

Politecnico di Torino

Politecnico di Torino

L.M. in Ingegneria Energetica e Nucleare

A.a. 2025/2026

Sessione di Laurea Luglio 2026

Numerical Investigation of Flow Regime Transitions in Lidinoid TPMS Geometries

Relatori:

Laura Savoldi

Luca Marocco

Candidato:

Pablo Cejas

Abstract

Triply Periodic Minimal Surfaces (TPMS) have emerged as highly efficient geometric frameworks for advanced engineering applications, ranging from heat exchangers in fusion reactors to bioengineering implants. Their high surface-to-volume ratio and tunable porosity make them ideal for enhancing thermal-hydraulic performance. This thesis focuses on the Lidinoid TPMS. The primary objective of this work is to identify the onset of the non-laminar flow regime within a Lidinoid structure at a fixed porosity of $\phi = 50\%$. This was performed analyzing the variation of average Turbulence Intensity (TI) through increasing Reynolds numbers (Re). Given the inherent unpredictability of transitory flows, pinpointing the Reynolds number at which the flow shifts from steady laminar to unsteady and eventually turbulent behavior is critical for reliable industrial design. Numerical simulations were conducted implementing the open-source OpenFOAM software, by using the simpleFoam and pimpleFoam solvers for steady-state and transient flows respectively. The geometries were generated via implicit modeling in nTopology and discretized using cfMesh. The study employs key hydrodynamic parameters, including tortuosity (τ), turbulent intensity (TI), and the skin-friction coefficient (C_f), to characterize flow physics. A Grid Convergence Index (GCI) study was performed to ensure the independence of the results from the mesh density. The results provide a comparative analysis between steady and unsteady laminar, as well as transitory, behaviors, indicating Re limits for laminar simulations in Lidinoid TPMS geometries. Ultimately, the transition from unsteady to turbulent flow was analyzed.

Table of Contents

List of Tables III

List of Figures IV

1 Introduction 1
- 1.0.1 Additive Manifacturing . . . 2
- 1.0.2 Nuclear Applications . . . 3

1.1 Triply Periodic Minimal Surfaces . . . 5
- 1.1.1 Lidinoid . . . 7

1.2 Objective . . . 8

2 Theoretical Basis 9

2.1 Governing equations in computational fluid dynamics . . . 9
- 2.1.1 TPMS parameters . . . 9

2.2 Turbulent flows . . . 11

2.3 Laminar flows . . . 12

2.4 Darcy and Forchheimer flows . . . 12
- 2.4.1 The Darcy Regime . . . 12
- 2.4.2 The Forchheimer Regime . . . 13
- 2.4.3 The Unsteady and Turbulent Porous Regimes . . . 13

3 Tools and Softwares 14

3.1 OpenFOAM . . . 14
- 3.1.1 Solvers . . . 18

3.2 cfMesh . . . 20

3.3 nTopology . . . 21

4 Simulation set-up 23

4.1 Laminar model . . . 23

4.2 Time model . . . 24

4.3 Geometry . . . 24

4.4 Boundary conditions 25
4.5 Mesh generation 26
4.6 Convergence study 29
4.7 Methodology 37
4.8 Data acquisition 38

5 Results 41
5.1 Variables dependence on Reynolds number 54
5.2 Flow Transition 57
5.2.1 Steady to Unsteady Flow Regime Transition 57
5.2.2 Unsteady to Turbulent Flow Transition 58

6 Conclusions 60

Bibliography 62

List of Tables

4.1 Characteristics of the meshes 30
4.2 Grid refinement values 32
4.3 Grid Convergence Index results for Δp 34
4.4 Grid Convergence Index results for τ 35
4.5 Grid Convergence Index results for C_f 36

List of Figures

1.1 Overview of TPMS possible applications[5] . . . 1
1.2 Additive manufacturing processes[22] . . . 2
1.3 In the red circle indicates the divertor . . . 3
1.4 Model studied in [19] . . . 4
1.5 Fuel configuration for primitive TPMS . . . 4
1.6 Example of Solid-TPMS . . . 6
1.7 Example of Sheet-TPMS [21] . . . 7

3.1 Step-by-step SIMPLE algorithm iteration[2] . . . 18
3.2 nTopology workflow . . . 22

4.1 Lidinoid of $\phi = 50\%$ . . . 25
4.2 Fluid volume of the geometry . . . 27
4.3 Off-plane view of section captured at z = 0.0025 m . . . 28
4.4 Close-up of the chokepoint in the geometry . . . 28
4.5 Isometric view . . . 29
4.6 Main caption for the triangular layout. . . . 30
4.7 Δp convergence . . . 34
4.8 τ convergence . . . 35
4.9 Grid Convergence Index results for C_f . . . 36
4.10 TI average and maximum values for the studied configurations . . . 37
4.11 Probe locations . . . 39

5.1 Skin-friction coefficient at different Reynolds numbers . . . 42
5.2 Pressure drop at different Reynolds numbers . . . 43
5.3 Residuals at different Reynolds numbers . . . 44
5.4 C_f convergence . . . 46
5.5 ΔP evolution . . . 46
5.6 τ evolution . . . 47
5.7 Residuals evolution . . . 47
5.8 C_f convergence . . . 48

5.9 ΔP evolution 49
5.10 τ evolution 49
5.11 Residuals evolution 50
5.12 C_f convergence 51
5.13 ΔP evolution 51
5.14 τ evolution 52
5.15 Residuals evolution 52
5.16 Friction factor evolution at increasing Re 54
5.17 C_f evolution at increasing Re 55
5.18 τ evolution at increasing Re 56
5.19 TI evolution with respect to the Reynolds number 57
5.20 Intersection of stable and increasing trends 58
5.21 Average TI for higher Re values 59

Chapter 1

Introduction

This thesis is the result of an internship at Politecnico di Milano in 2025 and the subsequent work completed under the supervision of Professor Luca Marocco as company tutor from Politecnico di Milano. It involved the setting up and performing of simulations of a TPMS Lidinoid cell at low-Reynolds flows using the open-source OpenFOAM tool. TPMS geometric structures have played an increasingly important role in the development of new technologies in recent years. The relevance and convenience of these structures have grown in the last decade thanks to the increased use of additive manufacturing, which has made the production of these geometries possible and relatively cheap.

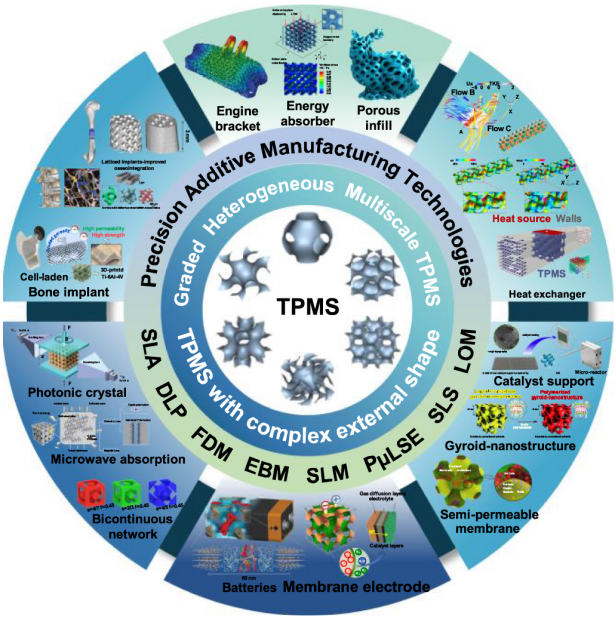


Figure 1.1: Overview of TPMS possible applications[5]

Therefore, these porous structures can be easily produced and calibrated to the specific needs of the application. They have been studied for various uses that space from bioengineering to fluid dynamics, from adsorption surfaces to structural materials, as membranes, implants, and many others as can be seen in Fig. 1.1. TPMS vary in geometry, and different structures are used in different contexts. Schwarz-P are often considered structural elements, while gyroids are among the best performing TPMS in the context of fluid heat exchange. Particularly interesting are applications with high heat load, such as in fusion reactors, where they can reach the order of MW/m^2. Additive manufacturing, smooth surfaces, and tunability make these geometries preferred to traditional porous systems, such as pebble beds, foams, or Raschig rings.

1.0.1 Additive Manifacturing

Additive manufacturing in the last two decades has seen ever increasing applications in the industrial field. It is also the technology that makes TPMS production affordable and practical. Additive manufacturing processes usually take the information from a computer-aided design (CAD) file that is later converted to a STL file. In this process, the drawing made in the CAD software is approximated by triangles and sliced containing the information of each layer that is going to be printed. However, it is also possible to directly generate STL files, when using specific software, such as nTopology. 3D printing, for example, is a name for additive manufacturing.

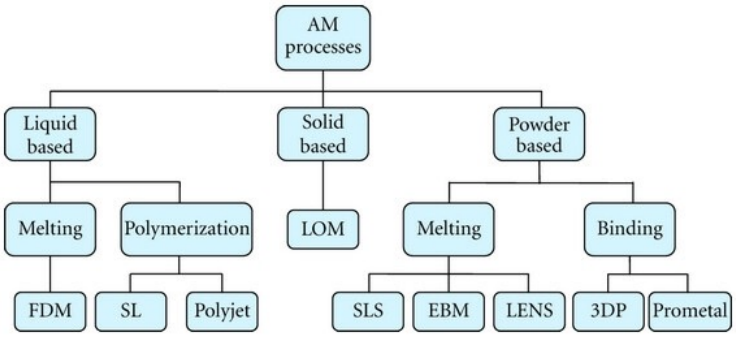


Figure 1.2: Additive manufacturing processes[22]

In Fig.1.2, the various processes possible for additive manufacturing are shown. Of these, Fused Deposition Modeling (FDM) consists in a plastic filament, fused and deposited so to compose a 3D geometry, and it is one of the most used technologies of additive manufacturing by hobbyists. Instead, Prometal is used instead to

cheaply build metallic structures, such as engines, and not only. Any of these could be used to produce TPMS geometries, depending on the specific applications.

1.0.2 Nuclear Applications

In energy-related applications, TPMS geometries have been applied mostly to thermal-exchange and cooling scenarios. Most simply, as heat exchangers, either solid-to-fluid or fluid-to-fluid, depending on the configuration, as will be mentioned later. These geometries offer large surfaces for small volumes, and their tunability makes them compatible to a wide range of uses and applications.

In particular, TPMS geometries have been studied in the past decade for possible applications as divertor tile cooling systems. Divertors are used in fusion reactors to extract ash and heat from the reactor. These components are subjected to high heat loads, from 10 to $20 MW/m^2$ and therefore a lot of research is being done on how to better extract these massive heat loads. In particular, for the W7-X stellarator in Germany.

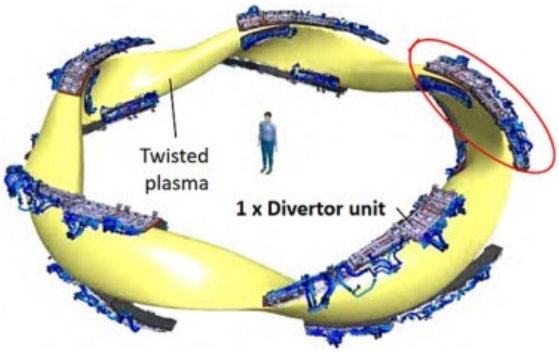


Figure 1.3: In the red circle indicates the divertor

Research is being conducted on the effectiveness of TPMS for this application(Fig.**??**) , where to ensure chemical compatibility with tritium, the divertor targets in W7-X typically utilize a tungsten armor layer due to its exceptionally high melting point, which is limited to a 1 mm thickness in order to minimize the weight. It is joined to the underlying cooling structure via a copper inter-layer. In the case of W7-X, the cooling framework itself is embedded within a structural block of a CuCrZr alloy. Under this external structure the TPMS structure is used[19].

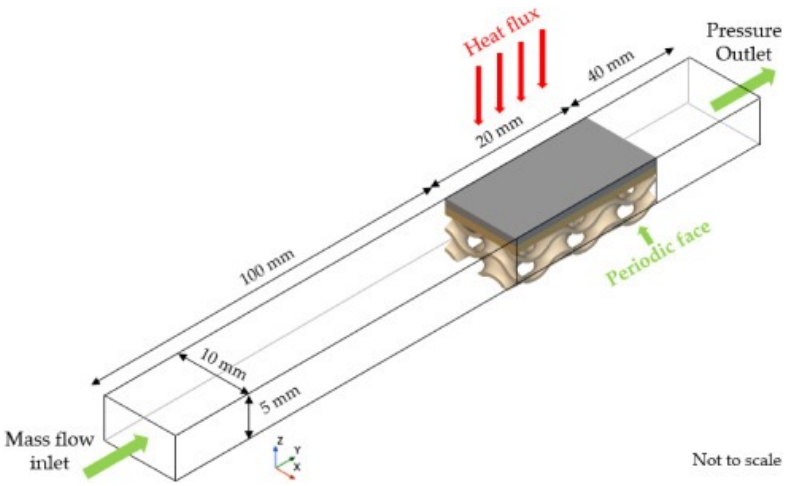
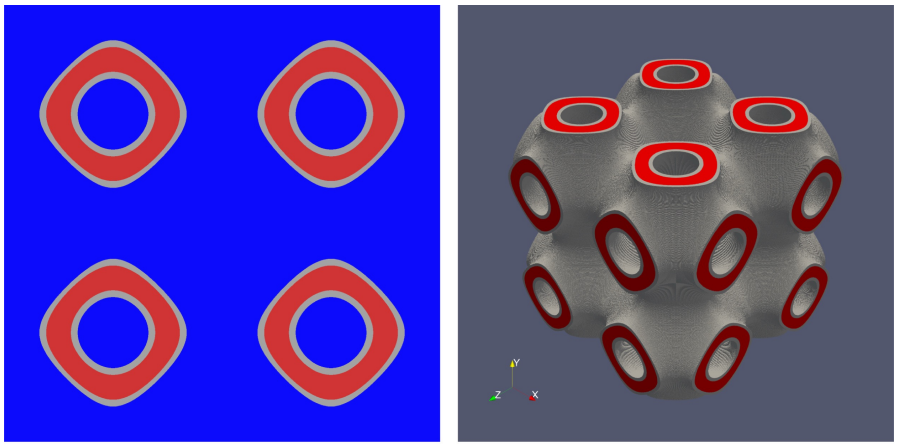


Figure 1.4: Model studied in [19]

In recent years, further applications have been studied. Such as for fuel rod geometries for nuclear fission reactors[7] as seen in Fig.1.5.



(a) OXY-plane cut view - coolant is blue, fuel is red, cladding is grey

(b) 3D view

Figure 1.5: Fuel configuration for primitive TPMS

1.1 Triply Periodic Minimal Surfaces

Minimal surfaces are defined as structures with zero mean curvature. A unique subset of these are Triply Periodic Minimal Surfaces (TPMS), which exhibit periodicity in three independent spatial directions. Found naturally in crystals and biological membranes, they were first described mathematically by Schwarz in the 1800s. His foundational definition of the Primitive (P) surface is expressed in (1.1).

$$cos(x) + cos(y) + cos(z) = 0 \tag{1.1}$$

As non-intersecting structures generated by combinations of trigonometric functions in 3D space, TPMS are implicitly defined and highly tunable. They offer exceptionally smooth surface profiles while maximizing the surface-to-volume ratio—a characteristic highly beneficial for enhancing heat transfer without causing a prohibitive increase in pressure drop.

Being implicit surfaces, TPMS are governed by characteristic trigonometric equations, they can be easily parameterized to meet specific thermal or hydraulic requirements. The generic governing equation is simplified as

$$f(x, y, z) = C \tag{1.2}$$

It can thus be defined as iso-surface. By setting the characteristic function equal to zero or to a specific constant value, the iso-surface can be offset to precisely control and alter the structure's porosity.

They can be calculated according to the Enneper-Weierstrass equation as

$$\begin{cases} x = \mathrm{Re}\left(\mathrm{e}^{\mathrm{i}\theta} \int_{\omega_0}^{\omega} (1-\tau^2) R(\tau)\, \mathrm{d}\tau\right) \\ y = \mathrm{Re}\left(\mathrm{e}^{\mathrm{i}\theta} \int_{\omega_0}^{\omega} \mathrm{i}(1+\tau^2) R(\tau)\, \mathrm{d}\tau\right) \\ z = \mathrm{Re}\left(\mathrm{e}^{\mathrm{i}\theta} \int_{\omega_0}^{\omega} 2\tau R(\tau)\, \mathrm{d}\tau\right) \end{cases} \tag{1.3}$$

where $i^2 = -1$, τ is a complex variable, θ is defined as the Bonnet angle, and Re is the real part of a complex variable. $R(\tau)$ is the Weierstrass function, which can be expressed as

$$R(\tau) = \frac{1}{\sqrt{\tau^8 - 14\tau^4 + 1}}. \tag{1.4}$$

Another method involves the generation of the function with

$$\phi(\mathbf{r}) = \sum_{k=1}^{K} A_k \cos\left[\frac{2\pi(\mathbf{h}_k \cdot \mathbf{r})}{\lambda_k} + P_k\right] = C \tag{1.5}$$

where A_k is the amplitude, λ_k is the period factor, and P_k is the function phase. Through the functions' parameters, not only the porosity can be easily changed and tuned, but they can also be stretched, or even the TPMS structure along a single geometry. Two main different configurations are possible, which can have different uses in thermal applications:

- Solid-TPMS: One subdomain is defined as a solid matrix and the other as a fluid domain. This creates a single-channel region ideal for heat sinks.

- Sheet-TPMS: The solid boundary itself is hollowed out, partitioning the space into two separate, continuous fluid channels. This matrix configuration is highly effective for fluid-to-fluid heat exchangers [6, 23].

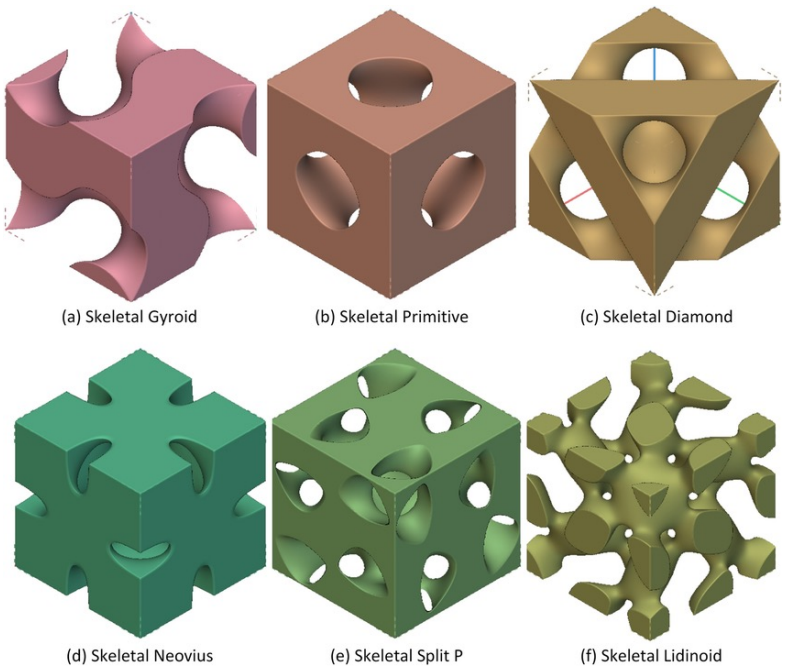


Figure 1.6: Example of Solid-TPMS

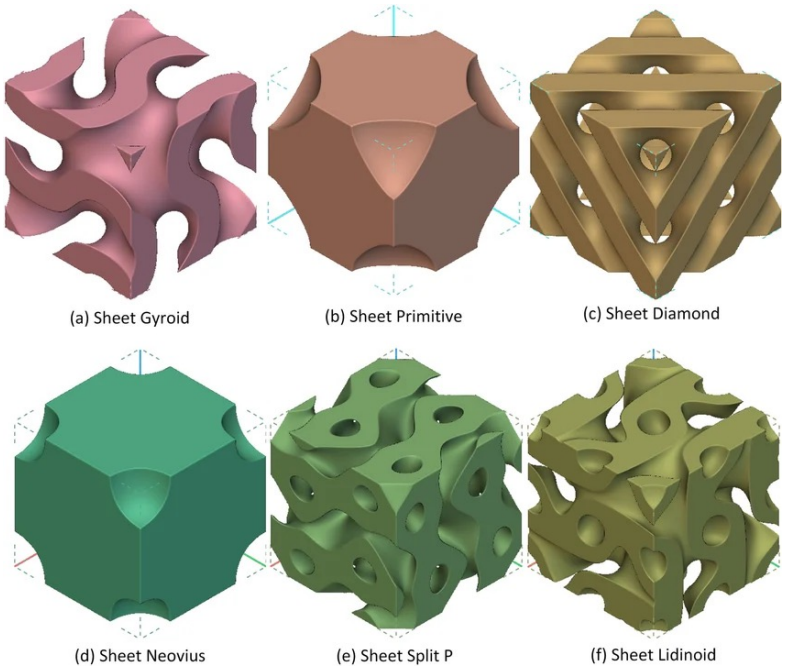


Figure 1.7: Example of Sheet-TPMS [21]

1.1.1 Lidinoid

During this thesis, only a solid Lidinoid cell will be considered. The Lidinoid TPMS, differing from the primitive Schwarz surface, was introduced in 1990 by Sven Lidin [10] and the defining iso-surface function can be expressed as:

$$\begin{aligned} f(x,y,z) = \; & \sin(2x)\cos(y)\sin(z) + \sin(2y)\cos(z)\sin(x) + \sin(2z)\cos(x)\sin(y) \\ & - \cos(2x)\cos(2y) - \cos(2y)\cos(2z) - \cos(2z)\cos(2x) + 0.3 \end{aligned} \tag{1.6}$$

From a topological perspective, the Lidinoid belongs to the associate family of the Schwarz Hexagonal (H) surface and can be generated at a specific Bonnet transformation angle ($\theta \approx 64.25°$). Unlike the highly symmetric orthogonal channels found in the Schwarz Primitive (P) or Diamond (D) matrices, the Lidinoid features a network of screw-like helical channels. This absence of reflectional symmetry and flat mirror planes creates a continuously smooth, winding pathway for passing fluids. Although the Lidinoid does not possess as good thermal capabilities or heat-transfer coefficients typically as observed in Gyroid or Diamond geometries [14, 3], its particular properties make it available for other purposes, in particular that require low pressure drops. Studies have shown that its continuous, sweeping channel curvature promotes a low-tortuosity fluid path along the core of the

channels [9]. Consequently, fluid elements tend to glide through the center of the passages with minimal frontal obstruction. This fluid-wall interaction profile yields a remarkably low pressure drop across the lattice and a lower Reynolds number per unit of mass flow rate compared to standard TPMS geometries [9]. By drastically mitigating localized stagnation zones, sharp directional shifts, and intense shear stresses at the solid boundaries, the Lidinoid reduces hydraulic resistance. Therefore, while it may underperform in applications demanding severe thermal dissipation via turbulent mixing, it serves as an excellent candidate for energy-efficient fluid transport

1.2 Objective

The primary objective of this work is to study the regime changes in a Lidinoid TPMS through the use of a number of numerical simulations. In particular, the detection of the onset of unsteady laminar flow, characteristic of porous flows at non-Darcy regimes. A secondary objective, is to identify the transition from unsteady to turbulent flow.

Chapter 2

Theoretical Basis

In the following chapter a quick theoretical basis of the simulations will be presented.

2.1 Governing equations in computational fluid dynamics

Depending on the physical conditions that are taken into account for the fluid, such as compressibility, the governing equations will be defined differently. During this thesis, only constant density, Newtonian, and isothermal fluids will be considered and therefore it is possible to define:

$$\nabla \cdot \mathbf{u} = 0. \tag{2.1}$$

The governing momentum equation, which under these conditions is the simplified Navier-Stokes equation for momentum can be expressed as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u} \otimes \mathbf{u}) - \nu \nabla^2 \mathbf{u} = -\nabla p \tag{2.2}$$

where U is the velocity, p the pressure, and ν the kinematic viscosity. This equation relates the acceleration of the fluid particles to body forces and surface forces experienced by the liquid. It also must be mentioned that given the small size of the cell considered in this thesis the gravity forces have been neglected. These equations will be applied by the OpenFOAM solvers for 3D fluids.

2.1.1 TPMS parameters

For porous structures, a number of parameters need to be introduced, which are either not used or described differently in studies concerning conventional

geometries, such as pipes. These are custom definitions that describe the flow and are strictly dependent on the geometry at hand. Porosity is the key parameter in the study of these geometries and is defined as the ratio between the void volume (V_f), and the total cell volume (V_{tot}):

$$\phi = \frac{V_f}{V_{tot}}. \tag{2.3}$$

Another parameter is the Reynolds number, which in porous structures is commonly expressed as

$$Re = \frac{D_h U_p}{\nu} \tag{2.4}$$

where D_h is the hydraulic diameter for porous geometries generally identified in this case as

$$D_h = 4\phi \frac{V_{tot}}{A_{wet}} \tag{2.5}$$

in which A_{wet} is the wetted surface of the geometry and V_{tot} is the total volume of the cell. U_p is called the pore velocity

$$U_p = \frac{U_s}{\phi} \tag{2.6}$$

where

$$U_s = \frac{\dot{m}}{\rho L_c^2}. \tag{2.7}$$

In Eq. (2.7), with L_c the characteristic length of the geometry is set, which in this case would be the streamwise length of the geometry.

A further parameter that is crucial in describing the fluid dynamics of porous structures is tortuosity (τ), expressed as the ratio between the volume averaged velocity magnitude and the volume averaged velocity in the main flow direction, which in this work would be x.

$$\tau = \frac{\langle U \rangle}{\langle u_x \rangle} \tag{2.8}$$

The tortuosity represents the ratio between the path length taken by the fluid particle and the straight-line path. Therefore, it gives an idea of the complexity of the flow inside the geometry, and is directly tied to the porosity, though not necessarily linearly[8]. This value is always > 1.

To detect turbulence and relate the Re number and flow regime in the Lidinoid cell, the turbulent intensity (TI) and the skin-friction coefficient (C_f) were also

observed (Eqs.(2.9)(2.10)). Turbulent intensity is a percentage value that relates fluctuations in velocity with the mean flow calculated in time. It gives an idea of the level of unsteadiness of a flow, as opposed to steady flows where this value is or close to zero. The skin-friction coefficient instead is an often-used variable to detect the transition from laminar to transitory behavior, as it goes down as the flow becomes cahotic. This is due to larger viscous forces dominance in lower Reynolds flows. It must be considered that in this work, as will be explained later, the bulk velocity was not changed, so the variation of C_f will be caused by the explained phenomenon.

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho U_\infty^2} \tag{2.9}$$

$$TI = \frac{u'_{rms}}{\overline{u}}. \tag{2.10}$$

In Eq.(2.9), the skin-friction coefficient is defined as the ratio between the average shear stress τ_w, and the density ρ and U_∞ squared. In addition, for the study of convergence, the friction factor was utilised. For porous structures it is generally defined as

$$f = \frac{2D_h \Delta P}{\rho L_c U_p^2} \tag{2.11}$$

In Eq.(2.10), $\overline{u}$ is the mean of the main velocity at a point[**?**], and u'_{rms} is defined as the root mean square of the local fluctuating components of the velocity (Eq.(2.12)).

$$u'_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(u'_i\right)^2} \tag{2.12}$$

where N is the number of dimensions considered during the simulation, in this case $N = 3$, and u'_i is equal to

$$u'_i = |u_i| - \overline{u} \tag{2.13}$$

where finally, u_i represents the fluctuating velocity component, or the instantaneous local velocity component.

2.2 Turbulent flows

Turbulent regimes are defined by stochastic, chaotic motion and are characterized by the presence of eddies. Fluid particles do not move in layered streamlines, but rather the flow undergoes constant lateral mixing, from the walls into the bulk

flow. This is aided by the generation of eddies that are circular, swirling fluxes that characterize these flows. In thermal contexts, turbulence is ideal and is often preferred in engineering contexts. In po

2.3 Laminar flows

As opposed to turbulent flows, laminar flows consists of an orderly fluid movement that does not generate eddies nor chaotic behavior of any kind. Laminar flows are the simplest to solve in that the fluid particles follow well-defined streamlines along the geometry of interest, and since no eddies need to be modeled no dissipation nor turbulent energy equations are needed. Between full laminar and turbulent, lies a transitory regime. These flows are notoriously difficult to predict due to the intermittent alternation between laminar and turbulent states. This happens due to inertial forces being high enough to destabilize the flow, and opposed to the viscous forces, which tend to stabilize the flow. Before the flow enters the transitory regime, it can become unsteady. What this implicates is that the results will not be constant in time, but they will change, notwithstanding the convergence having been reached in the simulation[4].

2.4 Darcy and Forchheimer flows

Although the classic definitions of laminar, transitional, and turbulent flows describe open-channel fluids well, fluid behavior of the fluid changes significantly when passing through a porous medium, such as packed beds, metal foams, or, in this case, TPMS. In these geometries, macroscopic flow regimes are defined not just by the fluid's intrinsic properties, but by its microscopic interactions with the solid structural boundaries.

2.4.1 The Darcy Regime

At very low velocities, corresponding to a Reynolds number on the pore-scale ($Re < 10$ [15]), fluid movement is completely dominated by viscous shear forces that resist the walls of the solid matrix. Under these conditions, inertial forces are mathematically negligible. This regime is governed by what is called Darcy's Law:

$$\left|\frac{\Delta P}{L}\right| = \frac{\mu}{K} U_s \tag{2.14}$$

where $\Delta P/L$ is the pressure drop per unit length, μ is the dynamic viscosity, K is the intrinsic permeability of the medium, and U_s is the superficial velocity. In this regime, the relationship between the driving pressure gradient and the resulting

fluid velocity is perfectly linear, acting as a porous medium equivalent to steady, fully developed laminar flow. This range has not been studied during this work.

2.4.2 The Forchheimer Regime

As the superficial velocity increases (generally speaking, $Re > 10$), inertial forces begin to gradually become more relevant, although the flow remains macroscopic and structurally ordered. The fluid particles moving through the intricate channels experience local accelerations and decelerations as they squeeze through narrow pore throats and expand into larger pore cavities, and thus inertial forces become more and more prevalent as the velocity, or, more generally, the Reynolds increases.

These spatial accelerations introduce kinetic energy dissipation that is not contemplated in Darcy's Law. To account for these inertial losses, the governing equation must be expanded to the quadratic Darcy-Forchheimer equation[11]:

$$\left|\frac{\Delta P}{L}\right| = \frac{\mu}{K_1} U_s + \frac{\rho}{K_2} U_s^2 \tag{2.15}$$

where ρ is the fluid density, K_1 is the Forchheimer permeability constant, and K_2 is the inertial permeability constant. It is vital to note that while this regime is Non-Darcy (also denoted as nDarcy) and unsteady due to inertia, it is not turbulent. The flow remains laminar in nature[11].

2.4.3 The Unsteady and Turbulent Porous Regimes

If the velocity continues to scale upward, the flow eventually breaks away from steady inertial behavior. This limit varies for each porous structure, from porosity to geometry and to cell size. Much like the transition described in open-channel flows, high inertial forces overcome the stabilizing viscous forces within the pores. The fluid detaches from the solid matrix boundaries, forming localized wake zones, unsteady vortex shedding, and eventually, full stochastic turbulence. As the flow increases its unsteadiness, this can be captured analyzing the velocity oscillations of the flow through the TI.

Chapter 3

Tools and Softwares

3.1 OpenFOAM

OpenFOAM is a powerful computational simulation toolbox and is the most widely used open-source CFD software available. Currently, it is used for both research and commercial purposes. This tool is capable of simulating a large variety of physical scenarios, not only CFD, but also stress analysis and electro-magnetic phenomena. Being a toolbox, it also allows for easy access to the codes and algorithms, and thus a flexible customization of the software is possible. In this study, only the incompressible fluid CFD capabilities have been used. Its accuracy and performance has already been tested and compared to commercial softwares such as Ansys Fluent and Star-CCM+ with positive outcomes [16]. The software can be obtained in two different versions, the openfoam.com and OpenFOAM Foundation products (openfoam.org). These two versions are almost identical and vary mostly in commands' syntax. During this work OpenFOAM.com v2412 was used.

Case file structure

In order to better understand the following chapters, it is important to explain the file organization on OpenFOAM. The single case structure consists of 4 main folders: `0`, `0.orig`, `constant`, `system`. In addition, geometry files, as well as function library folders, are also placed in this main folder. The `case/0` folder contains the initial BC and field values for velocity, pressure, and other variables that may be needed depending on the specific simulation, such as epsilon, nu, nut, et cetera. `case/0.orig` contains the saved data from the initial `0`, which may get modified. The constant folder contains data that are fixed during the simulation and represents constants during the simulation, such as which turbulence solver to use, kinematic viscosity values, as well as the mesh data.

Finally, the case/system folder contains all the dictionaries, such as meshDict and createPatchDict, and the text files containing strictly simulation information, such as time schemes to be used, solvers, solver data, time discretization, post processing functions, et cetera. Particularly important to the simulations are the fvSchemes dictionary, which contains the time, divergence, gradient, and laplacian schemes, and fvSolution, which contains the information required for the solution of the various fields. These include p and U, but also k and epsilon when required. The solver information is also included in this dictionary. Following the fvSchemes and fvSolution dictionaries:

```
ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;

    div(phi,U)      bounded Gauss limitedLinear 1;

    turbulence      bounded Gauss limitedLinear 1;
    div(phi,k)      $turbulence;
    div(phi,epsilon) $turbulence;
    div(phi,nuTilda) $turbulence;
    div(phi,R)      $turbulence;
    div(R)          Gauss linear;

    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
```

```
}

snGradSchemes
{
    default          corrected;
}
```

```
    solvers
{
    p
    {
  solver           GAMG;
        tolerance        1e-06;
        relTol           0.01;
  smoother  GaussSeidel;

  minIter   1;
    }


    U
    {
        solver           smoothSolver;
        smoother         symGaussSeidel;
        tolerance        1e-04;
        relTol           0.1;
  minIter   2;
    }

    pFinal
    {
        $p;
        tolerance        1e-06;
        relTol           0;
    }

    UFinal
    {
        $U;
        tolerance        1e-05;
        relTol           0;
    }
}
```

```
37
38 PIMPLE
39 {
40   nOuterCorrectors  1;
41   nCorrectors 3;
42   nNonOrthogonalCorrectors  1;
43   pRefCell  0;
44   pRefValue 0;
45   maxCo   1;
46 }
47 relaxationFactors
48 {
49     fields
50     {
51         p               0.5;
52     }
53     equations
54     {
55         U               0.5;
56
57     }
58 }
59
60 cache
61 {
62     grad(U);
63 }
```

3.1.1 Solvers

The CFD capabilities uses various solvers and algorithms depending on the nature of the phenomenon to be studied. Some of the algorithms are icoFoam for transient, laminar, and incompressible flows, pimpleFoam used for transient flows, and simpleFoam that can solve turbulent steady-state flows. Most of the solvers are defined by their own governing algorithms, e.g. PIMPLE, PISO, SIMPLE. Although this case study encompasses mainly laminar simulations, icoFoam and pimpleFoam have also been used to deal with unsteadiness of a purely steady state solver such as simpleFoam, as will be explained later. Following, a general description of the algorithms used will be made.

simpleFoam

SimpleFoam is a steady state solver for incompressible, steady state, laminar and turbulent flows. It is based on the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.

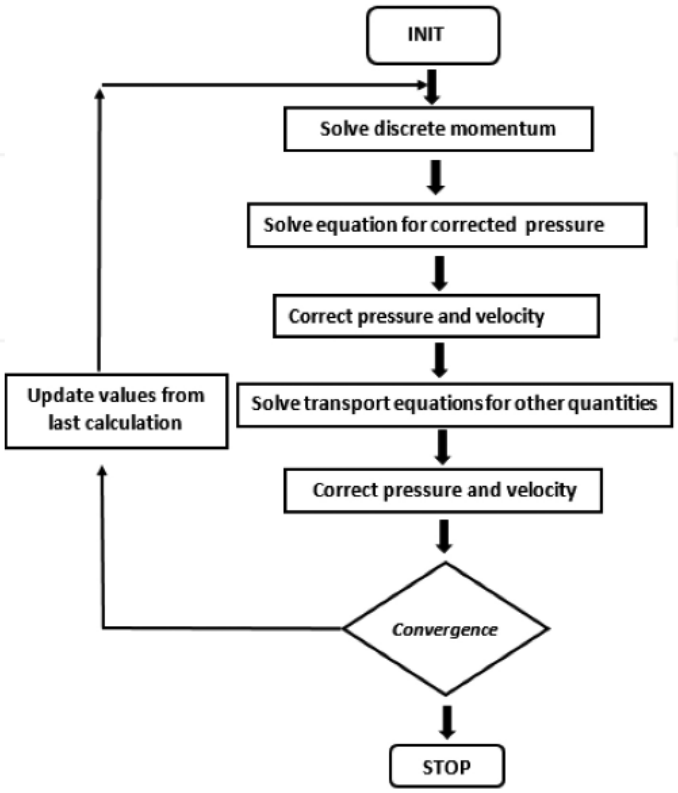


Figure 3.1: Step-by-step SIMPLE algorithm iteration[2]

SIMPLE (Semi-Implicit Method for Pressure Linked Equations) is part of the group of the pressure-velocity algorithms which also includes the PISO and PIMPLE algorithms. These algorithms solve the Navier-Stokes equations through an iterative process that follows the sequence shown in 3.1 in the case of SIMPLE. The name of the algorithm explains itself, as it iteratively goes through the pressure-velocity equations, solving implicitly the pressure linked equations and explicitly the velocity linked equations[1] . The solver employs this algorithm to solve the continuity and then the momentum equations. The input requirements are the p and U files, which contain the necessary data for the definition of the initial pressure and velocity fields. They include boundary conditions and field values. After these initializing values, given certain BC the algorithm will re-iterate the calculation of pressure and velocity, until convergence is reached. That is, the difference between the new and older values is smaller than a pre-defined value, usually defined as ϵ. However, the key difference with the following algorithm that has been used in this work is the number of correctors and the absence of under-relaxation. The corrector is used to correct the values for pressure and velocity initially calculated that were previously obtained in the iterative process. In the SIMPLE algorithm, p and U are corrected once.
This and PISO are also used by other softwares such as Ansys Fluent and STAR-CCM+.

icoFoam

IcoFoam is an incompressible, laminar, transient solver. It is based on the PISO (Pressure-Implicit with Splitting of Operators) algorithm, which does not include under relaxation of the pressure and performs multiple corrector steps, as mentioned earlier.

The algorithm solves the momentum equations to calculate an intermediate value of the velocity by using the initial pressure values. The calculated intermediate velocity field will not satisfy the continuity equation unless the pressure prediction is correct. Therefore, mass flux at cell faces is calculated and then the pressure is under-relaxed and corrected. Thus, a new velocity is calculated that satisfies continuity, but not the pressure. This is then calculated again using the new velocity field in order to obtain a new pressure value that finally satisfies continuity. The corrections can be more than 2, if so imposed. The first step, called predictor step, and the corrections are repeated at each time step when using icoFoam.

pimpleFoam

However, during this thesis the PIMPLE algorithm has been used. This has the capability of using both pimpleFoam and icoFoam solvers. In fact, the number of outer correctors and correctors can be set so that through it, an icoFoam simulation

can be produced. Outer correctors are iterations in the same time step of the PISO algorithm explained earlier. This number can be set to one, which results in a classic PISO algorithm, or to larger numbers as shown in Fig. **??**. This is generally useful for very unstable flows, such as transition regime flows. During this work, this solver has not been used with outer correctors for definitive analysis, given the generally laminar flow that has been studied.

Using a PIMPLE set as PISO has the advantage of utilizing the built-in equations of pimple, which is capable of dealing with both laminar and turbulent flows. In this way, variables like wallShearStress and `adjustTimeStep` can be used. Indeed, the former cannot be calculated using icoFoam, which does not have turbulent kinematic viscosity (ν_t) contemplated in its equations. This results in an error, and the variable cannot be saved. PIMPLE instead will register the absence of the turbulent viscosity due to the laminar turbulent solver and thus use directly only the

3.2 cfMesh

CfMesh is an open-source automatic mesh generation library that was built on top of OpenFOAM up to version v2306. In this case, v2212 was used for this reason, while for the simulations version v2412 was employed.

The library supports both 2D and 3D meshing. The workflow begins with the import of a geometry file, which is then processed according to the parameters specified in the `meshDict` dictionary located within the case's system directory. Within this dictionary, the global maximum and minimum cell sizes are established to bound the mesh resolution. After having defined the edges using the command `surfaceEdgeSurfaces` a .stl file is produced, which divides it into separate surfaces. The discriminating factor is the angle between a surface of a cell of the .stl file and another. With this file, it is now possible to refine the mesh at certain boundaries, such as walls, and add boundary layers. The `localRefinement` can be specified in two ways:

- `additionalRefinementLevels`, which calculates the local resolution relative to the boundary cell size using the formula:

$$\text{cellSize} = \frac{\text{boundaryCellSize}}{2^{\text{additionalRefinementLevels}}}$$

- `cellSize` can be explicitly defined for a chosen patch.

It is important to note that the `additionalRefinementLevels` is based on base cell size, which is extremely useful when the convergence study is performed,

as it changes accordingly when base size is reduced. After generating the mesh, OpenFOAM fills the `polyMesh` folder with the relevant cell and point data. To organize the mesh for simulation, boundaries must be precisely named using the `createPatch` command. By configuring textttsystem/createPatchDict, users can group surfaces into functional patches as required. These definitions are then recorded in the `constant/polyMesh/boundaries` file, serving as the foundation for subsequent boundary condition assignments.

3.3 nTopology

Due to the implicit characteristic, both the geometries and performances are depending on the mentioned simple implicit functions. Most of the current TPMS need to be discretized as mesh mod- els for visualization or additive manufacturing. Therefore, the geometry was generated using nTopology, an engineering software tool used to produce simulations, additive manufacturing, and lattice geometries. Oftentimes, in fact, TPMS geometries are convenient thanks to additive manufacturing technology. The core difference from other traditional CAD tools is that it is based on implicit modeling. This allows for easy production of geometries based on TPMS and other lattices. From the implicit body, a mesh is then produced, which can be exported as .stl and subsequently used, in this case, for CFD simulations. This core functionality of nTopology also allows for easy production, repetition, and rapid modification of workflows, Fig.3.3.

Particularly important is the Mesh from implicit by DC node, which allows for the generation of perfectly straight cell boundaries thanks to the preservation of the sharp edges. In fact, this is otherwise not guaranteed due to the implicit nature and fitting of the geometry and mesh. Due to the cyclic nature of these simulations, as will be explained later, during this work mismatching of cyclic neighbor surface patches has caused convergence issues. The mid-surface offset allows for effective control of the resulting porosity of the cell. A correct generation of the mesh for the .stl is based on a balanced combination of values for the adaptivity and Min. Feature Size. The re-meshing mainly helps reducing the file size of the generated geometry by simplifying the mesh.

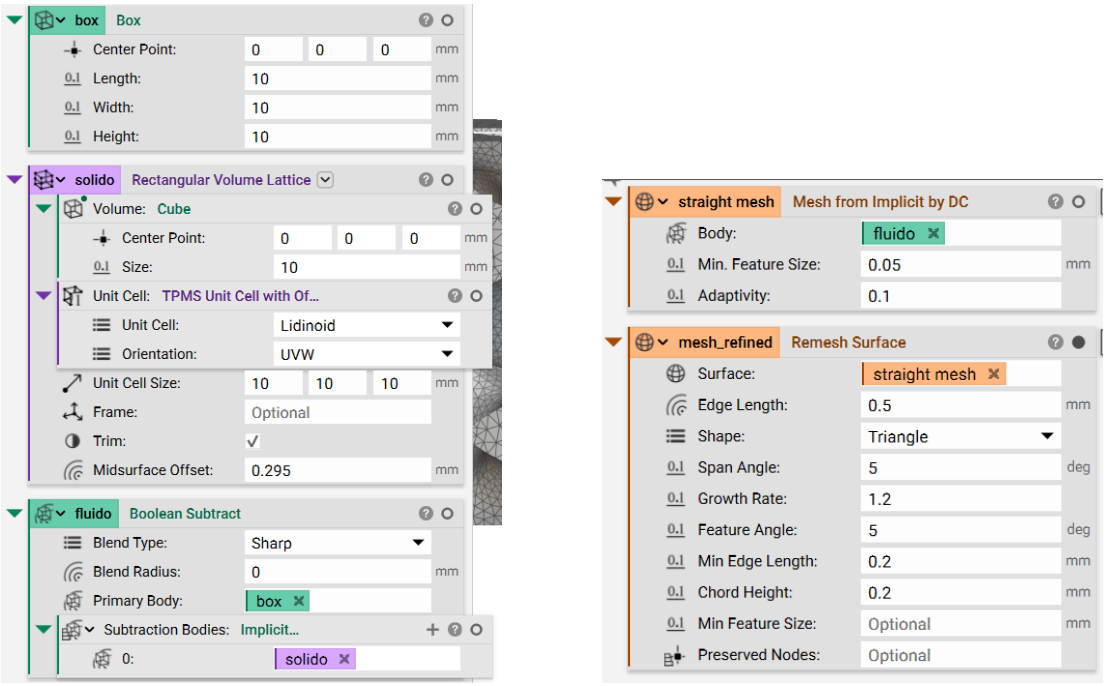


Figure 3.2: nTopology workflow

Chapter 4

Simulation set-up

The laminar transient simulations have all been performed through the pimpleFoam solver's "PISO mode", in order to use a less robust, but faster algorithm. Particular stability was not needed due to the low Reynolds numbers, and convergence has been successfully archived during all the simulations performed. A pure laminar setup was intentionally chosen over standard RANS turbulence models (such as $k-\epsilon$). At low Reynolds numbers, RANS models tend to overpredict turbulent energy, introducing an artificial viscosity that unphysically dampens and flattens out actual flow oscillations and vortex shedding. Operating without a turbulence model allowed us to capture these macroscopic fluctuations based purely on the exact Navier-Stokes equations. The following sections present the specific case study and detail the methods used for spatial and temporal discretization.

4.1 Laminar model

In the case of laminar simulations, on OpenFOAM this consists in applying the continuity and momentum equations as shown (2.2) (2.1), but with the turbulent kinematic viscosity ν_t set to zero. This variable is normally used in turbulent models to approximate turbulent phenomena, and used to express the effective kinematic viscosity as a sum between the actual fluid kinematic viscosity and the turbulent component(4.1):

$$\nu_{eff} = \nu + \nu_t. \tag{4.1}$$

The `simulationType` for laminar needs to be specified in the folder `constant/turbulenceProperties`. Additionally, it needs to be mentioned that laminar models do not use wall treatment functions related to $y+$. Therefore, no approximations are done when solving the near-wall flow.

4.2 Time model

The simulations presented in this work will be of both steady-state and transient nature. While the boundary conditions were not changed and are constant in time, the geometry produces intrinsically unstable flows at a high enough Re, as explained earlier. For this reason, pimpleFoam was employed in PISO mode, an Euler time scheme was used. It is denoted under `ddtSchemes` on the `fvSchemes` dictionary. The Euler scheme's properties are the following:

- Implicit
- First order

and is applied on OpenFOAM as

$$\frac{\partial}{\partial t}\phi = \frac{\phi - \phi_0}{\Delta t}. \tag{4.2}$$

The Euler scheme was chosen for its stability, in particular for being unconditionally stable resulting from the implicit method. This is especially important due to the unstable flow conditions analyzed. In addition, a time-step control was employed to keep the CFL number < 1, where

$$Co = \frac{u\Delta t}{\Delta x}. \tag{4.3}$$

In Eq. (4.3), Δx represents the mesh dimensions and Δt the time discretization used. The CFL condition dictates that the distance a fluid particle, and the information it carries, travels within a single time step must not exceed the characteristic spatial discretization scale of the local mesh cell. This restriction ensures that flow properties propagate exclusively to immediately adjacent elements during Δt, therefore preserving the temporal accuracy and numerical stability of the transient simulation. For this reason, although the CFL number varies cell by cell, the maximum was set at 1. At initialization, a $\Delta t = 0.0015s$ was imposed.

4.3 Geometry

The porosity chosen for this thesis work was $\phi = 50\%$ due to the possibility of printing it without problems, and as a way of making the study the most general, therefore, without considering too closed or too open geometry. In this way, it can be used as a starting point for further fluid dynamic analysis of Lidinoid TPMS at low Re values.

An interesting characteristic of this geometry is the presence of choke-points, which are absent at lower porosities, thus changing drastically the fluid dynamics

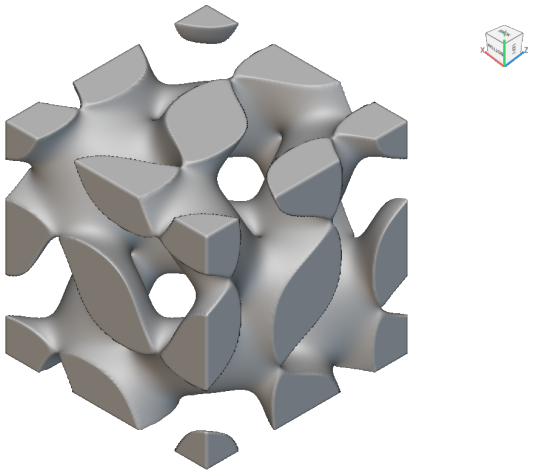


Figure 4.1: Lidinoid of $\phi = 50\%$

involved. The dimensions of the cell cube, which represents the base for a possible scaffold, has dimensions $10mm \times 10mm \times 10mm$. This size has been often used in papers as base cell size.

4.4 Boundary conditions

The boundary conditions applied to the Lidinoid cell were the following

- Imposed flow equal to $U_p = 0.001m/s$ in the x direction
- Cyclic conditions on all the boundaries
- noSlip condition imposed on the walls.

Absolute pressure was not relevant for this study and was uniformly imposed to zero.

MomentumSource

The flow imposition was set through the use of the `meanVelocityForce`, which enforces the pore velocity U_p or $\overline{U}$. This is obtained by the solver through a pressure gradient, which oscillates until it reaches a converged final value that allows for the velocity given the conditions of fluid and geometry. However, this BC causes the real calculated pressure drop, computed using functions in the `controlDict` dictionary, to go to about zero. This is due to the uniformity of momentum imposition used. Instead, this value needs to be sought in the imposed `gradP` that

this BC sets, which is the value that will be in fact used during this work.
In order to impose a constant flow it is also possible to impose a fixed pressure jump from inlet to outlet. This BC resolves the issue of having a zero pressure gradient in an infinite medium, such as a completely cyclic BC would generate. However, unlike the used BC, this condition makes it hard or, in any case, laboursome to precisely tune the simulation to the desired Re number. Therefore, the meanVelocityForce BC was selected.

Cyclic BC

This condition was imposed on all sides of the cubic cell. It requires the specification of a source boundary and a target boundary or, better, a `neighbourPatch` and the distance between these boundaries. In this way, OpenFOAM superimposes the source on the target to virtually have a continuation of the geometry in a cyclic fashion.

4.5 Mesh generation

For the numerical simulations performed in this thesis, a cartesian mesh was employed. Using nTopology a negative of the original geometry was produced, which was then imported on OpenFOAM 4.2. In this way, a better control on the domain to be meshed was possible. It is important to note that no boundary layers were used, due to lack of need to approximate dissipation and other values at the wall, which only apply in the case of turbulent models. The cartesian mesh was used because of the generally high quality of the mesh produced, particularly for the case of non-orthogonality.

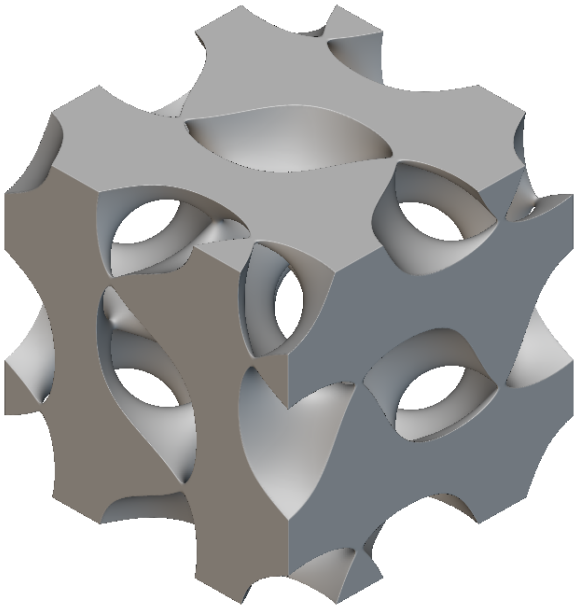


Figure 4.2: Fluid volume of the geometry

The conditions imposed to the mesh which has been used were the following:

- `minCellSize` = 0.0007 m
- `maxCellSize` = 0.0007 m
- `boundaryCellSize` = 0.0007 m
- `additionalRefinementLevels` = 4 applied to all the walls.

The mesh thus generated, satisfies quality checks of OpenFOAM and general good practice:

- Non-orthogonality - Max = 63.3 - Avg. = 7.06
- Max skewness = 1.23
- Max aspect ratio = 9.44
- Max cell openness = 3.5e-16 .

The resulting mesh can be observed in Fig.4.3, 4.4, 4.5. The section of the mesh was obtained through the slice filter on ParaView, which has been used to observe the meshing process and results during this work. In particular, the slices have

been performed parallel to what would be the flow direction x, off center by 0.0025 m in order to visualize the meshing along what will be the main flow in the results.

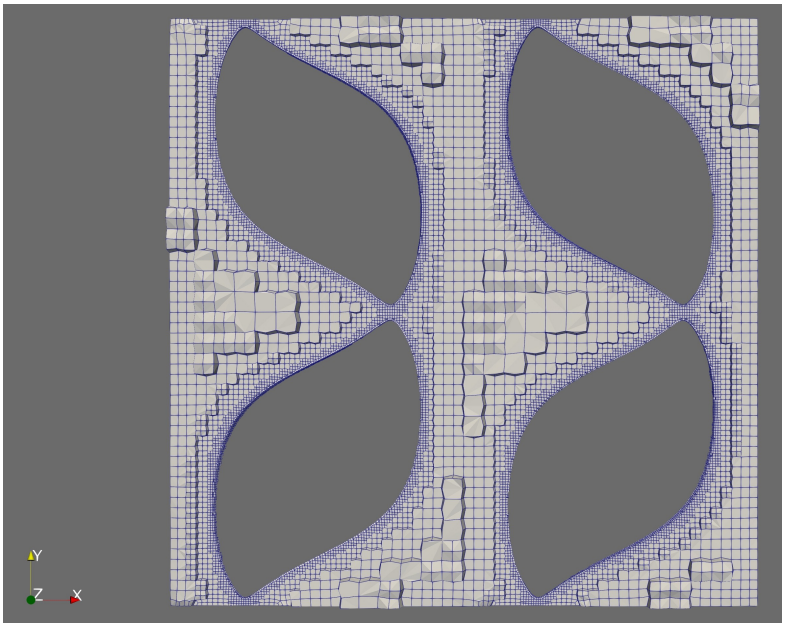


Figure 4.3: Off-plane view of section captured at z = 0.0025 m

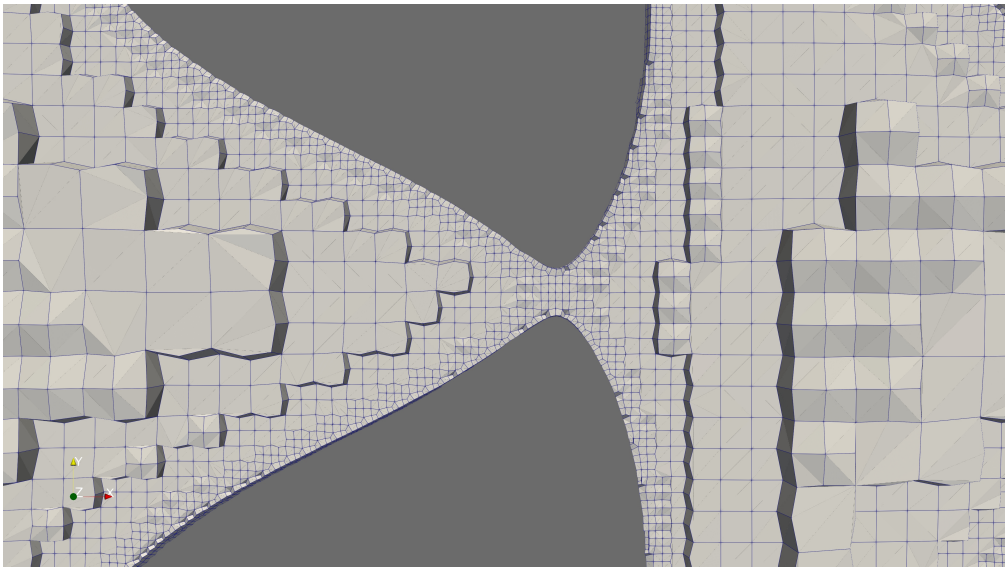


Figure 4.4: Close-up of the chokepoint in the geometry

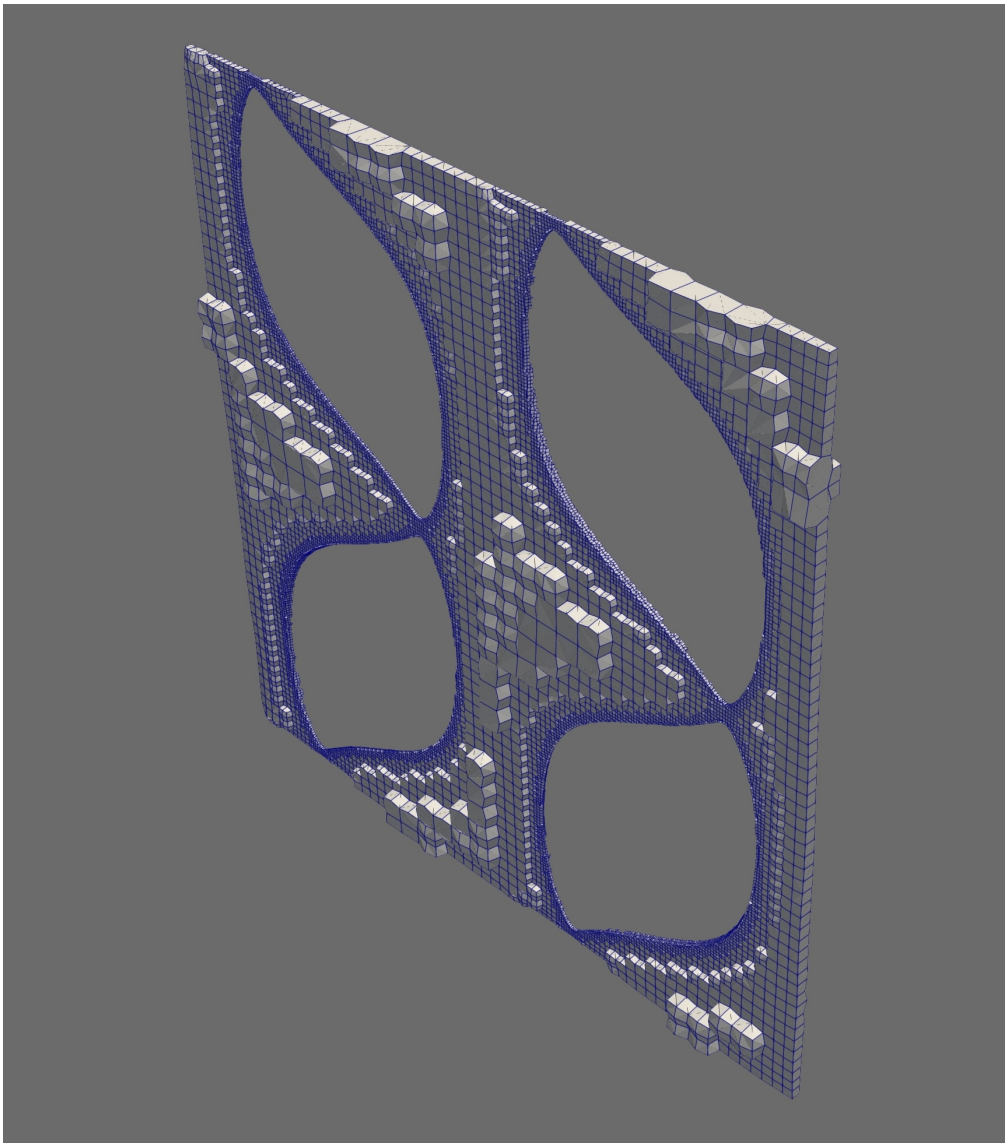


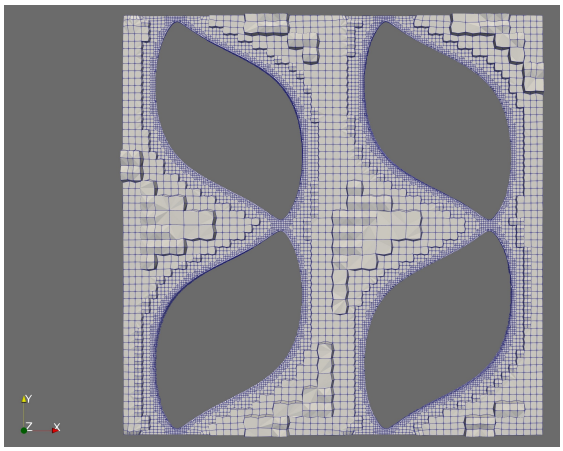
Figure 4.5: Isometric view

A critical meshing area is the choke-point, which can be observed in Fig.4.4. In this case, it has been adequately meshed.

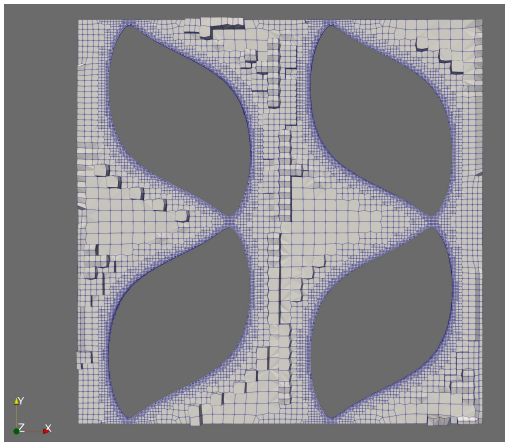
4.6 Convergence study

To verify that the numerical solutions are independent of the spatial discretization, this section details the results of a mesh sensitivity study. The mesh density was incrementally refined across the domain to evaluate the specific impact of grid

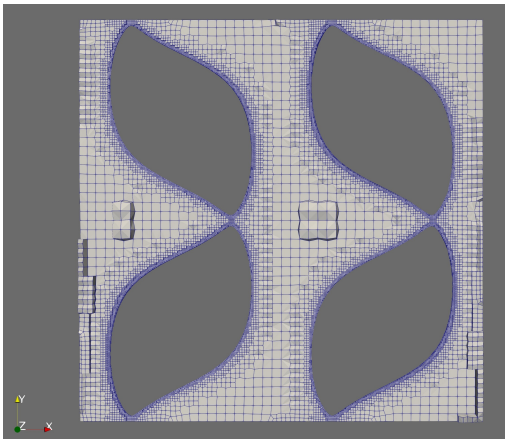
resolution on the calculated variables.



(a) Coarse configuration



(b) Medium configuration



(c) Fine configuration

Figure 4.6: Main caption for the triangular layout.

	Base size [mm]	Number of cells
1 FINE	0.46	3498434
2 MEDIUM	0.56	2360510
3 COARSE	0.7	1528609

Table 4.1: Characteristics of the meshes

The study was performed, most importantly, at flow regime of $Re = 150$. This choice was made to ensure credibility to the convergence results, being the flow at such Re in a crucial regime change region, as will be demonstrated in the thesis'

results chapter. The mesh for this study was reduced by reducing the base size. This is the most effective way of maintaining consistency between various meshes, given that the refinement at the wall boundaries is based on the base size, as mentioned earlier. The variables will then be compared, and the Grid Convergence Index (GCI)(4.11) will be calculated. Given the data from the meshes used in the convergence study, it is an instrument to indicate the precision of the results. The GCI itself is reported as a percentage and represents an error band around the fine-grid solution. A GCI of 2 % means that the results is, with a probability of 95%, within 2% of the infinitely fine grid result. This value can be approximated, having three grids, through the Richardson Extrapolation, which will be explained later.

$$GCI_{fine} = \frac{F_s \mid \epsilon \mid}{r^p - 1} \tag{4.4}$$

In Eq.(4.11), F_s represents a safety factor, equal to 3 in this case, ϵ (Eq. (4.8)) is the error in the results between the finest and coarsest mesh used in this work, r is the ratio, which relates the number of cells per volume of each configuration, and p represents the order of convergence of the study. Three meshes were studied as can be seen in Table 4.5, which is best practice to represent a more correct order of convergence of the simulations' results. In fact, p it is defined as

$$p = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)} \tag{4.5}$$

where the value of p was calculated through an iterative process. This is due to not having a constant ratio of mesh refinement through the three configurations. For the iterative process was therefore used, starting from guess values $q = 0$ and $p = 1$

$$p = \left[\frac{1}{\ln(r_{21})}\right]\left[\ln\left|\frac{\epsilon_{32}}{\epsilon_{21}}\right| + q(p)\right] \tag{4.6}$$

where

$$q(p) = ln\left(\frac{r_{21}^p - 1 * sign(\epsilon_{32}/\epsilon_{21})}{r_{32}^p - 1 * sign(\epsilon_{32}/\epsilon_{21}}\right) \tag{4.7}$$

and $\epsilon_{21} = f_2 - f_1$. To be distinguished from the absolute error defined as

$$\epsilon = \frac{f_2 - f_1}{f_1} \tag{4.8}$$

that is used for the GCI, where f_1, f_2, f_3 are the variable results at each mesh density. r is generally

$$r_{21} = \frac{h_2}{h_1} \tag{4.9}$$

where 1 in this case represents the finest mesh and 2 the coarsest. h is the grid spacing so defined:

$$h_n = \left[\Sigma_{i=1}^{N} \frac{\Delta V_i}{N_{cells}} \right]^{1/3} \tag{4.10}$$

with the total volume for the fluid equal to $4.997 \cdot 10^{-7} mm^3$, and where n represents the generic mesh configuration.

In this study, the GCI for the coarse grid was calculated and used in the reporting of the final results. This was done due to the high number of simulations required to make the needed observations related to the change of flow regime, as well as the computational time required for the convergence with the fine grid. For this, a few changes need to be made to the GCI definition:

$$GCI_{coarse} = \frac{F_s \mid \epsilon \mid r^p}{r^p - 1} \tag{4.11}$$

And is therefore related to GCI_{fine} as follows

$$GCI_{coarse} = r_{21}^p \cdot GCI_{fine}. \tag{4.12}$$

The aforementioned variables have thus been defined with respect to the grids as

	Values
h_{fine}	$5.23 \cdot 10^{-5}$
h_{medium}	$5.96 \cdot 10^{-5}$
h_{coarse}	$6.89 \cdot 10^{-5}$
r_{21}	1.140
r_{32}	1.156

Table 4.2: Grid refinement values

The mesh is best refined so that $r = 2$, what is called doubling. However, limits in hardware and computational time made only lower, but sufficient ratios > 1.1, possible.

Richardson Extrapolation

The Richardson extrapolation is a methodology of approximating through the grids results an "exact" solution, or rather, what would be an exact solution based on the solutions results. It assumes the discrete solutions to have a series representation

$$f = f_{exact} + g_1 h + g_2 h^2 + g_3 h^3 ... \tag{4.13}$$

which, using the grid ratio (4.9) becomes

$$f_{exact} \cong f_1 + \frac{f_1 - f_2}{r_{21}^p - 1}. \tag{4.14}$$

Although it is used usually with a $r = 2$, a grid doubling is not required. In this case, the ratio and results of the medium and fine configuration were used. The solution is not conservative [**?**], as the error based on this method has been proven to be less conservative than the GCI. However, it has been added for completeness.

Grid Convergence Results

For the calculation of the GCI, the pressure drop was used. However, other data has also been observed. Of course, it is essential that the results used for this study are not purely punctual results, but rather averaged outputs, such as pressure drop or tortuosity, which are based on the averaged shear stress on the walls. For the pressure drop:

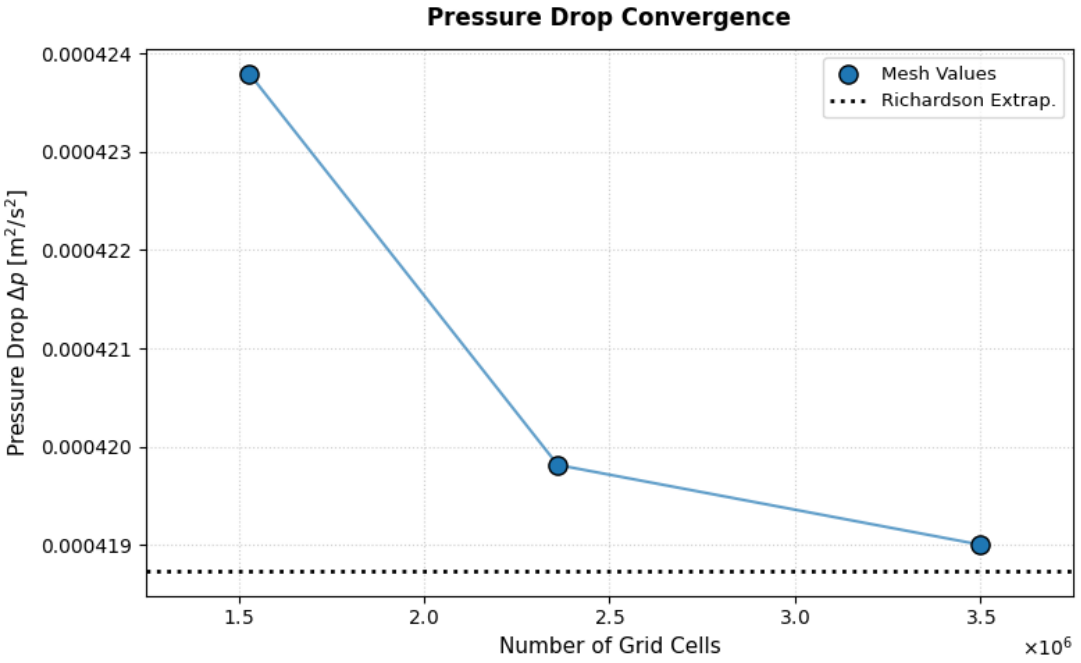


Figure 4.7: Δp convergence

GCI_{fine}	0.16%
GCI_{coarse}	0.47%

Table 4.3: Grid Convergence Index results for Δp

Where, as previously mentioned, GCI_{coarse} will be the one used in the results of this thesis.

The GCI values for both the coarse and fine mesh are more than satisfactory. Such certainty could be archived due to the sufficiently fine coarse configuration. A high value for p was obtained due to the converged results, which is also indicated by the low GCI values. At very low variations of the results as the mesh is refined, p tends to have not as realistic values. Otherwise, the theoretical convergence of

the numerical methods used in this thesis would be $p = 1$. It stems from having very little difference between the finest and the medium mesh results.

Then for tortuosity:

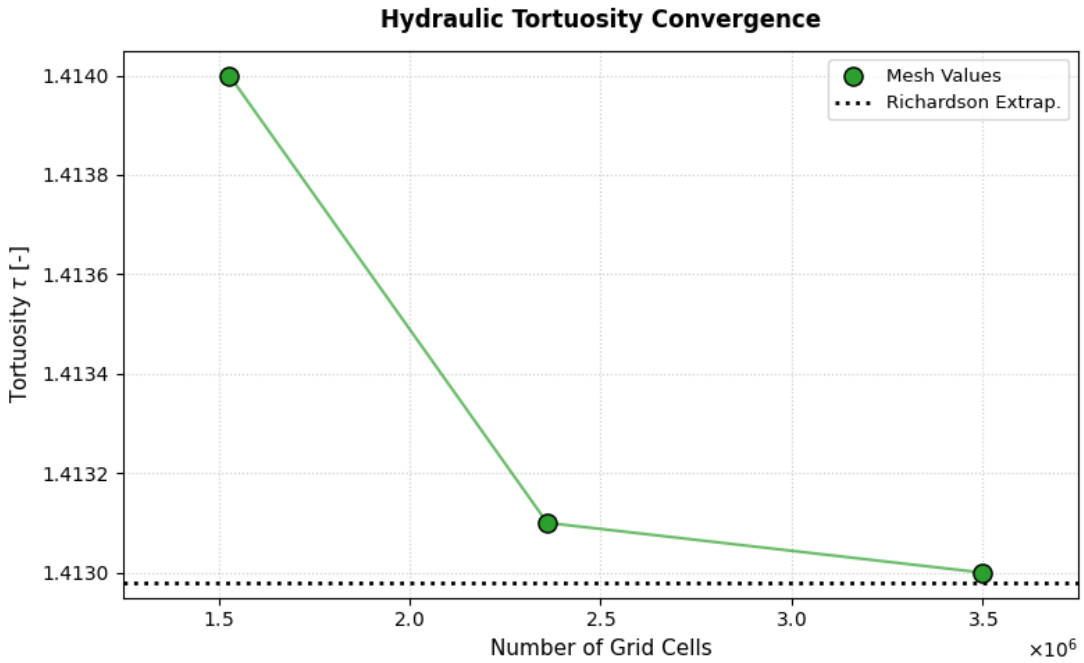


Figure 4.8: τ convergence

GCI_{fine}	0.0035%
GCI_{coarse}	0.0247%

Table 4.4: Grid Convergence Index results for τ

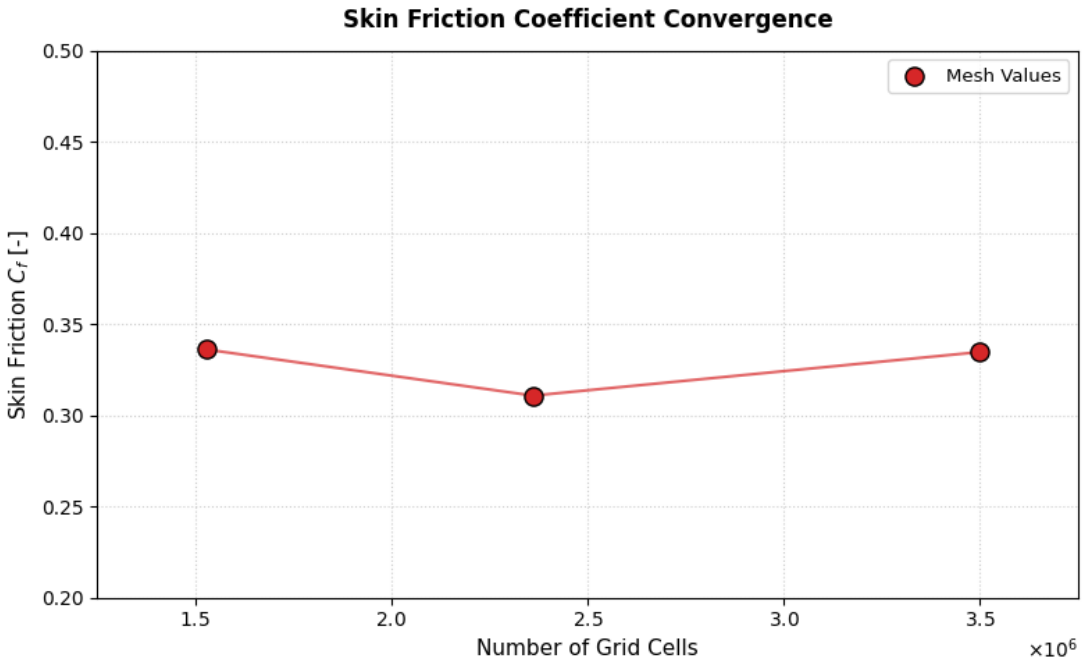


Figure 4.9: Grid Convergence Index results for C_f

GCI_{fine}	3.9494%
GCI_{coarse}	5.2046%

Table 4.5: Grid Convergence Index results for C_f

As can be seen, the skin-friction coefficient does neither decrease nor increase. A better refinement of the walls may have indicated a more noticeable trend. This may also be caused by the lack of $y+$ functions that approximate the values at the wall, and therefore a finer mesh may be required at the surface of the wall due to the laminar model used. However, the GCI is within acceptable limits and this results has been deemed acceptable, considered that a clear convergence is detected for the other variables used in the study. The main side effect may be a lower pressure drop than what would be detected in a validation study. However, this does not affect the objective of this study.

It may then be interesting to see the consistency of results of the TI, of which an actual convergence study was not performed due to the nature of the data acquired, that is highly dependent on the mesh discretization.

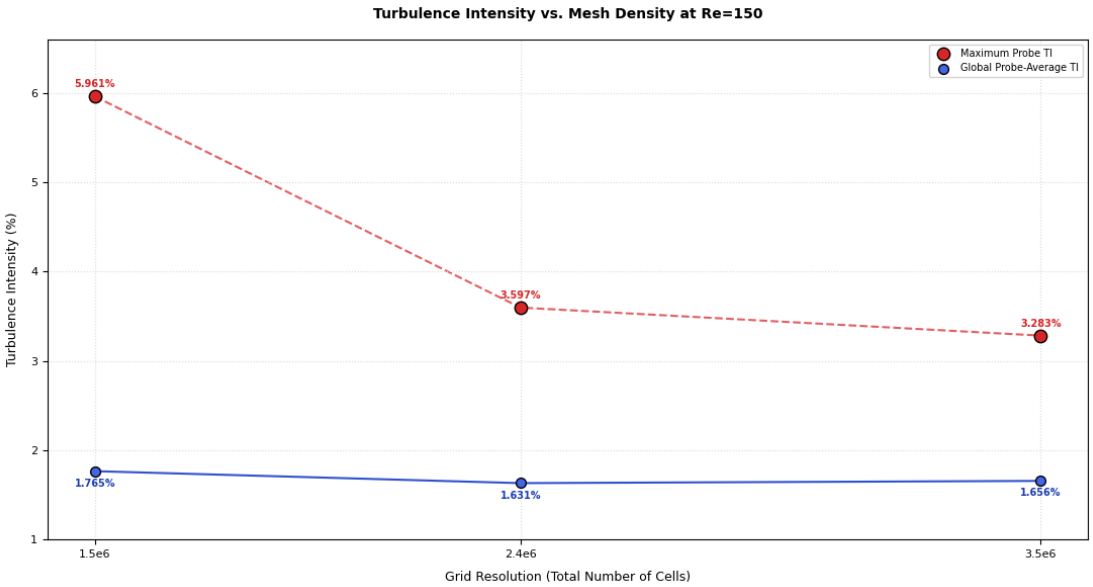


Figure 4.10: TI average and maximum values for the studied configurations

In Fig.4.10 can be seen that the TI average value remains constant throughout the meshes. The maximum fluctuation values demonstrate a clear asymptotic convergence trend, dropping on the coarse mesh to a stabilized value on the fine mesh. This sharp decrease indicates that the coarser grids suffered from discretization errors and artificial cell-to-cell velocity jumps as the sharp vortex fronts passed the probes. Conversely, the global domain-averaged fluctuations remained highly stable (1.76% to 1.65%), confirming that while global energy conservation is captured on coarser configurations, spatial refinement is strictly required to resolve true local oscillation amplitudes accurately. From these results it is clear that using the coarse configuration is sufficient.

4.7 Methodology

The simulations were performed at various Re numbers in order to be able to subsequently make a comparison between the results. Variables such as tortuosity, skin-friction coefficient, friction factor, and TI will be compared to various results at varying Reynolds numbers. In particular, the methodology proposed experimentally by Seguin et al [17] [18], and then used recently numerically by Vhora et al [20]

will be applied. The Reynolds number has been varied during this work by keeping constant the velocity U_p, or $\overline{U}$, and changing the kinematic viscosity ν. This can be controlled in the `constant/transportProperties` dictionary in the case folder. The simulations were performed using 7 cores in parallel. The domain decomposition was performed through the hierarchical method, decomposing it as n(7 1 1), thus maintaining in a single core the inlet and the outlet. This was done due to errors arising when divided. How the data for the study was acquired will be presented in the following section.

4.8 Data acquisition

For the analysis of the simulations that were performed various data were sampled. For most, the functions of OpenFOAM, which need to be specified in the `controlDict` dictionary, were employed. In particular, data was sampled at each time step and saved in a text file, which then was post-processed using a Python script. For what concerns the pressure drop, `gradP` as output of the log, was used as explained earlier.

Turbulence Intensity TI

Probes can be set at any given location of the geometry. At these points, data can be sampled, having specified the field of interest. In the case of these simulations, 10 probes were placed along the stream-wise direction, as shown in Fig.4.11 following what was done in [20] to study the turbulent intensity. At each time step, the velocity in the three directions for each probe was saved.

The precise locations of the probes are as listed:

```
    \texttt{probeLocations } \\
\texttt{(-0.0045 0 -0.0025)}\\
\texttt{(-0.0035 0 -0.0025)}\\
\texttt{(-0.0025 0 -0.0025)}\\
\texttt{(-0.0015 0 -0.0025)}\\
\texttt{(-0.0005 0 -0.0025)}\\
\texttt{(0.0005 0 -0.0025)}\\
\texttt{(0.0015 0 -0.0025)}\\
\texttt{(0.0025 0 -0.0025)}\\
\texttt{(0.0035 0 -0.0025)}\\
\texttt{(0.0045 0 -0.0025)}\\
```

The probes collected at each time step velocity data in the three directions. Through a post-processing set-up using Python, the turbulence intensity was

calculated for each probe using Eq.(2.10), then average and maximum values were calculated.

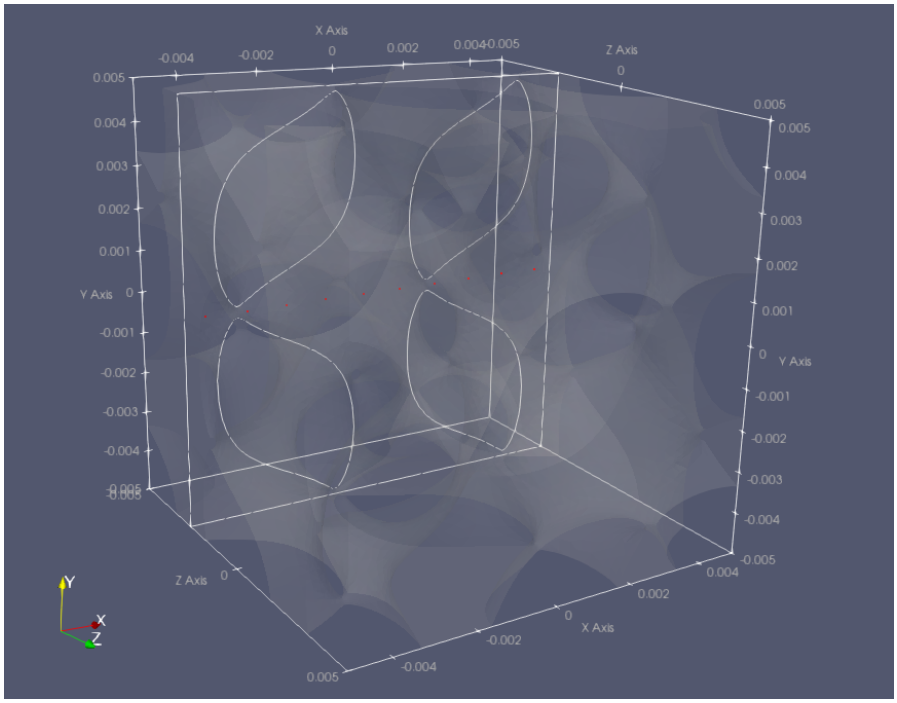


Figure 4.11: Probe locations

Skin-friction coefficient C_f

The skin-friction was calculated starting by the sampling at each time step of the wall shear stress, which is a recognized function type on OpenFOAM. To obtain a single value, and not having 3D shear stresses cancel each other due to the complex geometry, The average was performed on the magnitude of the sampled wall shear stress. It must be mentioned that the wall shear stress sampled by OpenFOAM is equal to $\tau_k = \frac{\tau_w}{\rho}$. Therefore, Eq.(2.9) would result in

$$C_f = \frac{\tau_k}{\frac{1}{2}U_\infty^2}. \tag{4.15}$$

This same change also applies to the friction factor, that from Eq.**??** it is expressed as

$$f = \frac{2D_h \Delta P}{L_c U_p^2}. \tag{4.16}$$

Tortuosity τ

The tortuosity was calculated as shown in Eq.(2.8). The data was sampled using the `volAverage` function for the velocity field in order to obtain the value of $\langle U \rangle$. Then, the magnitude of the U field was calculated using a `mag` type function, and then a `volAverage` of the magnitude of U was computed. Thus, for each time step a tortusoity value was sampled.

Chapter 5

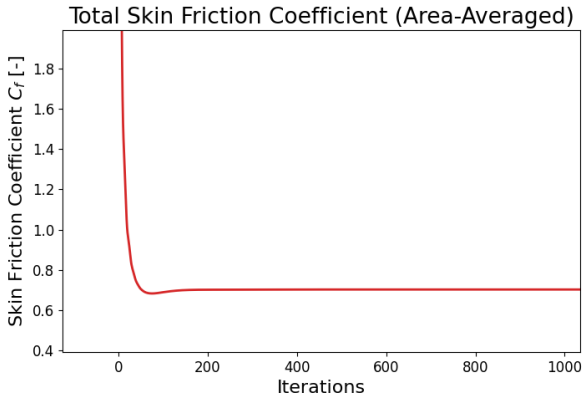
Results

In this chapter the results of the simulations that have been carried out will be presented and briefly discussed. In the representation of the data, it is important to note that the results have been non-dimensionalized in order to keep consistent results and allow replicability of the same, and comparison with other simulations that have different boundary conditions and dimensions. In particular, the velocity values obtained from the post-processing were divided by U_p, and the pressure values were divided by p_{ref}

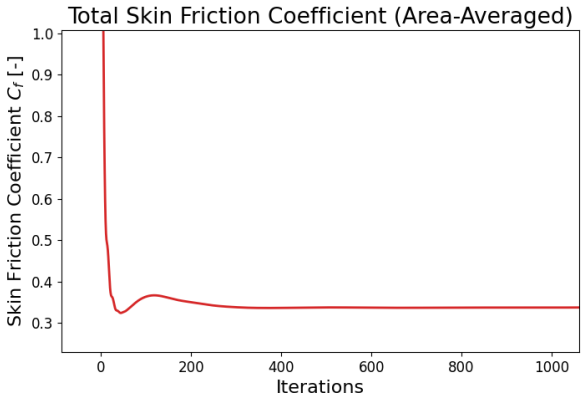
$$U_{ref} = U_p \tag{5.1}$$

$$p_{ref} = \frac{1}{2}\rho U_p^2 \tag{5.2}$$

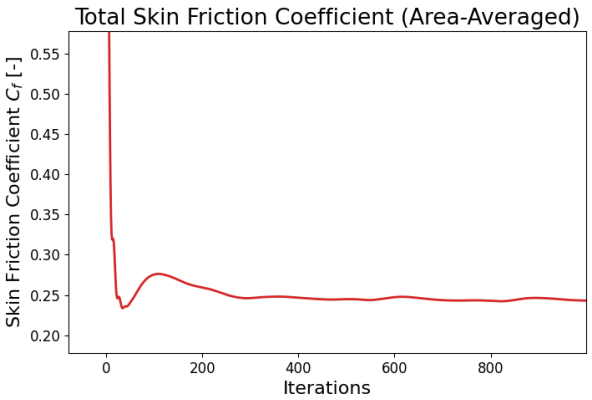
where ρ is removed due to the integration of the density constant performed by OpenFOAM in incompressible solvers, as mentioned. First, the steady state results will be analyzed, Fig.5.1.



(a) Re=50

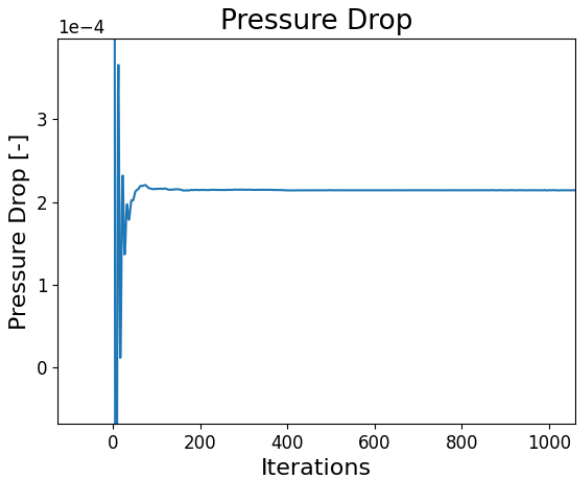


(b) Re=150

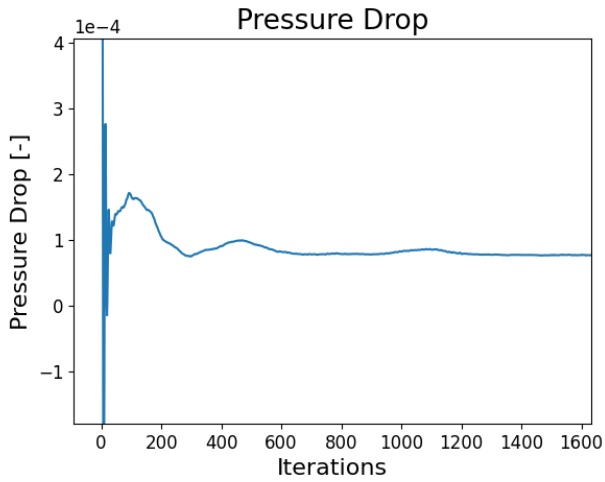


(c) Re=250

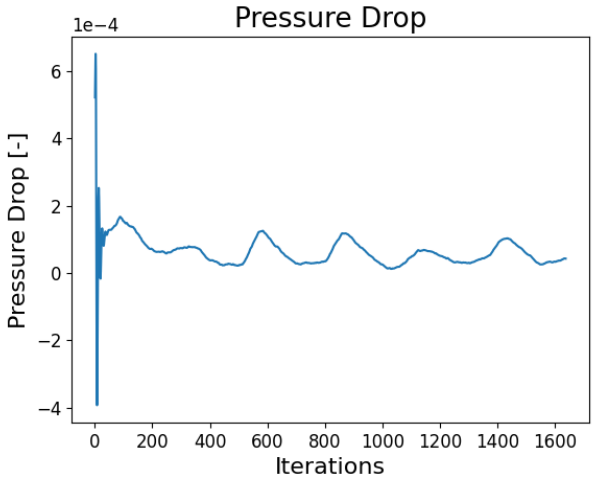
Figure 5.1: Skin-friction coefficient at different Reynolds numbers



(a) Re=50

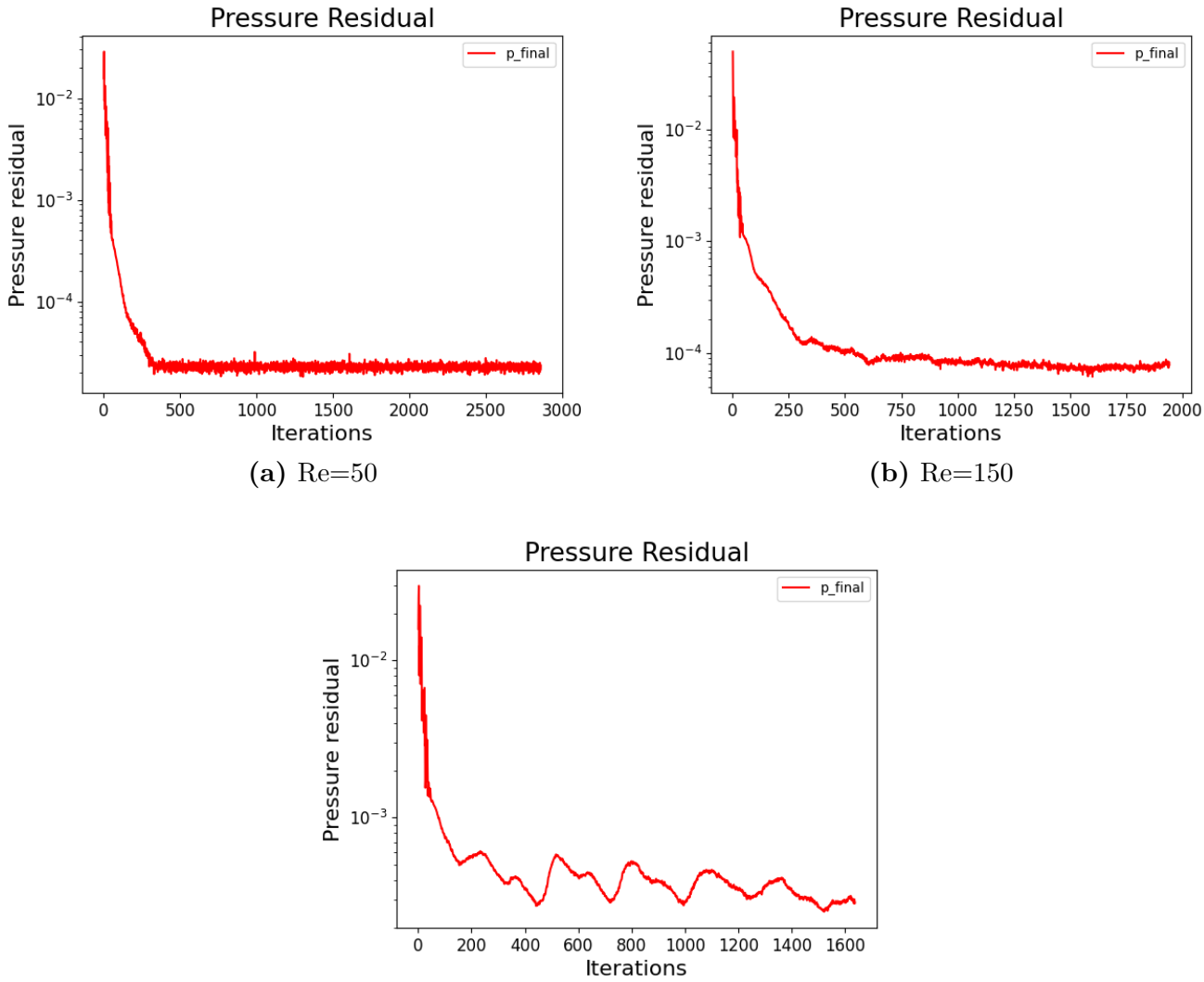


(b) Re=150



(c) Re=250

Figure 5.2: Pressure drop at different Reynolds numbers



(a) Re=50

(b) Re=150

(c) Re=250

Figure 5.3: Residuals at different Reynolds numbers

First of all, it can be observed that the skin-friction coefficient remains quite stable compared to the other results observed. This may be due to the large number of point values that are averaged to perform this final result at each iteration. In fact, the τ_k for each wall cell surface is used, producing a large pool of values. However, this is not necessarily true for all the averaged data that has been collected. In fact, the pressure drop, which results from the averaging of two surfaces contains less initial data overall and suffers, indeed, more oscillations and general instability as the Re number increases.

As can be seen in Figs. 5.1, 5.2 and 5.3, the simulations are quite stable

at Re=50, where the flow can still be assumed to be in the stable Forchheimer regime. In literature[15], the regime change is assumed to happen at around Re=10. However, the steadiness is usually maintained to about Re=150[17]. At 150 the results show increased instability, from which it could be already assumed that the flow has a component of instability in it. Much larger instabilities can be observed at Re=250, with non negligible oscillations for the pressure drop and residuals. Indeed, the residuals are also about an order of magnitude larger than in the previous simulations.

This marks a key difference with the following transient simulations that have been used, as the transient simulations are by definition more precise in describing intrinsically unsteady flows, and in general tend to average out small oscillations that may be detected in steady state simulations due to the division by time of the variation of results from one step to the next. In fact, steady state solutions assume that the solution when converged remains constant. However, the oscillating and varying nature of an intrinsically transient flow such as the one analyzed will not necessarily produce precise results. Additionally, the steady state will diverge or not converge when using Reynolds in the range of the hundreds. Furthermore, for the methodology used and explained earlier involving the TI, Eq.(2.10), the steady state cannot technically be used, as it is based on detecting the oscillations in velocity that cannot be represented by a steady state solver, which instead tends to detect generic and non physical oscillations of the unstable solver. However, if RANS were used this could have been possible even through the use of steady state solvers, as the definition for TI when using RANS, as k is calculated, is

$$TI = \frac{\sqrt{\frac{2}{3}k}}{U_p}. \tag{5.3}$$

Therefore, a number of transient PISO simulations were then performed in order to better observe these aforementioned unsteady behaviors. However, these have been run for all Reynolds number for consistency. The results that will be shown later are therefore only transient results. Following, three cases will be analyzed specifically to understand the evolution of the simulation and results and how they were then saved to produce the final comparison between Reynolds values. In particular, the cases will be at Re 50, 150, 250.

Re=50

As mentioned, at Re=50 the flow is in a stable Forchheimer regime. The flow is expected to be stable. As can be seen in the following Figs. 5.4 , 5.5 ,5.6, 5.7 , the results are satisfyingly steady. Convergence is reached sufficiently fast. In Fig.5.5, the pressure is shown to oscillate periodically. This may be due to slight instabilities produced by the periodic boundary conditions that are detected by the

transient solver, with respect to the previous results for steady state which did not show this kind of instability. Ultimately, the variation can be considered negligible.

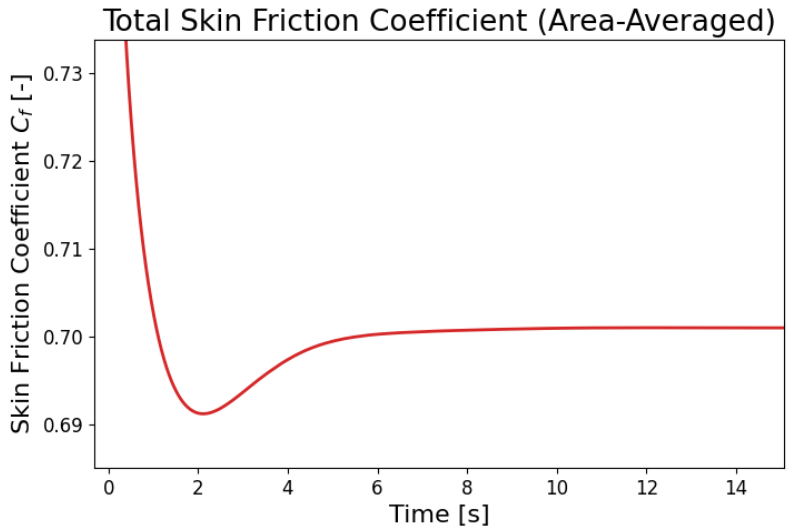


Figure 5.4: C_f convergence

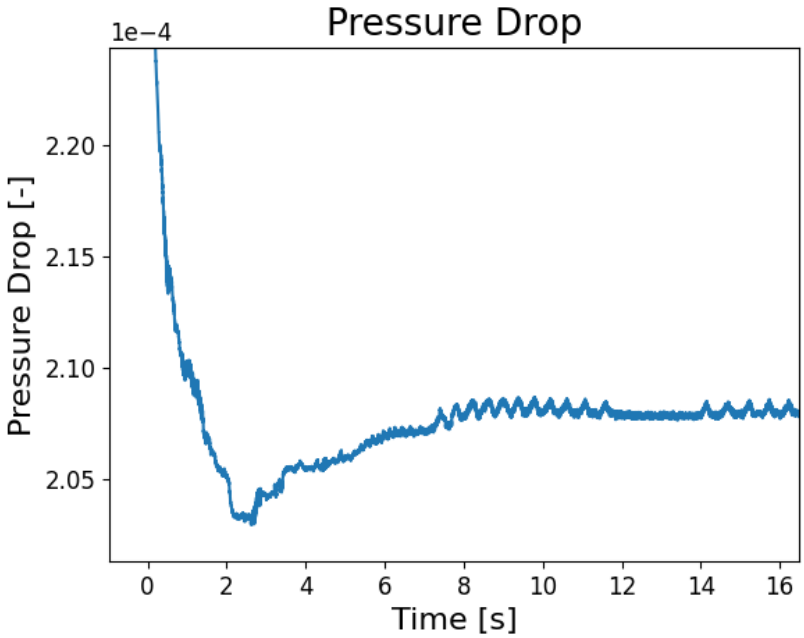


Figure 5.5: ΔP evolution

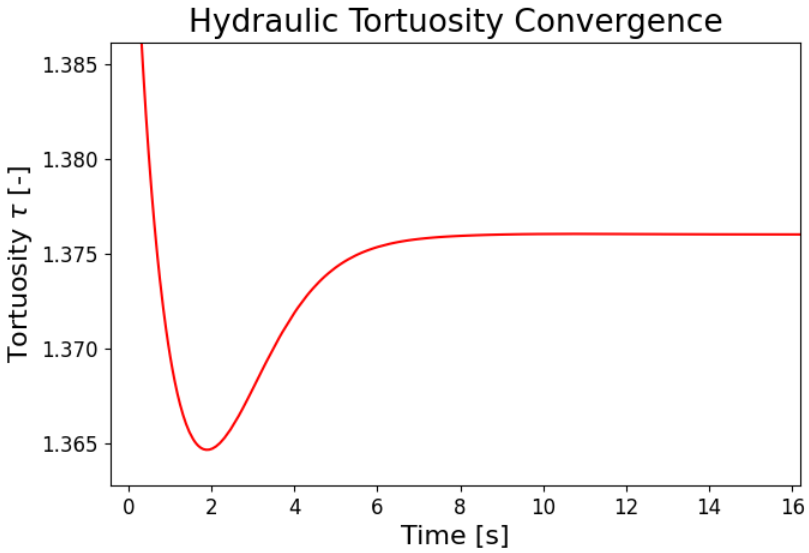
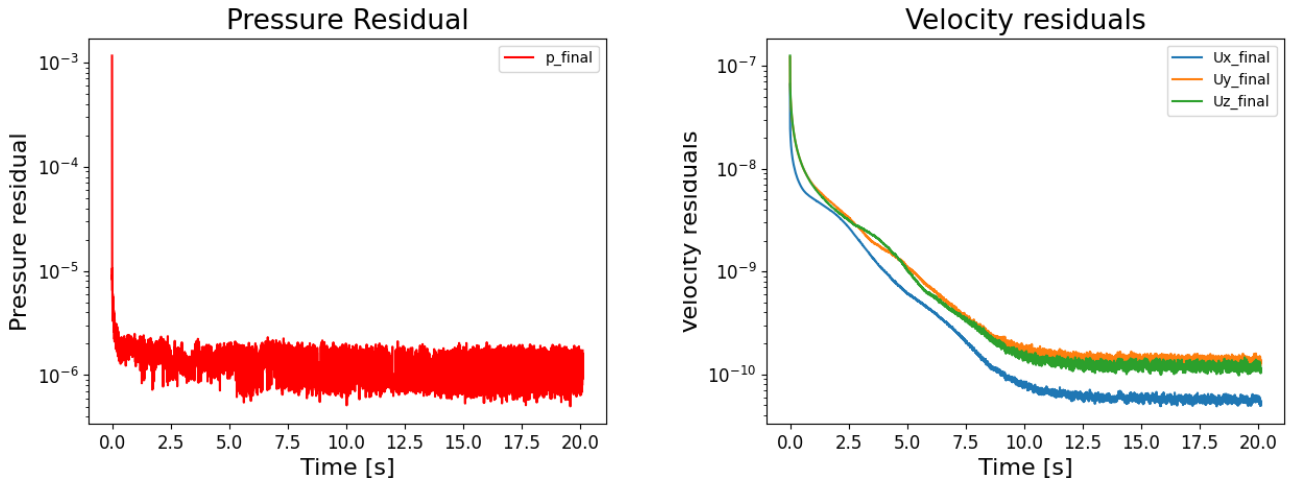


Figure 5.6: τ evolution



(a) Pressure residuals

(b) Velocity residuals

Figure 5.7: Residuals evolution

The residuals can be seen to reach a satisfactory order of magnitude $< 10^{-5}$ for pressure and $< 10^{-9}$ for velocity. These values are much lower than the steady state results and suggest a better convergence. In particular compared to the steady state Re=250 simulation residuals. The velocity may have lower residuals due to the imposed valocity BC, which may constrain the velocity rapidly at certain

values, while pressure is calculated and no values are imposed in the domain.

Re=150

The results shown in the following Figs.5.8, 5.10,5.9,5.11 have been obtained through the use of a converged transient solution at different Re number in order for the computational time to be reduced. Therefore, from the results it is not possible to appreciate the convergence evolution. Nevertheless, a similar convergence to the Re=50 simulation can be assumed, as the flow still relatively stable in time. Although some variations in time can be seen, compared to the steady state results for the flow at this Reynolds value, the result is quite more stable if the pressure drop is not considered. However, the pressure drop anticipates an intrinsic instability of the flow, which will be better shown in the next subchapter during the analysis of the TI. The slight periodic oscillations in the pressure drop presume a laminar unsteady flow, which is characterized by these as previously mentioned. Additionally, when the pressure drop is compared to the transient case Re=50, the difference is quite noticeable, as the pressure at Re = 50 had only slight spikes due to possibly some numerical oscillation. In this case the oscillations are much larger and a numerical cause can be excluded and attributed to the flow physics.

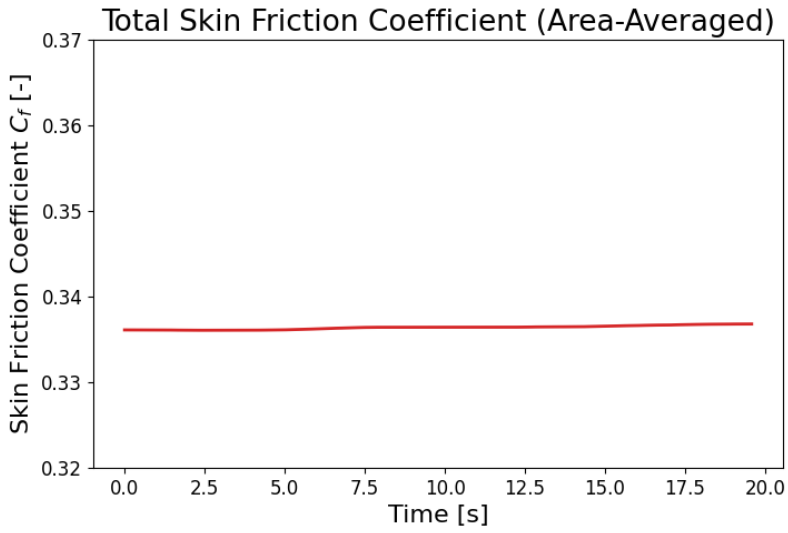


Figure 5.8: C_f convergence

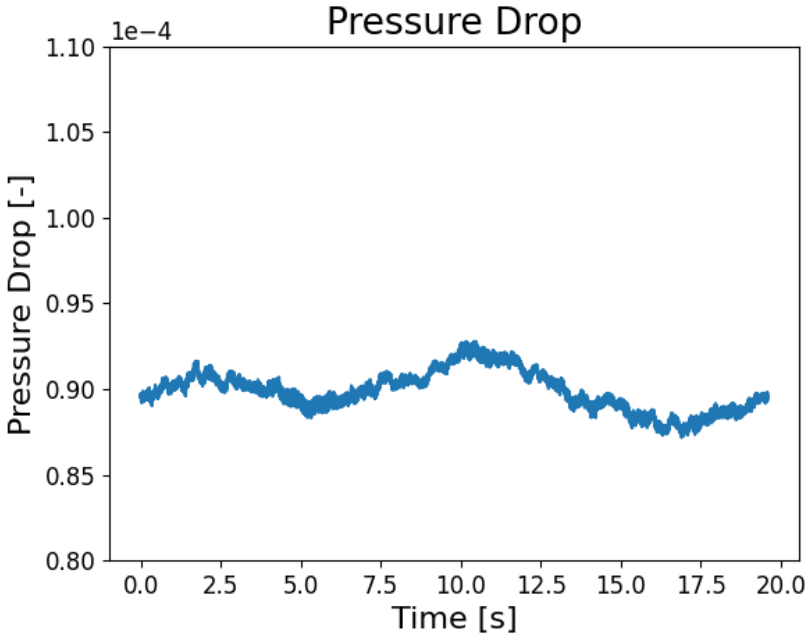


Figure 5.9: ΔP evolution

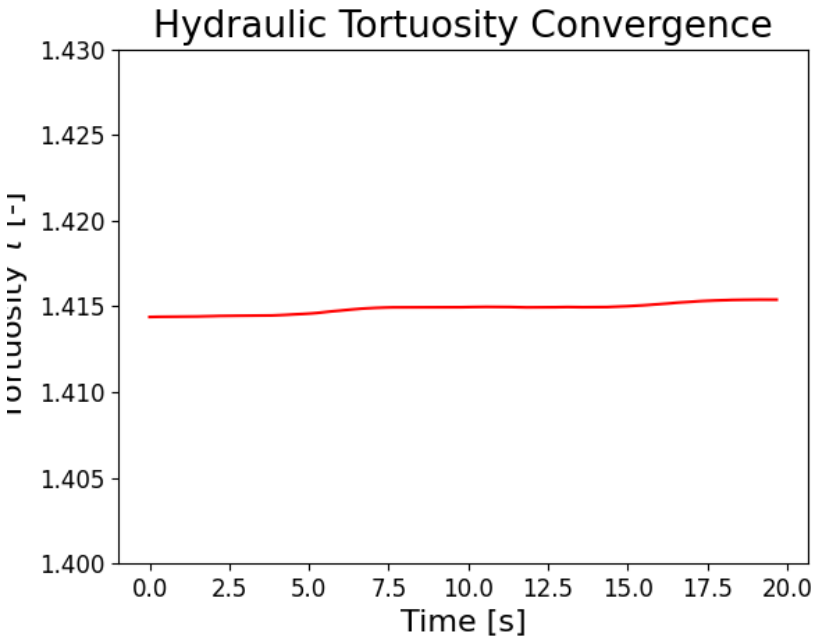
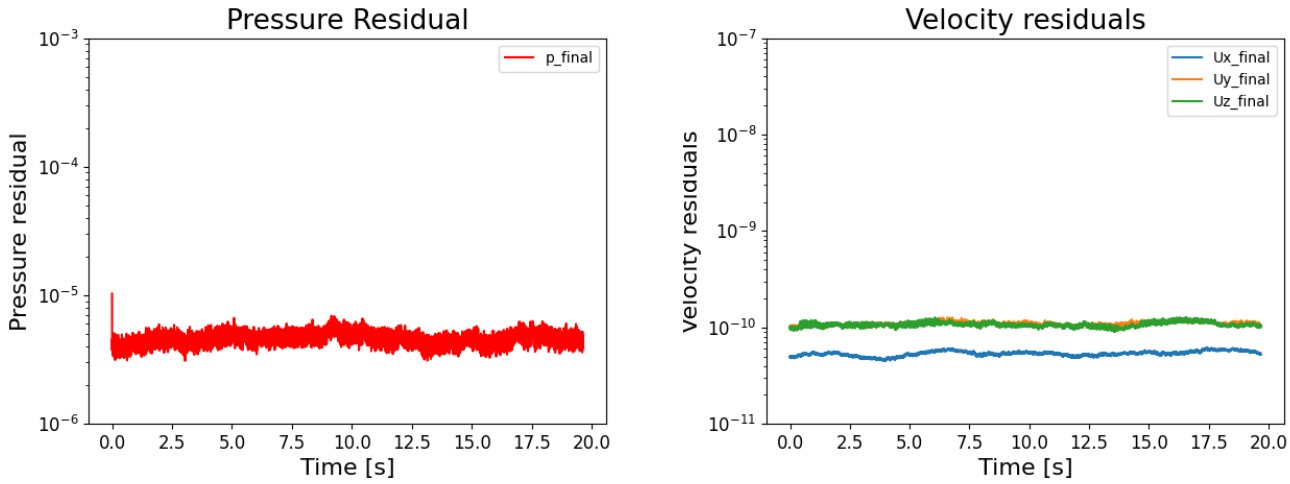


Figure 5.10: τ evolution



(a) Pressure residuals **(b)** Velocity residuals

Figure 5.11: Residuals evolution

The residuals also in this case reach satisfyingly low orders of magnitude, similarly to the previous case. Although relatively stable, some oscillations and instabilities are shown. Due to the instabilities in the steady state and in this case, it can be assumed that at this Reynolds the flow is at a crossroads region where it is becoming increasingly unsteady, and leaving the steady laminar Forchheimer regime.

Re=250

At Re=250 the flow has completely departed the steady Forchheimer regime. As can be seen in the following figures 5.12, 5.13, 5.14, 5.15, the flow is quite irregular even observing values that represent great averages of values, such as C_f and τ. The unstable results do converge to an approximate value. The oscillations present a converging periodic oscillation as shown particularly in Figs. 5.12, 5.14. This behavior, as Re increases further, is expected to increase.

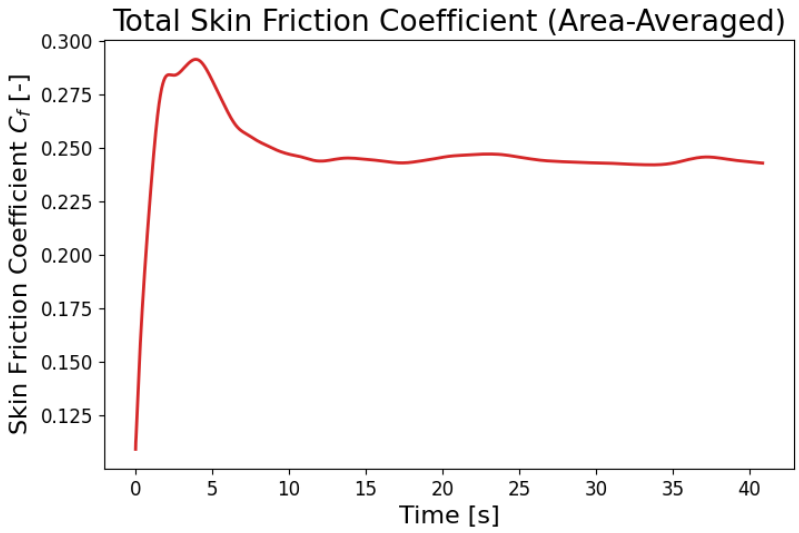


Figure 5.12: C_f convergence

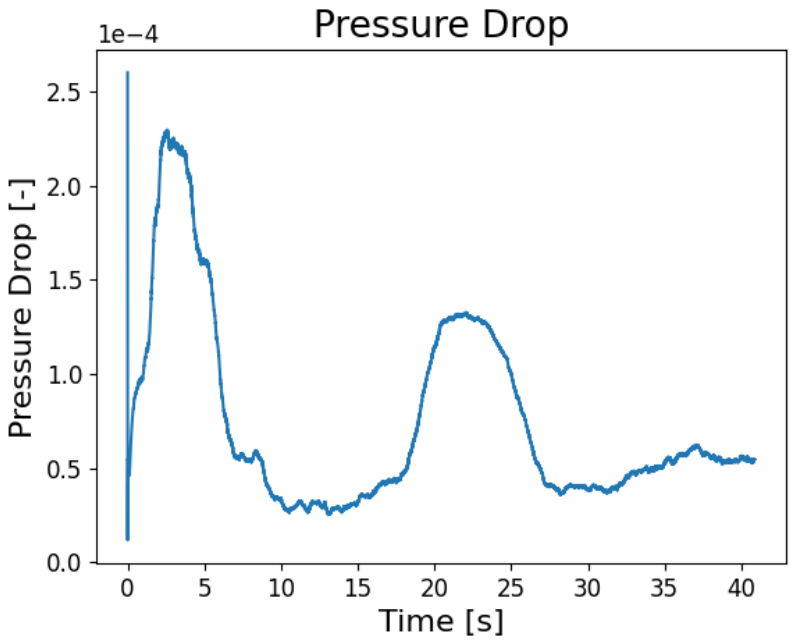


Figure 5.13: ΔP evolution

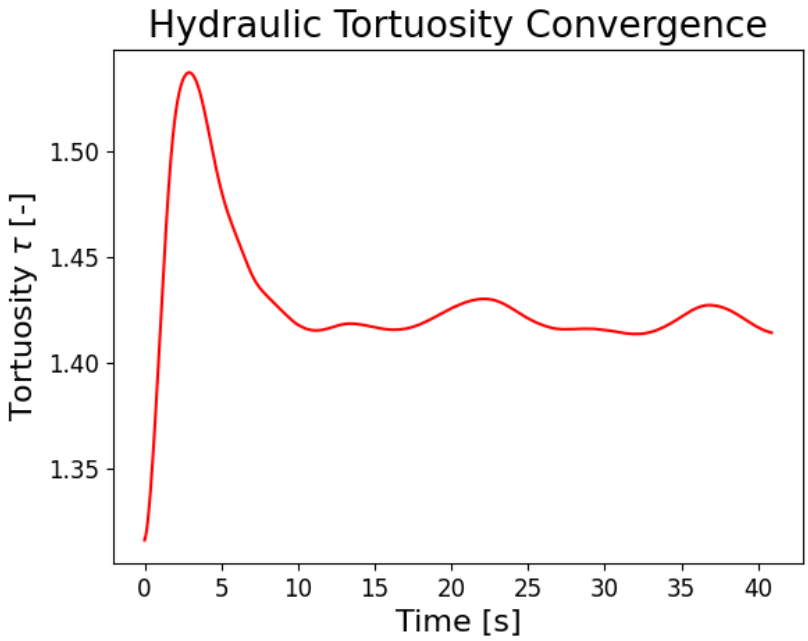
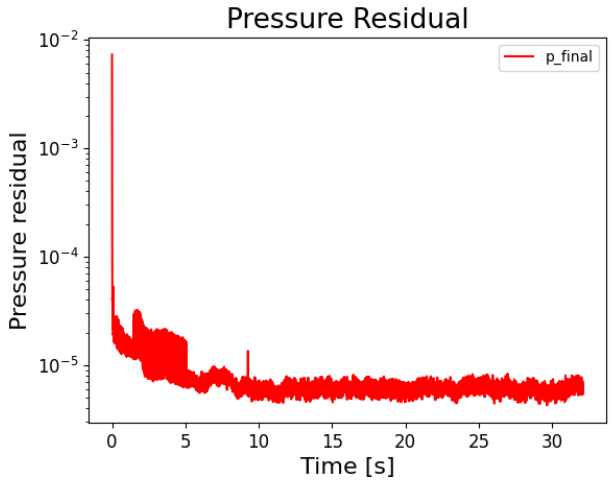
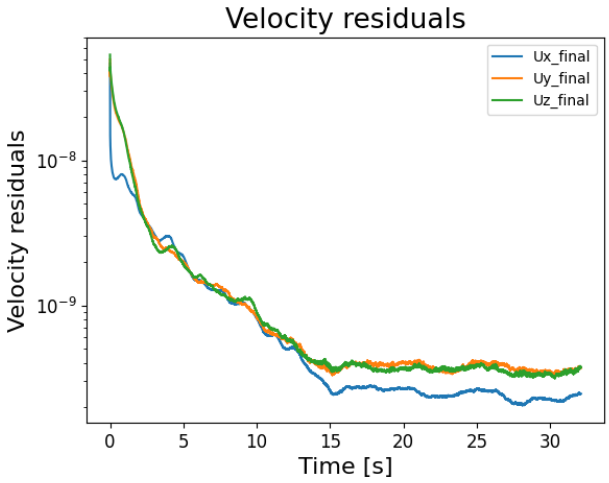


Figure 5.14: τ evolution



(a) Pressure residuals



(b) Velocity residuals

Figure 5.15: Residuals evolution

From the results, the most unstable is probably the pressure, which experiences high fluctuations and is testimony to the general instability of the flow at such Reynolds values. However, the residuals also drop to acceptable values, although

slightly larger. The velocity residual in particular, but the pressure one as well, can be seen to have more instabilities and a less smooth convergence compared, for instance, to the Re=50 case that was shown. This can be safely attributed to the general flow instability. Quite significant is the difference of two orders of magnitude for the pressure residual compared to the steady state simulation, which, together with the added stability of the transient residuals, demonstrates a better convergence overall.

Compared to the steady state simulations, the convergence is slower, which is to be expected due to the transient solvers being more complex than the steady state. They also require a much longer computational time.

The results can be expected to be closer to reality in transient solutions, in particular for higher Reynolds values as the intrinsic instability becomes less negligible. Although the final results for C_f at Re=250 do not change significantly between steady state and transient, it can be considered an exception due to C_f being generally quite stable in all the simulations that have been run during this work. Nevertheless, while steady state simulations can be preferred for surely steady simulations, transient simulations are safer in the sense that they should be preferred close to transition Reynolds values as the steady state solvers cannot handle well the instabilities.

5.1 Variables dependence on Reynolds number

In the case of the friction factor, the onset of the Forchheimer regime can be observed through it by comparing the obtained viscous-inertial-related friction and the friction factor depending only on the viscous pressure drop. The absence, or negligibility, of intertial forces characterizes the Darcy regime and is therefore considered as a meter of comparison. The viscous pressure drop is defined simply as the Darcy-Forchheimer pressure drop equation, Eq.(2.15), but with the removal of the inertial component resulting in

$$f = \frac{C_1}{Re_p} \tag{5.4}$$

where C_1 results from substituting the equation into the definition of the friction factor.

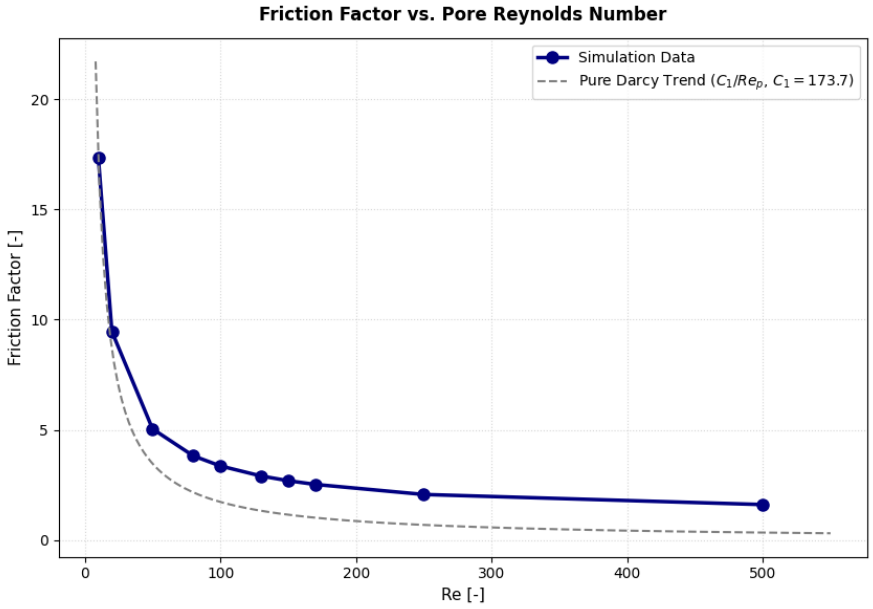


Figure 5.16: Friction factor evolution at increasing Re

As shown in Fig.5.16 shows the change in regime, as the viscous forces are reduced and inertial take over between Reynolds 20 and 50. As the Reynolds number increases then the friction factor stabilizes as expected. The values correspond to values of other TPMS, with expected changes.

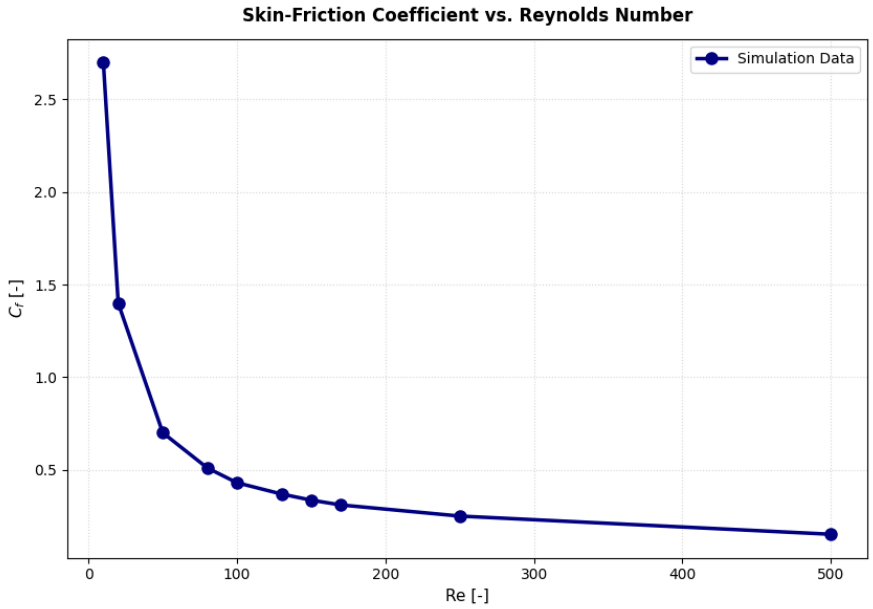


Figure 5.17: C_f evolution at increasing Re

For the skin friction factor the stabilization is less prominent. The value tends do reduce more than what can be observed in the friction factor, as the evolution does not tend to flatten out as much or as rapidly. This trend is quite in line with general f trends that take place in TPMS geometries and can be therefore considered as indicator of the physical validity of this study[13].

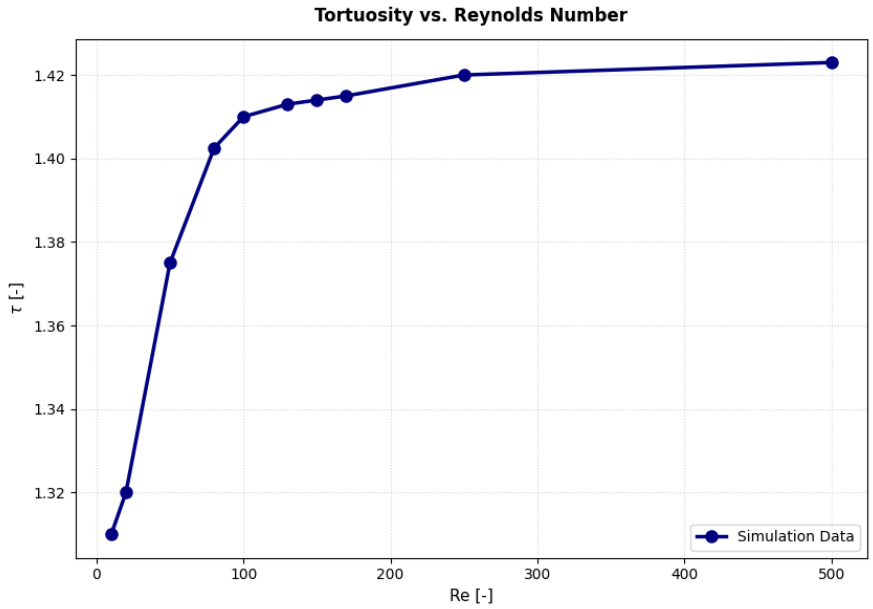


Figure 5.18: τ evolution at increasing Re

The evolution of the tortuosity is rather interesting. As can be observed, the increase at low Reynolds is not as immediate as the decrease seen in C_f and f. In fact, in Darcy regimes, tortuosity is in reality a constant, as discussed in [12]. Due to the highly viscous flow, the flow lines are clearly defined and hardly change. As the flow becomes non Darcy it experiences a sudden increase, or decrease depending on the definition used. In this case, the tortuosity is increasing as can be observed. This shift from costant through Re to an increase produces a curve with increasing slope up until right after Re=20. After this the slope starts reducing non linearly, until it reaches a rather asymptotic value. This reduction in the slope and asymptoticity takes place due to the reaching a physical limit related to the geometry and the main flow of this case. The velocity is imposed, and therefore a particle cannot take, due to force majeure, a path too long or have too large velocities in different directions.

However, despite having expected trends as the Reynolds number increases, none of the variables present noticeable evolution changes to indicate a regime change. In fact, only a convergence to certain values or linear changes can be noticed. This is clearly not indicative of regime changes at specific Reynolds values. Therefore, the evolution of turbulence intensity has been analyzed, as mentioned, with respect to various Reynolds values.

5.2 Flow Transition

5.2.1 Steady to Unsteady Flow Regime Transition

According to Seguin et al.[17] the onset of unsteady behaviors can be observed through the rapid increase in TI. The authors of the paper observed a relatively steady quantity of fluctuations up until a certain point, and the In particular, it can be pinpointed to the intersection of the lines interpolating the steady TI values at low-Re and the sudden surge of fluctuations. Oscillations in a complex geometry such as a Lidinoid do not happen generally, but locally. For this reason a number of probes were used to detect the velocity profiles, as already explained. The turbulence intensity was then averaged in order to have a meaningful value.

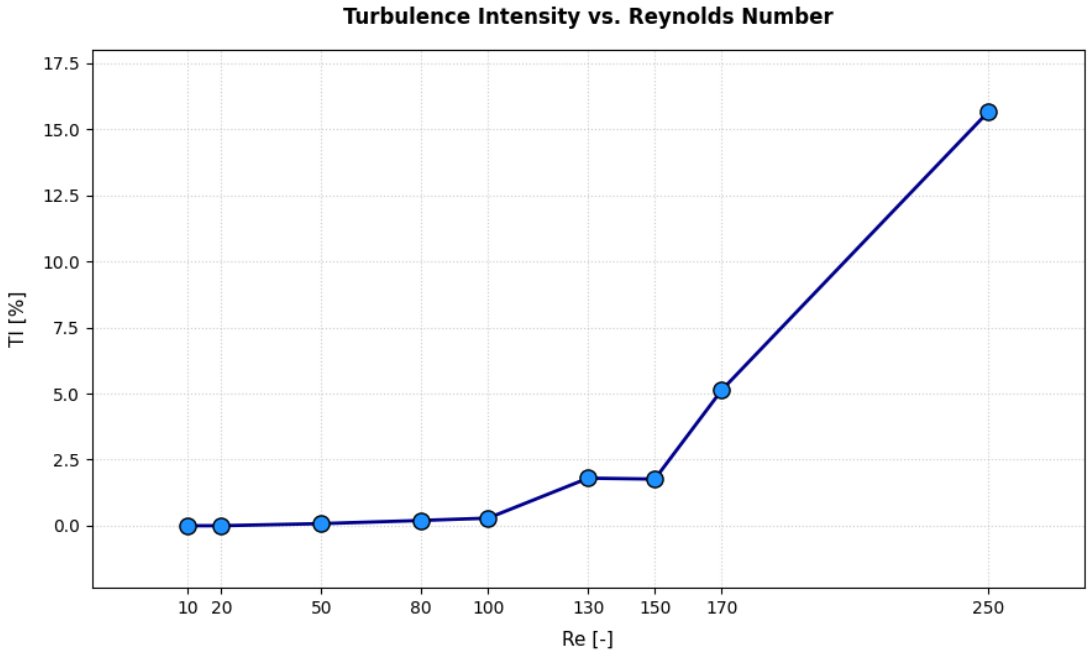


Figure 5.19: TI evolution with respect to the Reynolds number

From these results, it is clear that in the range of Re=150 the transition takes place from laminar steady to unsteady. It is seen that while steady results are obtained at Re equal and lower of 100, this value increases rapidly from 130 onward. Although not zero from Re=20, at Re=100 the values are around 0.2%, which can be deemed negligible.

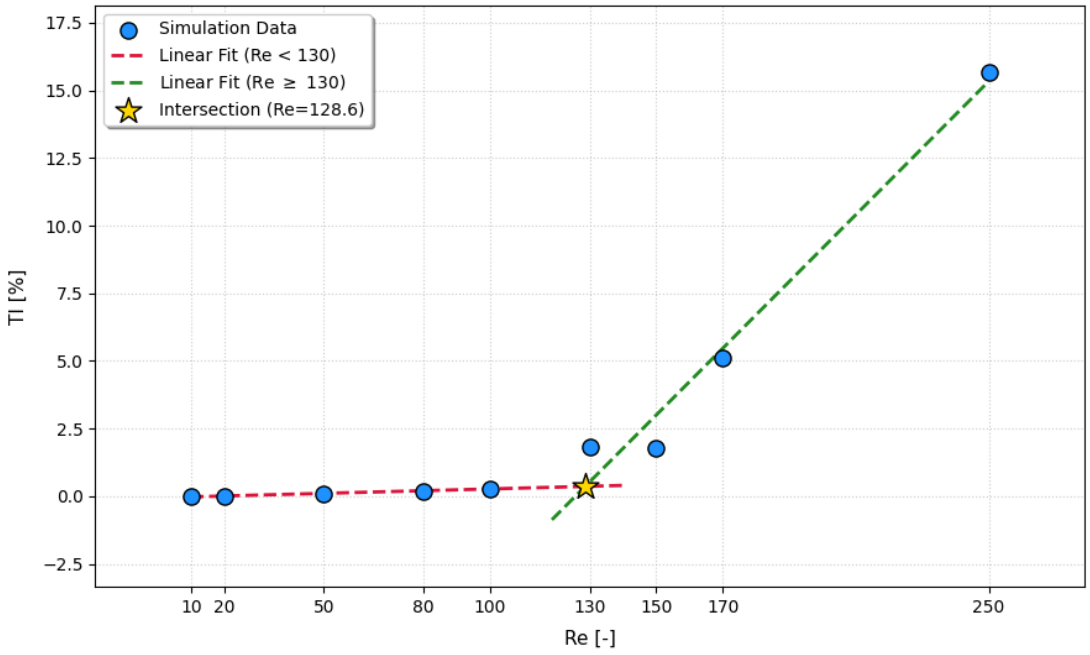


Figure 5.20: Intersection of stable and increasing trends

This intercept value is lower than Re=150, which is generally expected for porous media. In fact, the unsteady Forchheimer regime generally starts at around Re=150. However, this value is close to expected. Additionally, Lidinoid geometries tend to have generally lower pressure drops and more open flow as a result of their configuration. It is possible therefore that instabilities develop earlier than in other porous media as the streamlines are not as constricted.

5.2.2 Unsteady to Turbulent Flow Transition

Comparably important is the detection of the beginning of turbulence in the porous structure. This phenomenon is harder to depict and to validate, particularly due to the local high variations as the Reynolds increases. Additionally, in porous flows turbulence is not reached at once. Some regions should reach turbulence before than other regions, according to [18]. This may be true also for this case, due to increased speeds and fluctuations due to accelerated or decelerated flow in certain local spots. However, the averaging of the various TI values, should give a realistic perspective. In the cited work, in fact, the study was done using electrode sensors, which may be compared to the probes in this numerical case. It is important to add that the work is experimental and therefore is not an infinite medium, but presents outer walls. The detection of the onset of turbulent regime can be performed by

considering the beginning of stabilization of the values of turbulent intensity with respect to the increase in Reynolds.

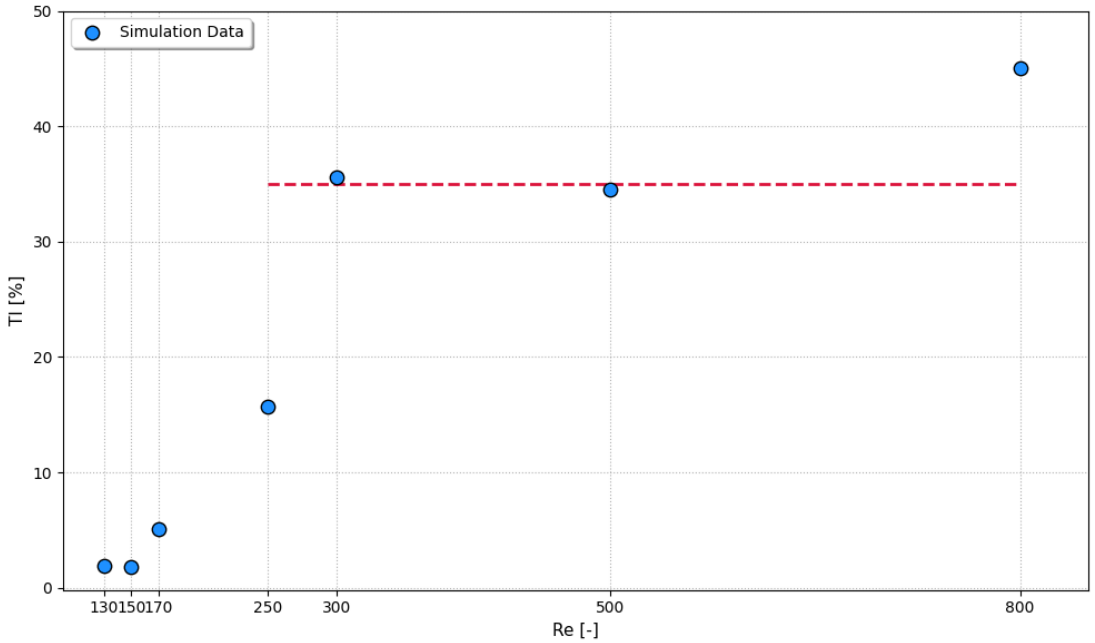


Figure 5.21: Average TI for higher Re values

It must be added that, generally, this stabilization does not take place in reticulated media. However, in the next results, this has been verified, as in the reference paper [18] this phenomenon has only been mentioned but not shown since the studied media were pebble based porosities with pebbles of various diameters. From what is shown in Fig.5.21 it is clear that the stabilization does not happen on the long term. In fact, although it is present from 300 to 500 at least, it does not represent the stabilization seen in [18].

Chapter 6

Conclusions

This thesis investigated the fluid dynamic behavior of a generic flow through a solid Lidinoid TPMS with a porosity of 50% through numerical simulations performed using OpenFOAM.

The results showed that the Lidinoid geometry exhibits the characteristic behavior expected of porous media flows. At low Reynolds numbers, the flow remained stable and dominated by viscous effects, corresponding to the stable Forchheimer regime. In this region, the pressure drop, skin-friction coefficient, and residual histories converged smoothly and exhibited only minor oscillations.

As the Reynolds number increased, increasing flow instabilities were observed. Steady-state simulations revealed progressively larger oscillations in pressure drop and residuals, particularly around Reynolds numbers of 150 and above. These observations suggested the emergence of unsteady flow structures that cannot be accurately represented by a purely steady-state solution. For this reason, transient simulations were adopted as the primary tool for flow-regime characterization.

The analysis methodology based on turbulence intensity, TI, proved effective in identifying the growth of average velocity fluctuations within the porous structure. The results indicate that the transition from a stable Forchheimer regime toward an unsteady laminar regime occurs within the investigated Reynolds-number range. The observed increase in average TI, ΔP fluctuations, and residuals demonstrates that inertial effects progressively destabilize the flow, even in the absence of fully developed turbulence, at around R=130.

Overall, the work proves the numerical framework used in the reference papers of this work, previously cited [17, 18, 20], and applied it to the Lidinoid geometry in the case of an infinite medium in order to analyze the regime changes that take place in TPMS geometries at increasing Re values. The results provide in expanding the understanding of flows in Lidinoid geometries at low Reynolds.

In conclusion, the objectives of the thesis were successfully achieved. The numerical methodology developed proved capable of capturing the onset of flow

unsteadiness in a Lidinoid TPMS, while the obtained results improve the understanding of flow-regime transitions in porous periodic structures.

Bibliography

[1] OpenFOAM Documentation - SIMPLE algorithm — doc.openfoam.com. https://doc.openfoam.com/2306/tools/processing/solvers/pressure-velocity/simple/#. [Accessed 06-05-2026].

[2] Alejandro Alonzo-García, Claudia Gutiérrez-Torres, and José Jiménez-Bernal. *Computational Fluid Dynamics in Turbulent Flow Applications*, pages 1–24. 08 2016.

[3] Hamed Barokh, Seyed Pooya Zoiee, Hamidreza Najafi, Majid Siavashi, and Mohammad Amin Sobati. A detailed comparison of heat transfer and fluid dynamics in voronoi foam and triply periodic minimal surfaces (tpms) via pore-scale investigation. *International Journal of Heat and Mass Transfer*, 245:127007, 2025.

[4] R Farhoud, Saber Amiralaie, Ghader Jabbari, and Solmaz Amiralaie. Numerical study of unsteady laminar flow around a circular cylinder. 01 2012.

[5] Jiawei Feng, Jianzhong Fu, Xinhua Yao, and Yong He. Triply periodic minimal surface (tpms) porous structures: from multi-scale design, precise additive manufacturing to multidisciplinary applications. *International Journal of Extreme Manufacturing*, 4(2):022001, mar 2022.

[6] Jiawei Feng, Jianzhong Fu, Xinhua Yao, and Yong He. Triply periodic minimal surface (tpms) porous structures: from multi-scale design, precise additive manufacturing to multidisciplinary applications. *International Journal of Extreme Manufacturing*, 4(2):022001, 2022.

[7] Paul A Ferney and Nicolas P Martin. Implementation of triply periodic minimal surfaces (tpms) as surface objects in openmc. *Annals of Nuclear Energy*, page 111661, 2025.

[8] E Gajetti, G Boccardo, L Savoldi, and L Marocco. Hydrodynamic characterization of gyroid, diamond and split-p triply periodic minimal surfaces as porous medium. *International Journal of Heat and Mass Transfer*, 252:127439, 2025.

[9] Chengkai Liang, Zhan Su, Zhenyu Ma, and Yushan Gao. Analysis of heat transfer characteristics of lidinoid, neovius and splitp based on numerical simulation. *Journal of Physics: Conference Series*, 2932(1):012028, jan 2025.

[10] Sven Lidin and Stefan Larsson. Bonnet transformation of infinite periodic minimal surfaces with hexagonal symmetry. *J. Chem. Soc., Faraday Trans.*, 86:769–775, 1990.

[11] Daniel Padrão, David Hancock, James Paterson, Frank Schoofs, Chris Tuck, and Ian Maskery. New structure-performance relationships for surface-based lattice heat sinks. *Applied Thermal Engineering*, 236:121572, 2024.

[12] Quanyu Pan, Linsong Cheng, Pin Jia, Renyi Cao, and Peiyu Li. Predicting non-darcy inertial resistance from darcy regime characterization and pore-scale structural descriptors. *Processes*, 14(6):1025, 2026.

[13] Andreas Passos and Andreas Passos. Laminar flow and heat transfer in triply periodic minimal surfaces. 2019.

[14] Cecilia Piatti. *Assessment of thermal-hydraulic performance of circular pipes equipped with Triply Periodic Minimal Surfaces*. PhD thesis, Politecnico di Torino, 2023.

[15] Surendra Singh Rathore, Balkrishna Mehta, Pradeep Kumar, and Mohammad Asfer. Flow characterization in triply periodic minimal surface (tpms)-based porous geometries: Part 1—hydrodynamics. *Transport in Porous Media*, 146(3):669–701, 2023.

[16] E. Robertson, V. Choudhury, S. Bhushan, and D.K. Walters. Validation of openfoam numerical methods and turbulence models for incompressible bluff body flows. *Computers Fluids*, 123:122–145, 2015.

[17] D Seguin, Agnès Montillet, and J Comiti. Experimental characterisation of flow regimes in various porous media—i: Limit of laminar flow regime. *Chemical engineering science*, 53(21):3751–3761, 1998.

[18] D Seguin, Agnès Montillet, J Comiti, and François Huet. Experimental characterization of flow regimes in various porous media—ii: Transition to turbulent regime. *Chemical engineering science*, 53(22):3897–3909, 1998.

[19] Alessio Quamori Tanzi, Antonio Cammi, Luca Marocco, and Laura Savoldi. Design of a 10 MW/m^2-heat-flux removal system for a W7-X divertor tile using triply periodic minimal surfaces. *Case Studies in Thermal Engineering*, 63:105405, 2024.

[20] Kasimhussen Vhora, Dominique Thévenin, Kai Sundmacher, and Gábor Janiga. Investigation of laminar steady and unsteady flows in gyroid tpms structures. In *Conference on Modelling Fluid Flow (CMFF'25): The 19th International Conference on Fluid Flow Technologies*, pages 1–6. University of Technology and Economics, 2025.

[21] Yosef Wakjira, Arturo Cioni, and Hirpa Lemu. Current status of the application of additive-manufactured tpms structure in bone tissue engineering. *Progress in Additive Manufacturing*, 10:1085–1102, 07 2024.

[22] Kaufui V. Wong and Aldo Hernandez. A review of additive manufacturing. *International Scholarly Research Notices*, 2012(1):208760, 2012.

[23] Shixiang Yu, Jinxing Sun, and Jiaming Bai. Investigation of functionally graded tpms structures fabricated by additive manufacturing. *Materials Design*, 182:108021, 2019.