria strutturale*. E. Giangreco, UTET, Torino, 2002.

- [10] M. Como, S. D. Ferraro, and A. Grimaldi. A parametric analysis of the flutter instability for long span suspension bridges. *Wind and Structures*, 8:1–12, 2005.
- [11] F. Gazzola. *Mathematical models for suspension bridges*. MS&A Vol.15. Springer, 2015.
- [12] A. Lazer and P. McKenna. Large-amplitude periodic oscillations in suspension bridges: some new connections with nonlinear analysis. *SIAM Rev.*, 32:537–578, 1990.
- [13] P. McKenna and W. Walter. Nonlinear oscillations in a suspension bridge. *Arch. Rat. Mech. Anal.*, 98:167–177, 1987.
- [14] C. Navier. *Mémoire sur les ponts suspendus*. Imprimerie Royale, Paris, 1823.
- [15] B. Pittel and V. Yakubovich. A mathematical analysis of the stability of suspension bridges based on the example of the Tacoma Bridge (Russian). *Vestnik Leningrad Univ.*, 24:80–91, 1969.
- [16] S. Ramanujan. Modular equations and approximations to π . *Quarterly Journal of Mathematics*, 45:350–372, 1914.
- [17] Y. Rocard. *Dynamic instability: automobiles, aircraft, suspension bridges*. Crosby Lockwood, London, 1957.
- [18] R. Scanlan and J. Tomko. Airfoil and bridge deck flutter derivatives. *J. Eng. Mech.*, 97:1717–1737, 1971.
- [19] R. Scott. *In the wake of Tacoma. Suspension bridges and the quest for aerodynamic stability*. ASCE Press, 2001.
- [20] D. Steinman. Rigidity and aerodynamic stability of suspension bridges. *Proceedings ASCE*, 69:1361–1397, 1943.
- [21] D. Steinman. Design of bridges against wind: IV, Aerodynamic instability - prevention and cure. *Civil Engineers ASCE*, pages 20–23, 1946.
- [22] S. Timoshenko. Theory of suspension bridges - Part I. *Journal of the Franklin Institute*, 235:213–238, 1943.
- [23] S. Timoshenko. Theory of suspension bridges - Part II. *Journal of the Franklin Institute*, 235:327–349, 1943.

A | Geometry and Numerical Setup in OpenFOAM

This appendix collects the main configuration files and numerical settings adopted to set up and solve the bridge flow problems discussed.

A.1. Initial and Boundary Conditions

Velocity field: 0/U

The first thing to set up is the velocity field in 0/U. Since in Chapter 3 the goal was to study the critical threshold as a function of p , the inlet was defined with a coded power-law profile:

Listing A.1: 0/U

```

1  flowVelocity      0.0048;
2  internalField     uniform ($flowVelocity 0 0);
3  boundaryField
4  {
5      inlet
6      {
7          type        codedFixedValue;
8          value        uniform ($flowVelocity 0 0);
9          name        parabolicInlet;
10         code
11         #{
12             scalar Umax = 0.0048;
13             const fvPatch& patch = this->patch();
14             const vectorField& Cf = patch.Cf();
15             vectorField& field = *this;
16             forAll(Cf, faceI)
17             {
18                 scalar y = Cf[faceI].y();
19                 scalar z = Cf[faceI].z();

```

```

20         scalar fy = 1.0 - (y * y);
21         scalar fz = 1.0 - (z * z);
22         if (fy < 0) fy = 0;
23         if (fz < 0) fz = 0;
24         field[faceI] = vector(Umax * fy * fz, 0, 0);
25     }
26     #};
27 }
28 outlet
29 {
30     type            zeroGradient;
31 }
32 walls
33 {
34     type            noSlip;
35 }
36 "(.*) "
37 {
38     type            noSlip;
39 }
40 }

```

For each simulation in Chapter 3, $U_{\max}$ and `flowVelocity` were updated for each value of λ , while the exponent p was changed by replacing $y*y$ and $z*z$ with `pow(fabs(y),p)` and `pow(fabs(z),p)`. In Chapter 4 and Chapter 5, the inlet profile is kept as the Poiseuille profile throughout; $U_{\max}$ is the only quantity that changes between runs, as it is increased to locate λ_c in Chapter 5 and swept across different values to track the equilibrium positions in Chapter 4. In every case the quantity set in $U_{\max}$ is λc_p , so the stable values obtained from the simulations were divided by c_p in MATLAB to recover the corresponding values of λ . No-slip was set on all walls. At the outlet, `zeroGradient` means the normal derivative of velocity is zero, so the flow can leave the domain without any imposed constraint on its value.

Pressure field: 0/p

Listing A.2: 0/p

```

1  internalField      uniform 0;
2  boundaryField
3  {
4      inlet
5      {
6          type            zeroGradient;

```

```

7      }
8      outlet
9      {
10         type          fixedValue;
11         value          uniform 0;
12     }
13     walls
14     {
15         type          zeroGradient;
16     }
17     "(.*)"
18     {
19         type          zeroGradient;
20     }
21 }

```

Pressure is fixed to zero at the outlet, which acts as the reference value for the whole domain. At the inlet and on the walls `zeroGradient` is used, so pressure there is not imposed but follows from the solution. This is the standard setup for incompressible solvers: velocity is prescribed at the inlet and pressure is only anchored at the outlet.

A.2. Mesh and Domain

Background mesh: `system/blockMeshDict`

The domain is defined in `system/blockMeshDict`. The eight corners of the channel $[-10, 10] \times [-1, 1] \times [-1, 1]$ are listed as vertices, and the block is divided into a $60 \times 30 \times 30$ structured mesh. All four walls have a no-slip condition, inlet and outlet are plain patches.

Listing A.3: `system/blockMeshDict`

```

1  scale    1;
2
3  vertices
4  (
5      (-10 -1 -1) // 0: inlet bottom left
6      ( 10 -1 -1) // 1: outlet bottom left
7      ( 10  1 -1) // 2: outlet top left
8      (-10  1 -1) // 3: inlet top left
9      (-10 -1  1) // 4: inlet bottom right
10     ( 10 -1  1) // 5: outlet bottom right
11     ( 10  1  1) // 6: outlet top right
12     (-10  1  1) // 7: inlet top right

```

```

13 );
14
15 blocks
16 (
17     hex (0 1 2 3 4 5 6 7) (60 30 30) simpleGrading (1 1 1)
18 );
19
20 boundary
21 (
22     inlet
23     {
24         type patch;
25         faces ( (0 4 7 3) );
26     }
27     outlet
28     {
29         type patch;
30         faces ( (1 2 6 5) );
31     }
32     walls
33     {
34         type wall;
35         faces
36         (
37             (0 1 5 4) // bottom
38             (3 7 6 2) // top
39             (0 3 2 1) // side 1
40             (4 5 6 7) // side 2
41         );
42     }
43 );

```

Cylinder geometry and mesh refinement: `system/snappyHexMeshDict`

The cylinder, radius 0.2, spanning $y \in [-1, 1]$, was imported as `constant/triSurface/cylinder.stl`. The `snappyHexMesh` utility, together with `system/surfaceFeatureExtractDict`, was used to cut the cylinder into the background mesh and refine the cells around it. The geometry block specifies the surface file and a refinement box in the wake region, as shown in Figure 3.3:

Listing A.4: `snappyHexMeshDict` – geometry and refinement region

```

1 geometry
2 {

```

```

3     cylinder.stl
4     {
5         type triSurfaceMesh;
6         name cylinder;
7     }
8     refinementBox
9     {
10        type searchableBox;
11        min (-0.5 -10 -2);
12        max ( 2.0  10  2);
13    }
14 }

```

The refinement level on the cylinder surface is set to (4,4), which gives enough resolution to capture the boundary layer and the curvature of the surface:

Listing A.5: snappyHexMeshDict – surface refinement level

```

1  refinementSurfaces
2  {
3      cylinder
4      {
5          level (4 4);
6          patchInfo
7          {
8              type wall;
9              inGroups (walls);
10         }
11     }
12 }

```

Cylinder displacement for Chapter 4

For Chapter 4, the cylinder was shifted along the z -axis to find the equilibrium positions. This was done from the terminal with `surfaceTransformPoints`. For instance, to move the center to $z = 0.2$:

Listing A.6: Terminal – cylinder translation

```

1  surfaceTransformPoints -translate "(0 0 0.2)" cylinder.stl target.stl

```

The positions tested are shown in Figure 4.1.

Elliptical cross-section geometry for Chapter 5

As already discussed in Chapter 5, the only element that changes with respect to the previous chapters is the cross-sectional geometry of the obstacle. For each aspect ratio $k = a/b$, the file `constant/triSurface/cylinder.stl` is replaced by the corresponding `constant/triSurface/ellipse.stl`, generated specifically for that case. All references to `cylinder.stl` in the configuration files — in particular in `system/snappyHexMeshDict` and `system/controlDict` — are updated accordingly to point to `ellipse.stl`.

A.3. Numerical Setup

Discretization schemes: `system/fvSchemes`

The numerical schemes used in the simulations are set in `system/fvSchemes`:

Listing A.7: `system/fvSchemes`

```

1  ddtSchemes
2  {
3      default          steadyState;
4  }
5  gradSchemes
6  {
7      default          Gauss linear;
8      grad(U)          cellLimited Gauss linear 1;
9  }
10 divSchemes
11 {
12     default          none;
13     div(phi,U)        bounded Gauss linearUpwindV grad(U);
14     turbulence        bounded Gauss upwind;
15     div(phi,k)        $turbulence;
16     div(phi,omega)    $turbulence;
17     div((nuEff*dev2(T(grad(U)))) Gauss linear;
18 }
19 laplacianSchemes
20 {
21     default          Gauss linear corrected;
22 }
23 interpolationSchemes
24 {
25     default          linear;
26 }
```

```

27  snGradSchemes
28  {
29      default          corrected;
30  }
31  wallDist
32  {
33      method          meshWave;
34  }

```

The most important setting is `steadyState` in `ddtSchemes`, which follows directly from the steady-state Navier-Stokes formulation used in both chapters and matches the `simpleFoam` solver. The remaining schemes use standard Gauss linear discretization for gradients and laplacian terms, and a bounded upwind scheme for the convective term, which is a common choice for incompressible flows of this kind.

Linear solvers and relaxation: `system/fvSolution`

Listing A.8: `system/fvSolution`

```

1  solvers
2  {
3      p
4      {
5          solver          GAMG;
6          smoother        GaussSeidel;
7          tolerance       1e-7;
8          relTol          0.01;
9      }
10     U
11     {
12         solver          smoothSolver;
13         smoother        GaussSeidel;
14         tolerance       1e-8;
15         relTol          0.1;
16         nSweeps         1;
17     }
18     k
19     {
20         solver          smoothSolver;
21         smoother        GaussSeidel;
22         tolerance       1e-8;
23         relTol          0.1;
24         nSweeps         1;
25     }

```

```

26     omega
27     {
28         solver            smoothSolver;
29         smoother          GaussSeidel;
30         tolerance         1e-8;
31         relTol            0.1;
32         nSweeps           1;
33     }
34 }
35 SIMPLE
36 {
37     nNonOrthogonalCorrectors 0;
38     consistent yes;
39 }
40 relaxationFactors
41 {
42     equations
43     {
44         U                0.9;
45         k                 0.7;
46         omega             0.7;
47     }
48 }

```

Pressure is solved with GAMG while velocity and turbulence quantities use a `smoothSolver`, both with Gauss-Seidel smoothing. The relaxation factors and tolerances were tuned to keep the iterations stable across all the simulations.

A.4. Simulation Control and Run-time Settings

Solver control: `system/controlDict`

The run-time settings are defined in `system/controlDict`:

Listing A.9: `system/controlDict`

```

1  application      simpleFoam;
2
3  startFrom        startTime;
4  startTime        0;
5  stopAt           endTime;
6  endTime          2000;
7  deltaT           1;
8

```

```

9  writeControl    timeStep;
10 writeInterval   100;
11 purgeWrite     0;
12 writeFormat     binary;
13
14 functions
15 {
16     forces
17     {
18         type          forceCoeffs;
19         libs           ("libforces.so");
20         writeControl   timeStep;
21         writeInterval  1;
22         patches        (cylinder);
23         rho            rhoInf;
24         rhoInf         1.225;
25         CofR           (0 0 0);
26         liftDir         (0 0 1);
27         dragDir         (1 0 0);
28         pitchAxis      (0 1 0);
29         magUInf        0.0049;
30         lRef           0.4;
31         Aref           0.8;
32     }
33     pressure_point
34     {
35         type          probes;
36         libs           (sampling);
37         writeControl   timeStep;
38         writeInterval  1;
39         fields          (p);
40         probeLocations  ((0.5 0 0.1));
41     }
42 }

```

The end time is set to 2000 iterations, which was enough to reach a fully developed flow and let the solution stabilize, given the low Reynolds numbers used in the simulations. Results are written every 100 steps to limit the output size.

The most important part of this file is the **functions** block, on which the whole analysis is based. The first function, **forces**, computes the force coefficients on the cylinder surface at every time step. The lift direction is set to the positive z -axis, and **magUInf** must be updated each time $U_{\max}$ is changed in $0/U$. From these data the lift force history is

extracted and used to detect the onset of oscillations.

The second function, `pressure_point`, records the pressure at the probe location (0.5, 0, 0.1) at every time step. This gives a direct way to spot oscillations, since a steady pressure corresponds to a stable flow while persistent fluctuations indicate that the critical threshold has been crossed.

Both functions write output files that were then processed in MATLAB with the scripts described in the following section.

Modifications for Chapter 5

As far as Chapter 5 is concerned, the only change in `controlDict` concerns the force computation. Since the reference area A_{ref} changes with the aspect ratio k , using `forceCoeffs` would have required updating `Aref` for every ellipse. To avoid this, the `forces` function was used instead, which computes the lift force directly without any normalisation:

Listing A.10: `controlDict` – forces for Chapter 5

```

1  forces
2  {
3      type            forces;
4      libs            ("libforces.so");
5      writeControl     timeStep;
6      writeInterval    1;
7      patches         (ellipse);
8      rho              rhoInf;
9      rhoInf           1.225;
10     CofR             (0 0 0);
11 }
```

This is purely a matter of convenience, since the result is identical to what would be obtained by multiplying the lift coefficient by $\frac{1}{2}\rho U^2 A_{\text{ref}}$.

B | MATLAB Scripts

B.1. Analysis of Wake Oscillations

As mentioned in the previous chapter, two MATLAB scripts were used to detect the presence of oscillations and to study the flow behavior. The first one reads the `coefficient.dat` file produced by OpenFOAM and extracts the lift coefficient C_L . In Chapter 4, it was also used to compute the lift force directly. The value of $U_{\max}$ was updated for each simulation accordingly.

Listing B.1: MATLAB – lift force analysis from `coefficient.dat`

```

1 data = readmatrix('coefficient.dat', 'FileType', 'text');
2 data = data(~any(isnan(data), 2), :);
3 cl = data(:, 5);
4 passi = 1:length(cl);
5 rho = 1.225;
6 D = 0.4;
7 Aref = 0.8;
8 U_max = 0.0048;
9
10 % Analysis of the last 100 steps
11 ultimi_cl = cl(end-100:end);
12 delta_cl = max(ultimi_cl) - min(ultimi_cl);
13 media_cl = mean(ultimi_cl);
14 F_lift_mean = 0.5 * rho * U_max^2 * Aref * media_cl;
15 F_lift = 0.5 * rho * U_max^2 * Aref * cl;
16
17 figure;
18 plot(passi, F_lift, 'LineWidth', 1.5);
19 grid on;
20 xlabel('Step'); ylabel('Lift');
21
22 fprintf('Mean Cl: %.6e\n', media_cl);
23 fprintf('Mean Lift: %.6e\n', F_lift_mean);
24 fprintf('Delta Cl (Stability): %.6e\n', delta_cl);
25 if delta_cl < 1e-5

```

```

26     fprintf('STATUS: Steady (below threshold)\n');
27 else
28     fprintf('STATUS: Oscillating (above threshold - Von Karman)\n');
29 end

```

The script takes column 5 as C_L , averages the last 100 steps to get a meaningful estimate even when oscillations are present, and plots the lift force history as shown in Figure 3.4.

Modifications for Chapter 5

Since the lift force is extracted directly from OpenFOAM via the `forces` function, the post-processing script was updated accordingly. The file read is now `force.dat` instead of `coefficient.dat`, and the lift is obtained by summing the pressure and viscous contributions along the z -direction, stored in columns 7 and 10 respectively, without any further scaling:

Listing B.2: MATLAB – lift force analysis

```

1  label={lst:matlab_lift_ellipse}]
2  data = readmatrix('force.dat', 'FileType', 'text', 'CommentStyle',
    '#');
3  data = data(~any(isnan(data), 2), :);
4  F_lift = data(:, 7) + data(:, 10); % Fpz + Fvz
5  passi = 1:length(F_lift);
6  U_max = 0.0024;
7
8  % Analysis of the last 100 steps
9  ultimi_lift = F_lift(end-100:end);
10 delta_cl = max(ultimi_lift) - min(ultimi_lift);
11 media_cl = mean(ultimi_lift);
12 F_lift_mean = media_cl;
13
14 figure;
15 plot(passi, F_lift, 'LineWidth', 1.5);
16 grid on;
17 xlabel('Step'); ylabel('Lift');
18 fprintf('Mean Lift: %.6e\n', F_lift_mean);
19 exportgraphics(gcf, 'Lift.png', 'Resolution', 300)

```

Pressure analysis

Together with the lift analysis, a second script was used to study the pressure signal at the probe location defined in `controlDict`, as described in the previous chapter.

Listing B.3: MATLAB – pressure probe analysis

```

1  clear; clc;
2
3  filename = 'p';
4  D = 0.4;
5
6  raw_data = readmatrix(filename, 'FileType', 'text', ...
7      'CommentStyle', '#');
8
9  tempo = raw_data(:,1);
10 pressione = raw_data(:,2);
11
12 p_media = mean(pressione(end-100:end));
13 residuo_fisico = abs(pressione - p_media);
14
15 figure('Color', 'w');
16
17 subplot(2,1,1);
18 plot(tempo, pressione, 'b', 'LineWidth', 1.5);
19 grid on;
20 title(['Pressure (', num2str(D), ', 0, ', num2str(D/2), ')']);
21
22 subplot(2,1,2);
23 semilogy(tempo, residuo_fisico + 1e-12, 'r', 'LineWidth', 1.5);
24 grid on;
25 title('Residual');
26
27 std_p = std(pressione(end-100:end));
28 fprintf('Mean pressure: %.2e\n', p_media);
29 fprintf('Standard deviation: %.2e\n', std_p);
30 if std_p < 1e-5
31     fprintf('Status: STEADY\n');
32 else
33     fprintf('Status: OSCILLATING\n');
34 end

```

The script plots the pressure signal and the residual over time, giving a clear picture of whether the flow has reached a steady state or not. The results are shown in Figure 3.5.

B.2. λ_c vs p

For the study in Chapter 3, a simple script was used to plot the critical threshold λ_c as a function of p :

Listing B.4: MATLAB – critical threshold vs inlet exponent

```

1  P_vals = [1, 2, 4, 8, 16, 30, 50];
2  lambda_vals = [0.0040, 0.0085, 0.0092, 0.0096, 0.0097, 0.0097,
    0.0097];
3
4  figure('Color', 'w');
5  hold on;
6  plot(P_vals, lambda_vals, '-o', 'LineWidth', 0.5, ...
7       'MarkerSize', 4, 'MarkerFaceColor', 'b');
8  grid on;
9  xlabel('$p$', 'Interpreter', 'latex', 'FontSize', 14);
10 ylabel('$\lambda_c$', 'Interpreter', 'latex', 'FontSize', 14);
11 xlim([0 55]);
12 hold off;

```

The resulting plot is shown in Figure 3.6.

B.3. Inlet Profile Comparison

This script was used to visualize the inlet velocity profiles for different values of p , in order to compare them as shown in Figure 3.7:

Listing B.5: MATLAB – inlet profile visualization

```

1  clear; clc; close all
2
3  p = 50;
4  N = 81;
5
6  y = linspace(-1, 1, N);
7  z = linspace(-1, 1, N);
8  [Y, Z] = meshgrid(y, z);
9
10 U = (1 - abs(Y).^p) .* (1 - abs(Z).^p);
11 U(U < 0) = 0;
12
13 figure('Color','w')
14 surf(Y, Z, U, 'EdgeColor','none')
15 shading interp
16 colormap parula
17
18 xlabel('$y$', 'Interpreter','latex')
19 ylabel('$z$', 'Interpreter','latex')
20 zlabel('$u$', 'Interpreter','latex')

```

```

21 title(sprintf('$p = %g$', p), 'Interpreter','latex')
22
23 set(gca, 'FontName','Times', 'FontSize', 11)
24 view(-37.5, 30)
25 axis tight
26 grid on

```

B.4. Equilibrium Position Analysis

In Chapter 4, after computing the lift force at each cylinder position with the first script, the following script was used to plot the lift as a function of the cylinder center z -coordinate:

Listing B.6: MATLAB – lift vs cylinder position

```

1 clear; close all; clc;
2
3 rho = 1.225;
4 U = 0.0048;
5 A = 0.8;
6
7 list = dir('cilindro_*');
8
9 z = [];
10 L = [];
11
12 for i = 1:length(list)
13     nome = list(i).name;
14     tok = regexp(nome, 'cilindro_(-?\d+\.? \d*)', 'tokens');
15     if isempty(tok)
16         continue
17     end
18     zi = str2double(tok{1}{1});
19
20     file = fullfile(nome, 'coefficient.dat');
21     if ~exist(file,'file')
22         fprintf('missing coefficient.dat in --> %s\n', nome);
23         continue
24     end
25
26     dati = importdata(file);
27     Cl = dati.data(:,5);
28
29     n = 100;

```

```

30     Cl_avg = mean(Cl(end-n:end));
31     Lift = 0.5 * rho * U^2 * A * Cl_avg;
32
33     z(end+1) = zi;
34     L(end+1) = Lift;
35 end
36
37 [z, idx] = sort(z);
38 L = L(idx);
39
40 figure
41 plot(z, L, '-o', 'LineWidth', 0.5, 'MarkerSize', 4, ...
42      'MarkerFaceColor', 'b');
43 hold on
44 plot([min(z) max(z)], [0 0], 'k--');
45 grid on
46 xlabel('Cylinder center  $z$ ', 'Interpreter', 'latex')
47 ylabel('Lift', 'Interpreter', 'latex')

```

The zero line is added as a reference to identify the positions where the lift vanishes, which correspond to the equilibrium positions. The result is shown in Figure 4.3 and 4.4.

B.5. λ_c vs ratio

To plot λ_c as a function of the ratio a/b for ellipses with constant perimeter and constant area, the following MATLAB script was employed:

Listing B.7: MATLAB – λ_c vs ratio

```

1  Ratio_vals = [0.24, 0.62, 1.00, 1.46, 2.00, 2.25, 2.50, 3.17];
2  lambda_vals = [0.0024, 0.0035, 0.0048, 0.0072, 0.0114, 0.0145, 0.0180,
3      0.0300].*16/9;
4  figure('Color', 'w');
5  hold on;
6  plot(Ratio_vals, lambda_vals, '-o', 'LineWidth', 0.5, 'MarkerSize', 4,
7      'MarkerFaceColor', 'b');
8  grid on;
9  xlabel('Ratio', 'Interpreter', 'latex', 'FontSize', 14);
10 ylabel('Critical Threshold', 'Interpreter', 'latex', 'FontSize', 14);
11 xlim([0 3.5]);
12 ylim([0 max(lambda_vals) * 1.1]);
13 hold off;

```

The same script was used to produce both Fig. 5.2 and Fig. 5.3, with the appropriate λ_c

values substituted in each case. To compare the two cases, the script used to generate Fig. 5.4 is reported below:

Listing B.8: MATLAB – λ_c vs ratio comparison

```

1 Ratio_vals = [0.24, 0.62, 1.00, 1.46, 2.00, 2.25, 2.50, 3.17];
2 lambda_vals_perimetro = [0.0027, 0.0035, 0.0048, 0.0073, 0.0132,
   0.0168, 0.0220, 0.0590].*16/9;
3 lambda_vals_area = [0.0024, 0.0035, 0.0048, 0.0072, 0.0114, 0.0145,
   0.0180, 0.0300].*16/9;
4 figure('Color', 'w');
5 hold on;
6 plot(Ratio_vals, lambda_vals_perimetro, '-o', 'LineWidth', 0.5, '
   MarkerSize', 4, 'MarkerFaceColor', 'b');
7 hold on
8 plot(Ratio_vals, lambda_vals_area, '-o', 'LineWidth', 0.5, 'MarkerSize
   ', 4, 'MarkerFaceColor', 'r');
9 grid on;
10 xlabel('Ratio', 'Interpreter', 'latex', 'FontSize', 14);
11 ylabel('Critical Threshold', 'Interpreter', 'latex', 'FontSize', 14);
12 xlim([0 3.5]);
13 ylim([0 max(lambda_vals_perimetro) * 1.1]);
14 legend('constant perimeter', 'constant area', 'Interpreter', 'latex',
   'FontSize', 14);
15 hold off;

```


List of Figures

2.1	Rigid computational domain geometry.	5
2.2	Comparison of wake oscillations.	6
3.1	Poiseuille inlet profile	9
3.2	View of the computational domain.	10
3.3	Detail of the mesh refinement around the cylinder.	11
3.4	Comparison of lift force	12
3.5	Comparison of Pressure	13
3.6	Critical threshold λ_c as a function of the inlet profile exponent p	14
3.7	Inlet velocity profiles for different values of the exponent p	15
4.1	Cylinder at 0.2.	18
4.2	U profile around cylinder at 0.2.	19
4.3	Lift force L as a function of h	20
4.4	Lift force $L(h)$ as a function of the vertical displacement h for each value of λ	21
4.5	The Leonardo da Vinci ferry on the river Adda.	24
4.6	The Leonardo da Vinci ferry mechanism.	24
5.1	Comparison of three elliptical cross-sections with the same perimeter at aspect ratios $k = 0.24$, $k = 1.46$, and $k = 3.17$	27
5.2	Critical threshold λ_c as a function of the aspect ratio $k = a/b$	28
5.3	Critical threshold λ_c as a function of the aspect ratio $k = a/b$	29
5.4	Comparison of λ_c vs ratio for constant perimeter and constant area ellipses.	30

List of Tables

3.1	Critical values of the bifurcation parameter λ_c for different inlet profile exponents p in (3.3).	13
4.1	Displacement h and corresponding Lift	19
4.2	Equilibrium positions and number of equilibria for each value of λ	20
5.1	Ellipses at constant perimeter.	26
5.2	Ellipses at constant area.	26
5.3	Critical values of λ_c	28
5.4	Critical values of λ_c	29

