



The logo for Inria (Institut national de recherche en sciences et technologies du numérique) is written in a red, flowing cursive script font.

## Research Internship (PRe)

Field of Study: Mechanical Engineering  
Scholar Year : 2022-2023

# Stochastic Optimization of X-ray Micro-CT Domain Size Using PuMA

### Confidentiality Notice

Non-confidential and publishable report

Author: Raphael ALVES HAILER

Promotion: 2024

ENSTA Paris Tutor: Marica PELANTI

Host Organization Tutor: Pietro MARCO CONGEDO

Internship From 11/05/2023 to 18/08/2023

Host Organization Name: Institut national de recherche en sciences et technologies du numérique (Inria) Saclay Île-de-France  
Address: 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau



## Confidentiality Notice

This present document is not-confidential. It can be communicated outside in paper format or distributed in electronic format.

---

## Acknowledgements

First and foremost, I would like to praise and thank God, who has granted me the opportunity to work alongside wonderful people and gave me all the strength and energy needed to complete this research internship.

I want to express my love and fondness for my wife Gabriely, who gave me support and strength since the beginning of my internship, always motivating me to work hard and give my best. Thank you for believing in my potential, I love you.

Moreover, I am extremely grateful to have been tutored by Professor Pietro Marco Congedo. Thank you for all the support and patience along this period. Many problems emerged, but with your help and guidance we managed to deal with everything and complete the research.

I would like to express my gratitude to the Italian Space Agency member Alessandro Turchi, for the collaboration with the present work. Thanks to you the research was able to be well aligned and coherent with the operation of real X-ray microtomography machines.

I want to thank the developers of PuMA for creating such a great tool for analysing porous materials for space applications, and also for making tutorials available online for learning how to use the software efficiently.

Throughout this research, simulations were conducted using the Plateforme Fédérative pour la Recherche en Informatique et Mathématiques (PlaFRIM) cluster. Thus, I want to express my gratitude to every organization that maintains this cluster running and to the support team, with whom I was in contact during difficult periods.

Furthermore, I want to thank the Institut national de recherche en sciences et technologies du numérique (Inria) Saclay Île-de-France and the École Nationale Supérieure de Techniques Avancées (ENSTA) Paris, for making this experience possible. This has been my first research experience and I have loved it. In addition, I want to thank the Platon team for welcoming me at Inria.

This study was partly financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. I must express my thankfulness to CAPES for the BRAFITEC program, which has granted my financial support to maintain myself in France. Without this support my dreams would have been left in the drawer.

Last but not least, I want to express my love and gratitude to my family and friends, for motivating me and pushing me forward.

## Abstract

In this present work we are interested in characterizing a porous material mechanical properties by using the PuMA software, developed by NASA. An uncertainty quantification analysis will be conducted in order to provide more reliable results for future experimentalists. More precisely, the main objective of the uncertainty quantification analysis is to formally define the problem under an optimization under uncertainty framework and use the correct mathematical tools to determine the necessary size of the sample's physical domain captured in the X-ray microtomography, in order to capture the macroscopic features of the material using a microscopic domain. The domain sought is called the representative elementary volume, and determining it is crucial for obtaining trustful results from experimental data, while reducing the need for a high number of samples to validate the experimental campaign, thus saving money and time. This work is a proof of concept, and its main purpose is to provide a guideline for future research on the topic of porous materials characterization for thermal protection systems used in space applications.

**Keywords**— Uncertainty quantification, Gaussian process, Carbon/Phenolic ablators, Thermal protection systems, Porous materials, Microstructure, Thermal conductivity, Representative elementary volume, PuMA, Porous microstructure analysis, Optimization, Surrogate model, Robustness and reliability.

## Résumé

Dans ce travail, nous nous intéressons à la caractérisation des propriétés mécaniques d'un matériau poreux en utilisant le logiciel PuMA, développé par NASA. Une analyse de quantification de l'incertitude sera effectuée afin de fournir des résultats plus fiables aux futurs expérimentateurs. Plus précisément, l'objectif principal de l'analyse de quantification de l'incertitude est de définir formellement le problème dans un cadre d'optimisation avec incertitude et d'utiliser les outils mathématiques appropriés pour déterminer la taille nécessaire du domaine physique capturé par la micrographie à rayons X, afin de capturer les caractéristiques macroscopiques du matériau à l'aide d'un domaine microscopique. Le domaine recherché est appelé volume élémentaire représentatif, et sa détermination est cruciale pour obtenir des résultats fiables à partir de données expérimentales, tout en réduisant la nécessité d'un grand nombre d'échantillons pour valider la campagne expérimentale, ce qui permet d'économiser de l'argent et du temps. Ce travail est une preuve de concept, et son objectif principal est de fournir une ligne directrice pour les recherches futures sur le thème de la caractérisation des matériaux poreux pour les systèmes de protection thermique utilisés dans les applications spatiales.

**Mots clés**— Quantification des incertitudes, Processus Gaussien, Ablateurs de carbone/phénoliques, Systèmes de protection thermique, Matériaux poreux, Microstructure, Conductivité thermique, Volume élémentaire représentatif, PuMA, Analyse de la microstructure poreuse, Optimisation, Modèle de substitution, Robustesse et fiabilité.

# Contents

<b>1</b>	<b>Introduction</b>	<b>14</b>
<b>2</b>	<b>Experimental X-ray microtomography for porous materials</b>	<b>17</b>
2.1	Characterization of porous materials properties . . . . .	17
2.1.1	Porosity . . . . .	17
2.1.2	Surface area and specific surface area . . . . .	17
2.1.3	Thermal and electrical conductivities . . . . .	18
2.2	X-ray microtomography . . . . .	20
<b>3</b>	<b>The Porous Microstructure Analysis (PuMA) software</b>	<b>22</b>
3.1	Description of the software . . . . .	22
3.2	Material properties in PuMA . . . . .	22
3.2.1	Porosity in PuMA . . . . .	22
3.2.2	Surface area and specific surface area in PuMA . . . . .	23
3.2.3	Thermal and electrical conductivities in PuMA . . . . .	23
3.2.4	Further possibilities with PuMA . . . . .	24
<b>4</b>	<b>Mathematical Formulation of the problem</b>	<b>26</b>
4.1	PuMA workspace variables and definitions . . . . .	26
4.2	Limitations of real X-ray microtomography scans imposed on PuMA . . . . .	27
4.3	Mathematical point of view . . . . .	27
<b>5</b>	<b>Introduction to constrained multi-objective optimization under uncertainty</b>	<b>30</b>
5.1	Deterministic constrained multi-objective optimization . . . . .	30
5.2	Optimization under uncertainty . . . . .	32
5.3	Probabilistic framework . . . . .	33
5.4	Coupling the optimization framework with PuMA . . . . .	35
<b>6</b>	<b>Brute force search for the REV</b>	<b>36</b>
6.1	Computing the reference values . . . . .	36
6.1.1	Input parameters . . . . .	36
6.1.2	Results obtained . . . . .	37
6.2	Considering the X-ray microtomography limitations . . . . .	41
6.2.1	Input parameters . . . . .	41
6.2.2	Results obtained . . . . .	42
6.3	Conclusion of the brute force search . . . . .	47
<b>7</b>	<b>Analysis conducted with the SAMATA algorithm</b>	<b>48</b>
7.1	Understanding the algorithm . . . . .	48
7.1.1	The Surrogate-Assisting strategy . . . . .	49
7.1.2	The Measure Approximation with Tunable Accuracy (MATA) technique .	49
7.1.3	SA and MATA pseudo-code algorithms . . . . .	50
7.2	Adapting the algorithm . . . . .	51
7.2.1	Random field surrogate based on a Kernel Density Estimation . . . . .	51
7.2.2	Constructing the SA model . . . . .	52
7.2.3	Adapting the SAMATA algorithm to a non-parametric uncertainties scenario	54
7.3	Coupling SAMATA with PuMA . . . . .	54
<b>8</b>	<b>Conclusion</b>	<b>59</b>
<b>9</b>	<b>References</b>	<b>60</b>

## List of Figures

1	Galileo Probe Heatshield Ablation. Figure from NASA, obtained from [6]. . . . .	14
2	Illustration of all surface combinations obtained from the marching cube algorithm. The black circles indicates that the vertices are inside the surface of the rendering. Credits to [13]. . . . .	18
3	Diagrammatic representation of the X-ray tomography scheme. Credits to [14]. . .	20
4	REV analysis conducted on a 2000 voxel <sup>3</sup> sample with isotropic random fibers, artificially generated using PuMA v2.1. Credits to [9]. . . . .	21
5	Examples of different microstructures generated with the use of PuMA v2.1: (a) Random transversely isotropic fiber preform, (b) Packed sphere bed with varying sphere diameters, and (c) Period fom structure with uniform void diameter. Credits to [9]. . . . .	25
6	Comparison between different fibers obtained by changing the value of $r_{vx}$ . Credits to [11]. . . . .	27
7	Surface renderings of a porous material X-ray micro-CT, using three different resolutions and domains, using PuMA v2.1. Credits to [9]. . . . .	28
8	Simple example to illustrate the search for the Pareto-optimal designs. . . . .	32
9	Thermal response simulation results for determining the reference values for thermal conductivities in all directions, using 1000 points per configuration ( $r_{vx}, D_{phy}$ ). . . . .	39
10	Objective functions results obtained in all directions, using 1000 points for each configuration pair ( $r_{vx}, d_{vx}$ ). . . . .	43
11	Objective functions results of thermal conductivities in all directions, using 1000 points for each configuration pair ( $r_{vx}, d_{vx}$ ). . . . .	45
12	The SAMATA algorithm flowchart. Credits to [28]. . . . .	48
13	Comparison between $\text{POP}_{min}$ results obtained for various values of thresholds $s_1$ and $s_2$ , considering $(D_{phy}, \ell_{vx}^{phy}) \in [100, 350] \times [0.75, 2.5]$ . . . . .	57

## List of Tables

1	Correlation coefficients Corr( $\log_{10}(\text{CV}(k_{ii}), D_{phy})$ for all values of $r_{vx}$ , being i the subscript to indicate the direction, being them $x, y$ and $z$ . . . . .	40
2	Pareto fronts found for each direction analysis and separated between the $d_{vx}$ values used. In order to fit the table in the page of this work, only the $r_{vx}$ values are indicated for identifying each Pareto front. . . . .	46

## List of Symbols

### Digital domain properties

$\ell_{vx}^{phy}$	Voxel length
$l_{x,vx}$	Number of voxels of the domain in the $x$ direction
$l_{y,vx}$	Number of voxels of the domain in the $y$ direction
$l_{z,vx}$	Number of voxels of the domain in the $z$ direction
$N_{void}$	Number of voxels in the void phase
$N_{solid}$	Number of voxels in the solid phase
$R_{phy}$	Average radius of the material's physical fiber
$r_{vx}$	Number of voxels used to represent the average radius of the fibers
$L_{phy}$	Average length of the material's physical fiber
$l_{vx}$	Number of voxels used to represent the average length of the fibers
$D_{phy}$	Physical length of the cubic domain side
$d_{vx}$	Number of voxels used to represent the physical length of the cubic domain side
$\sigma_{R_{phy}}$	Standard deviation associated to $R_{phy}$
$\sigma_{L_{phy}}$	Standard deviation associated to $L_{phy}$
$\sigma_{r_{vx}}$	Standard deviation associated to $r_{vx}$
$\sigma_{l_{vx}}$	Standard deviation associated to $l_{vx}$
$\Delta\theta$	Maximum fiber angle variation
$\ell_{min}$	Minimum voxel length in real X-ray micro-CT machines
$r_{max}$	Maximum number of voxels to represent the fiber's radius in real X-ray CT machines

### Geometrical properties

$V$	Physical domain volume
$V_{void}$	Void phase volume
$V_{solid}$	Solid phase volume
$S$	Total surface area
$s$	Specific surface area
$L_x$	Physical length of the domain in the $x$ direction
$L_y$	Physical length of the domain in the $y$ direction
$L_z$	Physical length of the domain in the $z$ direction

### Non-dimensional numbers

## LIST OF SYMBOLS

---

$K_n$  Knudsen number

### Material properties

$\varepsilon$  Porosity

$d$  Density

$c_p$  Specific heat at constant pressure

### Number sets

$\mathbb{R}$  Real numbers

$\mathbb{N}^*$  Natural numbers excluding zero

### Optimization framework symbols

$\xi$  Uncertain parameters

$\Xi$  Set containing the uncertain parameters

$\text{Var}$  Variance

$\mu$  Average

$f$  Objective function

$f^{rel}$  Relative objective function

$g$  Constraint function

$\mathbf{x}$  Design vector

$\chi$  Design space

$X$  set containing all evaluated vectors

$\mathbb{A}$  Admissible set

$\mathbb{F}$  Failure set

$\mathbb{P}$  Pareto front

$\rho_f$  Robustness measure

$\rho_g$  Reliability measure

$\rho$  Evaluated vector using statistical measures

$q$  Quantities of interest

$\mathbf{P}$  Random field

$P_\Omega$  Probability computed over the joint distribution of  $\Omega$

$\phi_\Omega$  Joint probability density function of  $\Omega$

$\mathbb{1}_\Omega$  Indicator function of set  $\Omega$

$\mathbf{P}(\cdot)$  Random vector that resides in the random field  $\mathbf{P}$

$\tilde{\rho}$	Estimation of the random evaluated vector
$s$	User-defined threshold for the SAMATA algorithm
$\chi_{new}$	Set containing new designs to be explored by the optimization algorithm
$\chi_c$	Set containing designs to compute in the SAMATA algorithm
$\chi_{new}$	Set containing the designs to be refined by the SAMATA algorithm
$\Delta$	Measure of variability for the SAMATA algorithm
$\mathbb{T}$	Training set
$N_{samples}$	Number of samples
$N_{MC}$	Number of samples using the Monte Carlo approximation
$Id$	Identity matrix
$\Sigma$	Covariance matrix
$\mathbf{N}(\mu, \Sigma)$	Normal distribution with mean $\mu$ and covariance matrix $\Sigma$
$\mu_{SA}$	Predictive mean of the Gaussian process
$\mathbf{h}$	Kernel bandwidth
$ker_{\mathbf{h}}(\cdot, \cdot)$	Kernel function with kernel bandwidth $\mathbf{h}$
$\bar{\rho}$	Normalized data
$\alpha$	Probability of acceptance for the Metropolis-Hastings algorithm
$\mathbf{Q}$	Evaluated quantity of interest
$\mathcal{Q}$	Set containing all evaluated quantities of interest as samples
$E$	Expectation
$\mathcal{U}$	Uniform distribution

### Other symbols

$\vec{q}_{thermal}$	Heat flux vector
$\vec{x}$	Position vector
$t$	Time
$T$	Temperature field
$k$	Thermal conductivity
$\mathbf{k}$	Thermal conductivity tensor
$\dot{q}_{thermal}$	Volumetric heat generation
$\sigma_{el}$	Electrical conductivity
$\Sigma_{el}$	Electrical conductivity tensor

---

## LIST OF SYMBOLS

---

$\Phi$	Electrical potential field
$\theta$	Angle
$\bar{\lambda}$	Medium's mean free path
$l_D$	Medium's characteristic length
$\bar{v}$	Mean thermal velocity

## List of Acronyms

**NASA** — National Aeronautics and Space Administration

**TPS** — Thermal Protection System

**micro-CT** — micro-Computed Tomography

**REV** — Representative Elementary Volume

**TIFF** — Tagged Image File Format

**PuMA** — Porous Microstructure Analysis

**UQ** — Uncertainty Quantification

**OUU** — Optimization Under Uncertainty

**3D** — Three-Dimensional

**2D** — Two-Dimensional

**FFTs** — Fast Fourier Transforms

**BiCGSTAB** — Bi-Conjugate Gradient Stabilized Method

**TPMS** — Triply Periodic Minimal Surfaces

**PlaFRIM** — Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

**DoE** — Design of Experiment

**RDO** — Robust Design Optimization

**RBDO** — Reliability-Based Design Optimization

**QoI** — Quantities of Interest

**PDF** — Probability Density Function

**POP** — Pareto-Optimal Probability

**CV** — Coefficient of Variation

**MATA** — Measure Approximation with Tunable Accuracy

**SA** — Surrogate-Assisting

**GP** — Gaussian Process

**KDE** — Kernel Density Estimation

**MISE** — Mean Integrated Squared Error

**MH** — Metropolis-Hastings

**i.i.d** — Independent and Identically Distributed

# 1 Introduction

Analysing porous materials has been quite a challenge for the scientific community for many decades. But the advantages of using porosity in our favor are matchless, as it has been exhaustively used in sound absorption and heat insulation applications, ranging from sound engineering to the aerospace engineering fields. One of the most famous applications of porous materials in heat insulation is the Thermal Protection System (TPS) found in most of the space projects.

Since the beginning of the space exploration era, humans have been searching for ways to ensure the success of missions in a safe and efficient manner. The reutilization of rockets and command capsules by space agencies and by the private space industry has been one of the most critical points of research in this field, as it is mandatory in order to recover the astronauts and it drastically reduces the costs of a space mission. Furthermore, being able to safely enter in an atmosphere of another planet has proven to be essential for the exploration of the solar system, with missions such as the Mars Science Laboratory mission from the National Aeronautics and Space Administration (NASA), with its Curiosity rover [1] and the Perseverance rover [2] being major examples.

Considering an atmospheric reentry, it is well known that the temperatures reach extremely high values. Considering the Space Shuttle from NASA as an example, it can reach temperatures up to 1477 °C in reentry [3]. Due to the atmospheric drag, the object entering the atmosphere experiences very high mechanical stress and aerodynamic heating, which is caused by skin drag and also by the compression of the air in front of the object, as it is traveling in speeds between 7 and 8 km/s. Such conditions are responsible for many missions to fail, due to the complex heat exchange processes during atmospheric entry, making the TPS of space crafts to be of the utmost importance. One unfortunate example of a disaster related to failure in the TPS is the Space Shuttle Columbia mission [4], which led to the death of all seven astronauts on board.

Considering the types of Thermal protection systems, there are re-usable TPS and ablative TPS, being the latter the most common used in the industry. Their fundamental characteristic is that they provide thermal protection to the spacecraft by losing mass, being thus decomposed when exposed to extreme temperatures and stress. This kind of thermal protection is produced through physico-chemical transformations and erosion, in order to dissipate the heat generated in atmospheric entry [5]. One example of a heat shield made out of ablative material that almost failed is that of the Galileo Probe, which was designed to study Jupiter and its moons. The consumption of the material after entry can be seen in Figure 1.

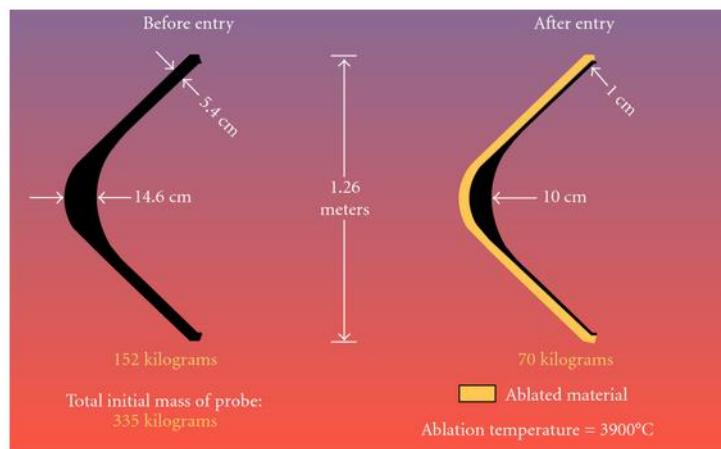


Figure 1: Galileo Probe Heatshield Ablation. Figure from NASA, obtained from [6].

Regarding the materials that can be used to develop a reliable ablative TPS, one must address porous materials. Porous materials contains a solid phase and a void phase, which is

filled with the matter in the medium that the material is inserted. For example, in space the medium would be the vacuum.

These materials have very light weight and high heat resistance, making them the perfect candidates for aerospace applications. There are three main types of porous materials, being them the cellular, the fibrous and the granular types [7]. Porous materials that are made out of fibers have specific properties to be addressed in this work.

First of all, not all fibers are oriented in the same manner, meaning that in general each sample taken from a porous material will be structured and behave differently. Therefore, one could say that there is a need for a statistical analysis to be conducted in porous materials, in order to help experimentalists on better characterizing them.

When it comes to analysing porous materials experimentally, scientists rely heavily on X-ray microtomography. More information on X-ray microtomography, or X-ray *micro-Computed Tomography* (micro-CT), can be read on [8]. Even though X-ray micro-CT has many applications in the medical field, it is also a very effective non-destructive technique for visualizing structures within solid objects, in order to obtain information on their three-dimensional geometry. However, preparing an X-ray micro-CT experimental campaign can be very expensive, specially due to the fact that it is very hard to know how many samples are needed in order to compute material properties in a way that it does not vary as much and that averages out to the correct result, in order to use it to describe the macroscopic behaviour. Furthermore, most of the time scientists need to take samples at various scales as well, in order to better comprehend the behaviour of the material.

The main problem for scientists studying porous materials with X-ray micro-CT is that there is a great lack of knowledge on what should be the correct sample size to faithfully capture the macroscopic behaviour of the material. The domain size which we are referring to is called the *Representative Elementary Volume* (REV).

In brief, the REV is the smallest physical domain where a measurement will be representative of the whole, meaning that we can extract bulk properties by taking measurements in the REV scale. Hence, it is of the utmost importance for scientists to correctly determine the REV in order to avoid experimental errors or misleading conclusions.

Hence, numerical tools are of great help to estimate the correct sample size and to minimize the discrepancy between the physical properties obtained for the samples. Many tools are available to help the scientists in the analysis of porous materials, by running numerical simulations on three-dimensional *Tagged Image File Format* (TIFF) files obtained from the X-ray micro-CT, but there is still the need to obtain this data experimentally.

For this study, we are going to use the *Porous Microstructure Analysis* (PuMA) software [9] to compute effective material properties and perform numerical response simulations on porous microstructures. PuMa can import experimental TIFF files from X-ray micro-CT and it can additionally generate the porous microstructures artificially, by taking into account a number of input parameters. By using the generated microstructures, we are able to obtain data from porous materials in the most variate configurations, meaning that we do not need to rely solely on experimental data obtained from real X-ray micro-CT scans.

It is necessary to supplement experimental analysis with numerical models for two main reasons. The first one is to investigate and reconstruct important properties, which cannot be directly observed, and the second one is to merge information coming from different measurements to learn about the underlying physical model driving the material properties. In addition, due to the stochasticity inherent of porous materials sampling, it is necessary to find a mathematical model in which we can determine the REV with good accuracy. PuMa makes it easier by creating an environment where we can run multiple simulations in parallel, in order to save time, thus being potentially one of the greatest tools available to better orient an experimental microtomography campaign.

The main objective of this work is to discuss the necessary resolution of the fibers and

the domain in PuMA, in order to identify the reference elementary volume in a robust way. Our focus is the influence of the domain resolution and other numerical parameters to attain converged prediction of the quantities of interest. In particular, we discuss the choice of the fiber's radius in PuMA, in unities of voxels (analogous to pixels, but in a three-dimensional space) for representing the fibers in the generated domain. Finally, we discuss a procedure to assess the appropriate resolution to measure average properties of the material depending on the microtomography machine limitations.

This study is a proof of concept for further research on the topic, in order to apply the numerical procedure proposed here to prepare an experimental campaign of microtomography measurements. Thus, the focus is to provide a guideline for future research to be done.

Considering that each and every sample coming from a fibrous porous material responds differently to a numerical simulation, then it is necessary to tackle this problem with a statistical framework, in order to reduce uncertainties. Hence, we will study the REV under an *Uncertainty Quantification* (UQ) scope and by considering the mathematical tools available for studying *Optimization Under Uncertainty* (OUU) problems, as the search for the REV is a bi-objective optimization under uncertainty problem, and it will be handled as such.

Many works on UQ methodology have been conducted before on TPS applications based on numerical models, such as seen in [10]. However, other than [11], there is no record of the use of UQ methodology alongside with PuMA.

Section 2 of this work will be dedicated to describing the most crucial properties of porous materials for this work and how X-ray microtomography works, together with the trade-offs imposed by the machinery. Afterwards, in Section 3 we are going to take a better look at PuMa and better describe its functioning, together with a description of the outputs given by the numerical simulations available for the user to run, specially those that interests us the most. Later, in Sections 4 and 5 we discuss the mathematical formulation of the problem studied, together with an introduction to the optimization framework, where we discuss the important points for analysing a multi-objective optimization under uncertainty problem. From there on, we are going to explore the software by running simulations massively, in order to obtain an initial estimate of the REV by brute force in Section 6, which is the way most people would approach this problem initially. Finally, an optimization under uncertainty analysis will be conducted using the SAMATA algorithm, presented in Section 7, in order to optimize the search for the REV considering a probabilistic framework, saving time and reducing the need for massive computational power.

## 2 Experimental X-ray microtomography for porous materials

### 2.1 Characterization of porous materials properties

Regarding the characterization of porous materials, it is crucial to address their unique properties, being them geometrical, thermal or transport properties. In the following, we are going to explore the most important properties of porous materials for this study, which can be computed numerically through the use of a given material microstructure that was obtained experimentally or not. Although there are more important properties related to porous materials, they are not playing a role in the context of this work.

#### 2.1.1 Porosity

Any porous material is constituted by a solid phase and a void phase, where the void phase is filled with the matter that is in the medium surrounding the material. The porosity  $\varepsilon$  is equal to the ratio between the volume occupied by the void and the volume occupied by the whole physical domain. Being  $V$ ,  $V_{\text{void}}$  and  $V_{\text{solid}}$  the total volume of the physical domain, the volume occupied by the void phase and the volume occupied by the solid phase, respectively, then we have

$$\varepsilon = \frac{V_{\text{void}}}{V} \quad (1)$$

Note that, from this definition,  $0 \leq \varepsilon \leq 1$ . The porosity is a very important property for any TPS, as the void phase is responsible for diminishing the thermal conductivity of these materials. Therefore, materials with high porosity are recommended for thermal applications.

#### 2.1.2 Surface area and specific surface area

The total surface area  $S$  of a porous material is defined as the sum of the surface area of all fibers, grains or cells inside the domain. Meanwhile, the specific surface area  $s$  is defined as the ratio between the total surface area and the total volume of the physical domain  $V$ . Therefore, we have:

$$s = \frac{S}{V} \quad (2)$$

When it comes to analysing the X-ray micro-CT scans or generating artificial domains digitally, we must make use of voxels (equivalent to pixel representation but in three-dimensional space) in order to represent the material in three dimensions. There are many difficulties in correctly rendering the surface of the material, as we must be able to separate the voxels that are part of the material from those that are not. There are many algorithms developed for this purpose, but the one that we are going to rely for this study is the Marching Cubes algorithm [12].

Essentially, the Marching Cubes algorithm, developed by Lorensen and Cline, works by dividing the input volume into a discrete number of cubes and generating triangular models of constant density surfaces [5]. The algorithm works by moving a cube (thus, "marching") over the whole domain that was divided into cubes and detecting whether the vertices of the cube are inside or outside the material, by using an implicit function such as  $f(x, y, z) = 0$  as criterion.

In theory, we could have  $2^8$  possible configurations, however all the configurations belong to a family of 15 distinct cases, which are displayed in Figure 2. The final mesh generated by the algorithm is the union of all triangles obtained during the iterations. One can notice that as the cubes in the discretization gets smaller, the better will be the final surface obtained,

therefore the total number of voxels available at a fixed domain is a very important parameter for determining the resolution of the three-dimensional rendering.

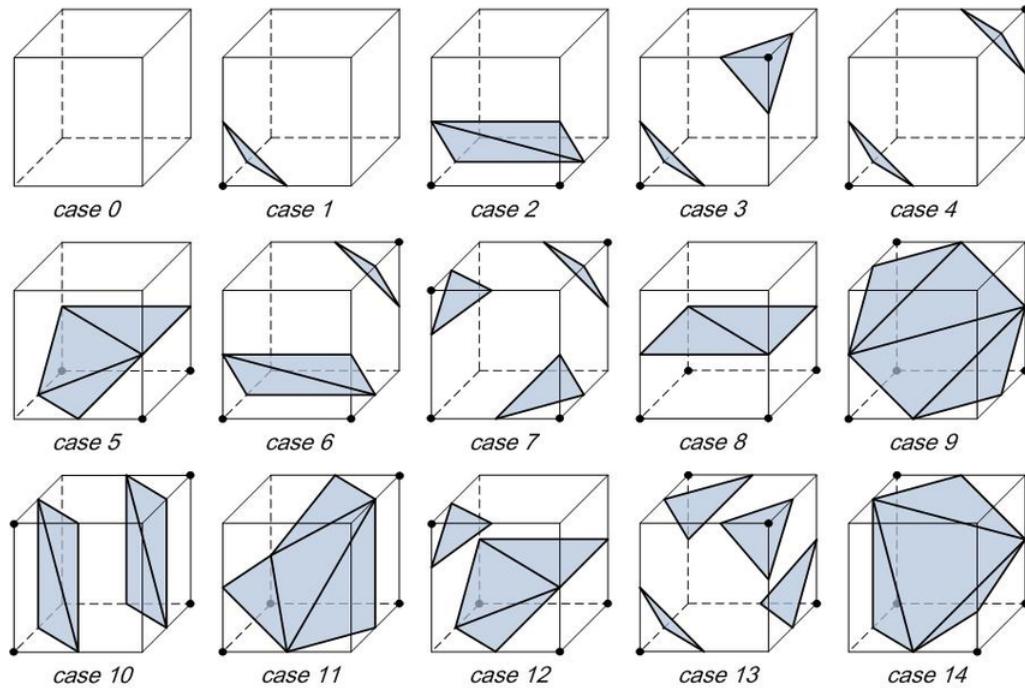


Figure 2: Illustration of all surface combinations obtained from the marching cube algorithm. The black circles indicates that the vertices are inside the surface of the rendering. Credits to [13].

After comprehending the concept behind the rendering algorithm for the material's surface, it's time to express the total surface area  $S$  in mathematical terms. We are going to call  $\ell_{\text{vx}}^{\text{phy}}$  the *Voxel Length*, in units of length per voxel. This number is used to convert between units of voxels and units of length. We can say therefore that the total surface area is equal to the sum of all the triangle areas in the computed triangulation of the iso-surface, using the Marching Cubes algorithm. Thus:

$$S = \frac{1}{2} \ell_{\text{vx}}^{\text{phy}} \sum_i |\vec{u}_i \times \vec{w}_i| \quad (3)$$

Where  $\vec{u}_i$  and  $\vec{w}_i$  are the vectors that defines the  $i^{\text{th}}$  triangle, giving the direction of two of the three sides of each triangle. Furthermore, considering the digitized domain, we can say that

$$V = L_x L_y L_z = l_{x,vx} l_{y,vx} l_{z,vx} (\ell_{\text{vx}}^{\text{phy}})^3 \quad (4)$$

Where  $L_x$ ,  $L_y$  and  $L_z$  are the physical lengths of the domain in the  $x$ ,  $y$  and  $z$  direction, respectively, while  $l_{x,vx}$ ,  $l_{y,vx}$  and  $l_{z,vx}$  are the total number of voxels for each length described. We have that the surface area and the specific surface area are very important geometrical properties of any porous material due to its relevance in chemical reaction rate.

### 2.1.3 Thermal and electrical conductivities

Firstly, let us talk about thermal conductivity. Thermal conductivity measures the ability of given material or medium to conduct energy through random molecular interactions when there is a difference in temperature between points in space. This energy flows due to a difference in

temperature that causes molecules to be more excited in one region than in another, and the molecular collisions between these molecules diffuse the energy. We call this phenomenon heat transfer by conductivity.

We can quantify the flow of heat by using a vector  $\vec{q}_{thermal}(\vec{x}, t)$ , which has units of W/m<sup>2</sup> in the international system of units (SI), where  $\vec{x}$  denotes the position vector in space and  $t$  denotes time. For instance, if we consider a domain in the Cartesian plane then  $\vec{x} = (x, y, z)$ .

Due to the second law of thermodynamics, we have that heat flows from high temperatures to low temperatures. One can simply imagine that this happens because lower temperature molecules have less kinetic energy, meaning that higher temperature molecules will collide and transfer energy to lower temperature molecules over time.

Due to this fact, we can imagine that in fact  $\vec{q}_{thermal}(\vec{x}, t) \propto -\nabla T(\vec{x}, t)$ , where  $T(\vec{x}, t)$  denotes the temperature field. In fact, this is what Fourier's law postulates, as the thermal flux and the temperature gradient are related to each other by a single constant  $k$ , which we call the *thermal conductivity*. Therefore, Fourier's law states that

$$\vec{q}_{thermal}(\vec{x}, t) = -k \nabla T(\vec{x}, t) \quad (5)$$

We have that  $k$  is the constant of proportionality and  $k > 0$ . Furthermore,  $k$  depends on  $T(\vec{x}, t)$ , therefore it depends implicitly on space and time. Considering porous materials as our reference frame for studying the thermal conductivity, then we must address the use of the thermal conductivity tensor  $\mathbf{k}$ , due to the fact that porous materials exhibits anisotropic behaviour. Thus, Equation 5 simply becomes

$$\vec{q}_{thermal}(\vec{x}, t) = -\mathbf{k} \cdot \nabla T(\vec{x}, t) \quad (6)$$

Where now we have a dot product between  $\mathbf{k}$  and the the temperature field gradient. We have that the thermal conductivity tensor can be written in matrix notation as follows:

$$\mathbf{k} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix} \quad (7)$$

Usually porous materials consisting of fibers are said to be transversely isotropic, which means that their physical properties are symmetric about an axis [5], which we are going to denote as the  $z$  axis for simplicity reasons. Thus we find that  $k_{xx} \approx k_{yy} \neq k_{zz}$ . Again, we are using Cartesian coordinates because we deal mostly with cubic domains in this sort of material analysis.

As result of this physical behaviour of transversely isotropic materials, the  $x$  and  $y$  directions are said to be the *In-Plane Directions*, while the  $z$  direction is called to be the *Through-Plane Direction*.

Considering the material response to heat conduction simulations, we must address the use of the heat equation. The general equation is written as follows:

$$c_p d \frac{\partial T(\vec{x}, t)}{\partial t} = \dot{q}_{thermal} + \nabla \cdot (\mathbf{k} \cdot \nabla T(\vec{x}, t)) \quad (8)$$

Where  $c_p$ ,  $d$  and  $\dot{q}_{thermal}$  stands for specific heat at constant pressure, density and volumetric heat generation, respectively. Note that  $\dot{q}_{thermal}$  has units of W/m<sup>3</sup> in the SI. In this study we are interested only in the steady case with no heat generation, so we must set  $\frac{\partial T(\vec{x}, t)}{\partial t} = \dot{q}_{thermal} = 0$ .

Concerning the electrical conductivity, the approach is very similar to that of the thermal conductivity. We denote  $\sigma_{el}$  the electrical conductivity and, in the same manner as the thermal conductivity, it can be defined as a tensor for the most general case, to take into account the anisotropic effects. The electrical conductivity tensor will be called  $\Sigma_{el}$ . The general equation expressing the electrical conduction is written as follows:

$$\nabla \cdot (\Sigma_{el} \cdot \Phi) = 0 \quad (9)$$

Where  $\Phi$  denotes the electrical potential field. This work is not interested in computing the electrical conductivity of porous materials, due to the fact that for a TPS what matters the most is the thermal response of the materials. However, it is important to highlight the physical properties of materials and specially for the porous materials, where anisotropy plays a big role.

## 2.2 X-ray microtomography

X-ray microtomography, or X-ray micro-CT, is a very well established non-destructive experimental technique used to obtain three-dimensional (3D) images from physical samples of materials. Considering the spatial resolution of the images, it is possible to observe details of the sample within the micrometer range, allowing us to also see the internal structure of the material. Its working principle is shown in Figure 3.

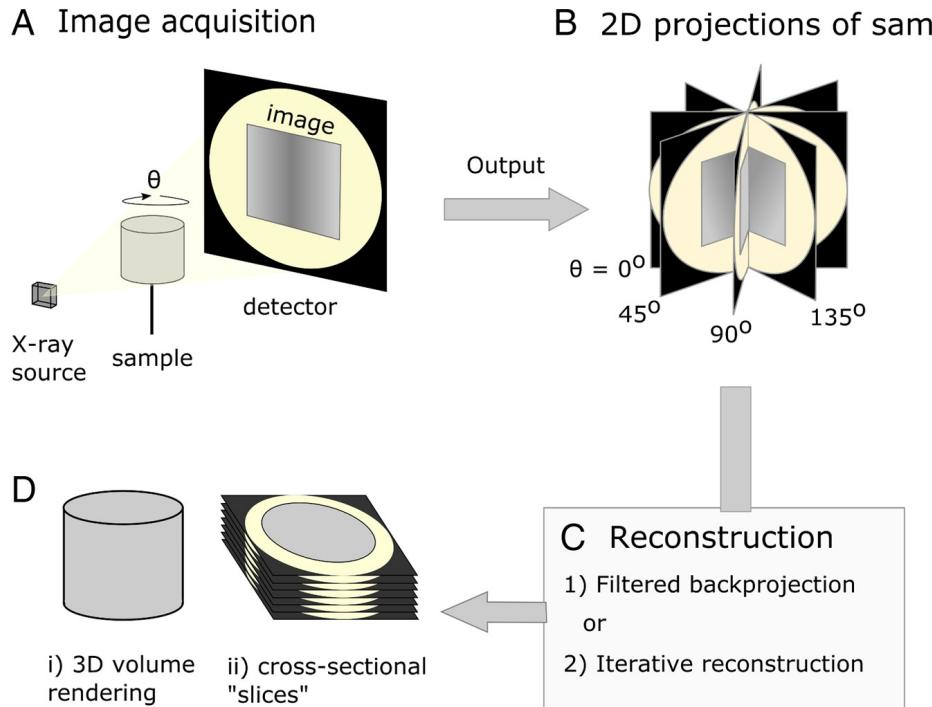


Figure 3: Diagrammatic representation of the X-ray tomography scheme. Credits to [14].

Pulses of X-ray are emitted while the sample is rotating in discrete increments of  $\theta$ , until it turns  $360^\circ$ . For each pulse, it is possible to obtain two-dimensional (2D) images projected on the detector, that are stored in gray-scale. By using these images, the 3D model of the sample can be reconstructed by using numerical algorithms [14]. This technique allows us to visualize also the interior of the fibers of a porous material, making it possible to look for microporosity or even to check if the fibers are hollow. Afterwards, the 3D volume render is stored in a TIFF file as stack cross-sectional layers with lossless compression [5]. The obtained scans can be thoroughly used for calculating effective material properties and performing numerical response simulations on the microstructures.

When it comes to preparing an experimental X-ray micro-CT campaign, we must consider the limitations inherent to the process. For instance, the microtomography machines have a fixed number of pixels for each 2D projection, which makes the total number of voxels in the 3D render to be a fixed constant, dependent on the capabilities of the machine. Therefore, knowing that the total number of pixels, and therefore voxels, is fixed, then the bigger the domain the

lower will be the resolution, since every voxel needs to represent an increased length. In a similar fashion, if we diminish the domain size of the sample, we can achieve higher resolutions, because the voxels are going to represent smaller scales of size.

Our work is interested in studying porous materials, and their sampling is inherently stochastic, as the fibers are oriented in a random manner. Thus, it is necessary to obtain multiple X-ray micro-CT scans in order to average out the results of numerical simulations. Therefore, the experimental campaign to obtain multiple scans turns out to be very expensive and laborious, since we must take multiple scans due to stochasticity and due to the fact that the experimentalist never knows if he is actually taking a sample that is representative of the whole.

In order to formally define the REV, we recall from the literature that the REV is *the smallest physical volume over which a measurement can be made that will yield a value representative of the whole*. It is important to point out that the REV is property dependent, meaning that each material property has its own REV scale. For any scientist conducting an X-ray micro-CT, it is hard to know the proper size of the sample to take the microtomography, because there is a trade-off between physical domain size and resolution. The best would be for the experimentalist to take samples knowing the REV, but in practice it is very hard to know it exactly.

In order to illustrate this, Ferguson [9] performed a REV analysis on a digital domain with volume of 2000 voxel<sup>3</sup>, by changing the physical domain size of a material sample and observing the evolution of the normalized standard deviation obtained from the results. The domains were artificially generated using the PuMA software and the results can be observed in Figure 4. It is notable that the standard deviation has asymptotic behaviour for all variables, because as the physical domain size increases we are more sure to be within the range of macroscopic behaviour of the material, however each property computed has its own asymptotic behaviour, meaning that we would find different values of REV, depending on what property we are interested in.

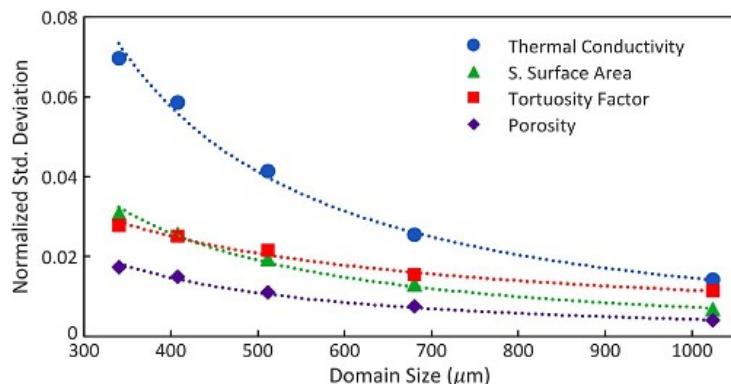


Figure 4: REV analysis conducted on a 2000 voxel<sup>3</sup> sample with isotropic random fibers, artificially generated using PuMA v2.1. Credits to [9].

In Figure 4, Domain size stands for the physical length of each side of a cubic domain.

In conclusion, our work is interested in studying porous materials, and since the fibers of these materials are oriented in a random manner, each different sample would generate different scan results. A scientist could take a scan of a big physical domain and zoom in different portions of the scan, at different scales, in order to search for the REV, however this approach will lead to low resolution. Another option would be to take multiple scans for each scale, in order to search the REV. Either way, it is an expensive and laborious process for any experimentalist, and at the end they can never be completely sure of the results because of the trade-off between resolution and domain size. Thus, we are going to rely on artificially generated domains to conduct the analysis on material response simulation results.

## 3 The Porous Microstructure Analysis (PuMA) software

### 3.1 Description of the software

The Porous Microstructure Analysis (PuMA) software was developed at the NASA Ames Research Center under a US & Foreign release with the main objective of performing response simulations on porous microstructures, in order to obtain the properties of the material. PuMA is an open-source software written in C++ with an additional version in Python [15] named pumapy. The software with all of its files can be downloaded via the official GitHub page [16]. The C++ version of the software is developed for Linux systems and optimized for multi-processor workstations.

There are many commercial softwares available, such as GeoDict [17] and Simpleware [18], designed for performing numerical response simulations on porous materials and analysing 3D models of porous microstructures. In addition, there are some academic softwares developed for the same purpose, such as the TauFactor [19]. However, these softwares are focused on low temperature applications, thus making the use of the PuMA software crucial on our study, as it was developed to study materials used in TPS applications.

PuMA is able to import digital three-dimensional images obtained from X-ray microtomography as TIFF files, and it can artificially generate the fibers of a material based on geometrical and mechanical constraints defined by the user.

Considering the difficulties inherent to any experimental X-ray micro-CT campaign, PuMA can be used exhaustively in order to generate artificial domains, eliminating the necessity of taking multiple scans, thus saving resources and time. Furthermore, it is possible to conduct a numerical analysis on the data collected from PuMA simulations in order to search for the REV.

Moreover, if we wish to obtain reference values for the material properties using real X-ray tomography scans it is very expensive, due to the fact that we must perform multiple scans to obtain averaged properties with good resolution. Additionally, there is the trade-off between resolution and domain size. Thus, using real tomography scans is not viable for obtaining reliable reference values.

In PuMA, we can compute numerical simulations for large domains and with good resolution, making it possible to obtain well founded averaged reference values for the physical properties of the materials. Nonetheless, performing numerical simulations of this magnitude is expensive and time consuming as well, hence being only a option for obtaining reference values for comparison with other simulation results with lower resolution and domain sizes.

### 3.2 Material properties in PuMA

All the material properties described in Section 2.1 are obtainable via PuMA. In the following, a brief description is given for understanding how PuMA compute these properties.

#### 3.2.1 Porosity in PuMA

Considering that the physical domain is imported or artificially generated in PuMA by using voxels to represent it in three dimensions, then it makes sense to compute volumes by simply counting the number of voxels. Therefore, given a phase  $\beta$ , we are going to call  $N_\beta$  the number of voxels inside this phase, meaning that the porosity will be simply given by:

$$\varepsilon = \frac{N_{\text{void}}}{N_{\text{tot}}} \quad (10)$$

Where  $N_{\text{void}}$  and  $N_{\text{tot}}$  are the number of voxels in the void phase and the total number of voxels inside the domain, respectively. Therefore, being  $N_{\text{solid}}$  the number of voxels in the solid phase, then the volume fraction of the solid phase would be simply given by  $\frac{N_{\text{solid}}}{N_{\text{tot}}} = 1 - \varepsilon$ .

### 3.2.2 Surface area and specific surface area in PuMA

The surface area given by Equation 3 and the specific surface area given by Equation 2 are both calculated in PuMA by using the Marching Cubes algorithm presented previously, therefore the procedure remain the same, as they were developed for an already digitized domain. Moreover, PuMA allows the user to visualize the domain in terms of the voxels that actually compose the physical fibers, by using a cuberille grid, meaning that the total surface area would be considered as equal to the sum of all exposed voxel faces of the solid. However, as presented by Ferguson in [9], this type of surface generation leads to errors that are beyond acceptable, meaning that the Marching Cubes algorithm is the more suited option.

### 3.2.3 Thermal and electrical conductivities in PuMA

Concerning the systems of partial differential equations 8 and 9, PuMA has many different solving methods available. One great remark is that since the heat equation is solved for the permanent regime with no heat generation, we can say that both systems of partial differential equations take the same form  $\nabla \cdot (\mathbf{A} \cdot \nabla B)$ , where  $\mathbf{A}$  is a second order tensor and  $B$  is a scalar field. Therefore, the same solving method used for the heat equation can be used for the electrical conduction equation. More specifically, the steady state heat equation without heat sources that PuMA solves is given by:

$$\nabla \cdot (\mathbf{k} \cdot \nabla T) = 0 \quad (11)$$

In order to solve the system of partial differential equations, PuMA can implement a simple finite difference method, the Explicit-jump finite difference method developed by Wiegmann et al. [20, 21], a recently added finite volumes method or a random walk method.

By considering all methods available, we are going to describe how the software deals with the thermal conductivity calculations, since they are the ones that matter the most for our study.

The two finite difference methods are implemented to compute the effective thermal conductivity of the porous microstructure by imposing a temperature gradient and solving it for the steady state temperature field. By knowing the temperature field, then the steady state heat flux can be calculated, meaning that we can compute the effective thermal conductivity of the material. The user can choose many different boundary conditions for the computation, which may or may not affect the final results, depending on the error associated with each different numerical method and boundary conditions settings. Further studies were developed for comparing the methods and the results obtained using the most variate boundary conditions in [5].

When comparing both finite difference methods, it is observable that the explicit jump method is faster than the simple finite difference method due to the fact that PuMA considers periodic boundary conditions in the simulation and side directions, which explains the difference between final results obtained when compared to the simple finite difference method. In addition, PuMA solves the linear system originated from the partial differential equations by using *Fast Fourier Transforms* (FFTs) [22, 23] and the bi-conjugate gradient stabilized method (BiCGSTAB) [24].

Additionally, PuMA v3.0 and above counts as well with tools for analysing anisotropic materials. There are anisotropic solvers for thermal and electrical conductivities available, however

these solvers require more computational power as they involve greater complexity and number of calculations. Nevertheless, it is a very interesting feature on PuMA in order to capture properties in a mesoscale, even if all constituents at the microscale are isotropic [15].

Another method PuMA uses for computing the thermal conductivities is the random walk method. Before diving into it it is necessary to comprehend the Knudsen number, as it plays a role in the diffusion regime of the porous microstructure. It is defined as follows:

$$K_n = \frac{\bar{\lambda}}{l_D} \quad (12)$$

Where  $\bar{\lambda}$  is the mean free path of the medium and  $l_D$  is a characteristic length, which can be taken as the diameter of the material's pores. It is a dimensionless number used to characterize the diffusion regime in porous microstructures. Regarding heat diffusion, it is used to describe effects in the micro- or nanoscale [25], as high Knudsen numbers suggest potential effects at these scales.

The random walk method consists on solving Equation 11 using Brownian motion to simulate the heat diffusion obtained from particle collisions inside the domain. This is a very general method that can be used for diffusion regimes with any Knudsen number, however this method takes considerably more time to run. It works by imposing the mean thermal velocity  $\bar{v}$  and mean free path  $\bar{\lambda}$ . Thereafter, particles are given random velocity vectors and free path based on an exponential distribution. One thing to point out is that the continuum assumption ( $K_n \ll 1$ ) is widely accepted as true in most common conditions, such as atmospheric pressure and temperature. This method is highly recommended when running simulations for higher Knudsen numbers, since the micro- or nanoscales effects cannot be ignored in these situations. Considering this and the increased time needed to run simulations using the random walk method, we can say that this method is not well suited for the present study. However it is an important feature present in PuMA for researchers to use in other works.

Finally, there is now a finite volume solver available for PuMA v3.0 and above. Further studies need to be carried out to compare this method with the others when it comes to computation time required and accuracy of the results obtained. Nonetheless, the finite volume method is widely known for being advantageous for conserving quantities at discretized level and for taking full advantage of meshes to approximate complex geometries [26]. It can generate higher accuracy than the finite differences method when using small mesh density as well [27].

### 3.2.4 Further possibilities with PuMA

Indeed, PuMA has many more capabilities that have not been made explicit here, due to the fact that they do not play an important role on this work. Nonetheless, it is important to point out the potential use of PuMA for future research, as it is a very versatile software for analysing porous microstructures.

The software can perform material orientation estimation, computation of elasticity tensor, computation of stress and deformation fields, computation of tortuosity factors and permeability. In addition, PuMA can artificially generate many different types of microstructures other than fiber based microstructures, such as microstructures with hollow regions and complex cross-sections, microstructures based on spherical components and even well ordered periodic microstructures such as *Triply Periodic Minimal Surfaces* (TPMS) or periodic structures with spherical components, and even advanced woven structures.

PuMA can be used as well to identify internal pores and artificially generate microstructures with more than one phase, in order to study multiphase response simulations on porous microstructures.

Finally, another great feature of PuMA that needs to be mentioned is the ability to perform microscale oxidation simulations, and by using this one can study the evolution of the material

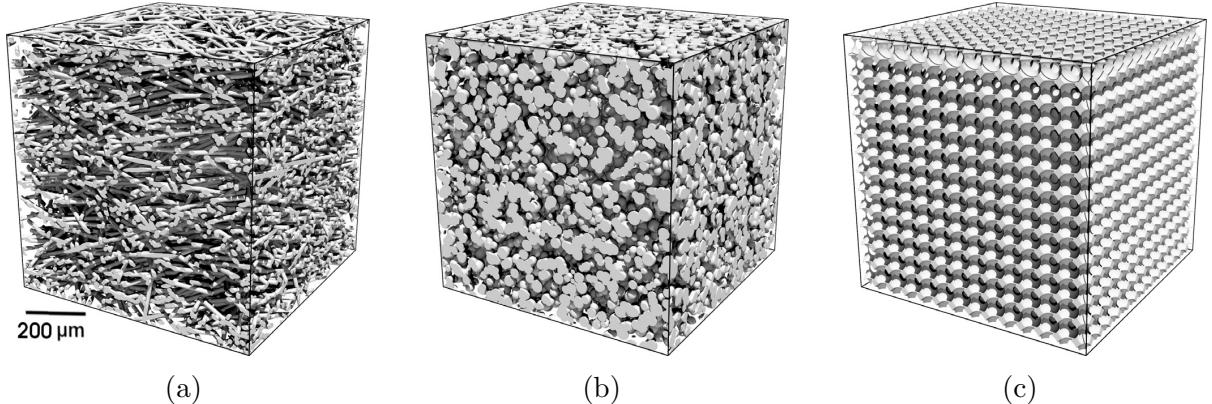


Figure 5: Examples of different microstructures generated with the use of PuMA v2.1: (a) Random transversely isotropic fiber preform, (b) Packed sphere bed with varying sphere diameters, and (c) Period fom structure with uniform void diameter. Credits to [9].

properties previously described as the material is decomposing. Multiphase domains are allowed for oxidation simulations as well, meaning that it can be used as a tool to study the oxidation of resin infused ablative materials, that are of major importance.

All the capabilities of the PuMA software are well explored within the files available at the official GitHub page [16], where new users can explore all the tutorials in python and C++ in order to fully experience the PuMA software at its finest.

## 4 Mathematical Formulation of the problem

### 4.1 PuMA workspace variables and definitions

In order to better comprehend and work with PuMA, we need to formally define the variables inside the workspace. Thus, we denote  $R$  and  $D$  the radius of the fiber and the domain length, respectively. Here we are considering the domain to be a cube of lengths  $D$ , thus with volume  $D^3$ . In fact, the domain could be of sizes  $L_x$ ,  $L_y$  and  $L_z$ , but we are considering  $L_x = L_y = L_z = D$  for simplicity reasons.

Moreover, the subscript  $phy$  is used to indicate a physical quantity, while the subscript  $vx$  indicates the respective number of voxels for the specific quantity. At the same time, upper case letters will be used for the geometrical quantities and lower case letters for the digitized PuMA variables. By considering this notation, then  $R_{phy}$  is the average radius of the material's physical fiber, while  $r_{vx}$  is the number of voxels used to represent the radius of the fibers with PuMA.

Additionally, let us denote the average length of the material's physical fiber as  $L_{phy}$  and the number of voxels used to represent the average length of the fibers as  $l_{vx}$ , while  $D_{phy}$  and  $d_{vx}$  are going to be used to denote the length of the physical cubic domain and the number of voxels used to represent the length of the physical cubic domain, respectively.

In addition to the geometrical parameters on PuMA, there are also the values of standard deviation for the fiber radius and the fiber length, which will be noted as  $\sigma_{R_{phy}}$  and  $\sigma_{L_{phy}}$  for the physical quantities. Their respective quantities as number of voxels in PuMA are represented by  $\sigma_{r_{vx}}$  and  $\sigma_{l_{vx}}$ , respectively.

Furthermore, as this work is focused on fibrous porous materials, hence the fibers can be oriented in any manner in the 3D space. By consequence, we must insert the maximum variation in the angle distribution of the fibers as input on PuMA. Let  $\theta_x$ ,  $\theta_y$  and  $\theta_z$  be the angles that the fiber form with the  $x$ ,  $y$  and  $z$  axis, respectively. Moreover, let  $\Delta\theta_x$ ,  $\Delta\theta_y$  and  $\Delta\theta_z$  be the maximum angle variation that the fibers can have with respect to the  $x$ ,  $y$  and  $z$  axis, respectively. Therefore, PuMA will insert fibers inside the domain with angles  $\theta_i \in [-\Delta\theta_i; +\Delta\theta_i]$ , where  $i$  is the subscript to represent the  $x$ ,  $y$  or  $z$  direction. Using this notation, then PuMA can generate isotropic microstructures using  $\Delta\theta_x = \Delta\theta_y = \Delta\theta_z = 90^\circ$  as input parameters. Additionally, a transversely isotropic microstructure would require the angular variation inputs to be  $\Delta\theta_x = \Delta\theta_y = 90^\circ$  and  $\Delta\theta_z \neq 90^\circ$ , where the  $z$  axis has chosen here without loss of generality. By manipulating the angular variation input, PuMA can generate many sorts of different geometries for the microstructures of the porous materials.

Additionally, one can note that as  $r_{vx}$  increases, we are using more voxels to represent the same physical measurement, thus causing the rendering to be more precise. To obtain the conversion between physical measurement and number of voxels used in PuMA we can use the voxel length  $\ell_{vx}^{phy}$ , defined here by

$$\ell_{vx}^{phy} = \frac{R_{phy}}{r_{vx}} \quad (13)$$

It is noticeable that  $\ell_{vx}^{phy}$  can be seen as a measure of resolution for the PuMA software, as it can define the accuracy of the fibers geometry. Furthermore, since  $R_{phy}$  is fixed for a given material, then if we desire to change  $r_{vx}$ , we must change  $\ell_{vx}^{phy}$  by the same proportion. The effects of changing the value of  $r_{vx}$  can be seen in Figure 6.

Using the definition of  $\ell_{vx}^{phy}$ , we can obtain the value of  $D_{phy}$  as a function of  $r_{vx}$ . By knowing that  $D_{phy} = d_{vx}\ell_{vx}^{phy}$ , then:

$$D_{phy} = \frac{d_{vx}R_{phy}}{r_{vx}} \quad (14)$$

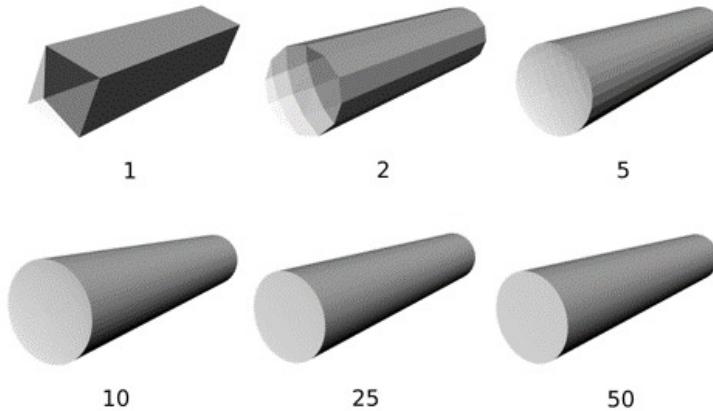


Figure 6: Comparison between different fibers obtained by changing the value of  $r_{vx}$ . Credits to [11]

## 4.2 Limitations of real X-ray microtomography scans imposed on PuMA

Since the number of voxels in each scan of the X-ray microtomography is fixed, then  $d_{vx}$  must be fixed. This will cause the total number of voxels present on the digitized domain to be fixed, when considering PuMA as a mean to generate samples that would be gathered using real X-ray microtomography scans. This limitation will be important when comparing these samples with samples that are obtained solely by using more computational power with PuMA.

Using also the fact that  $R_{phy}$  and  $d_{vx}$  are fixed in a real tomography scan, we can say that  $D_{phy} \propto \frac{1}{r_{vx}}$ . In a real microtomography scan machine, there is a minimum value for the voxel length available, which denotes the maximum resolution of the machine. This voxel length is denoted as  $\ell_{min}$ , and by using Equation 13 then there is a maximum value for  $r_{vx}$  that can be used to generate samples with PuMA that resembles the ones that could be obtained with X-ray micro-CT machines, which will be called  $r_{max}$ . In summary, in order to use PuMA to generate results obtainable via real X-ray micro-CT scans, then:

- $\ell_{vx}^{phy} \geq \ell_{min} \Leftrightarrow r_{vx} \leq r_{max}$
- $d_{vx} = \text{constant}$

In addition, in order to capture a bigger domain the value of  $r_{vx}$  must be decreased, therefore increasing  $\ell_{vx}^{phy}$ , causing a loss in resolution. Mathematically, this is the trade-off between resolution and physical sample size. As a sample has higher resolution, it is mandatory to diminish the physical domain length  $D_{phy}$  captured.

Further on,  $l_{vx}$  is an input parameter in PuMA, and since we have defined  $\ell_{vx}^{phy}$  with the use of the fiber's radius, it is only necessary to correctly insert the value of  $l_{vx}$ , so that  $L_{phy} = \ell_{vx}^{phy} l_{vx}$ . Consequently, this parameter is not going to be an influence on the results obtained by PuMA, *making  $r_{vx}$  the only geometric parameter that will influence the numerical study for microtomography obtainable results*. If we were able to change the value of  $d_{vx}$ , then we could in fact have  $d_{vx}$  and  $r_{vx}$  as parameters to influence the results of further numerical study on the samples generated by PuMA.

## 4.3 Mathematical point of view

Let us introduce the function  $h(\mathbf{y}, \xi)$ , where  $\mathbf{y}$  is a vector containing geometrical parameters and  $\xi \in \Xi$  is the set of uncertain parameters to take into account, such as the standard deviations of the fiber's radius and fiber's length. Therefore, we can say that  $h$  is stochastic.

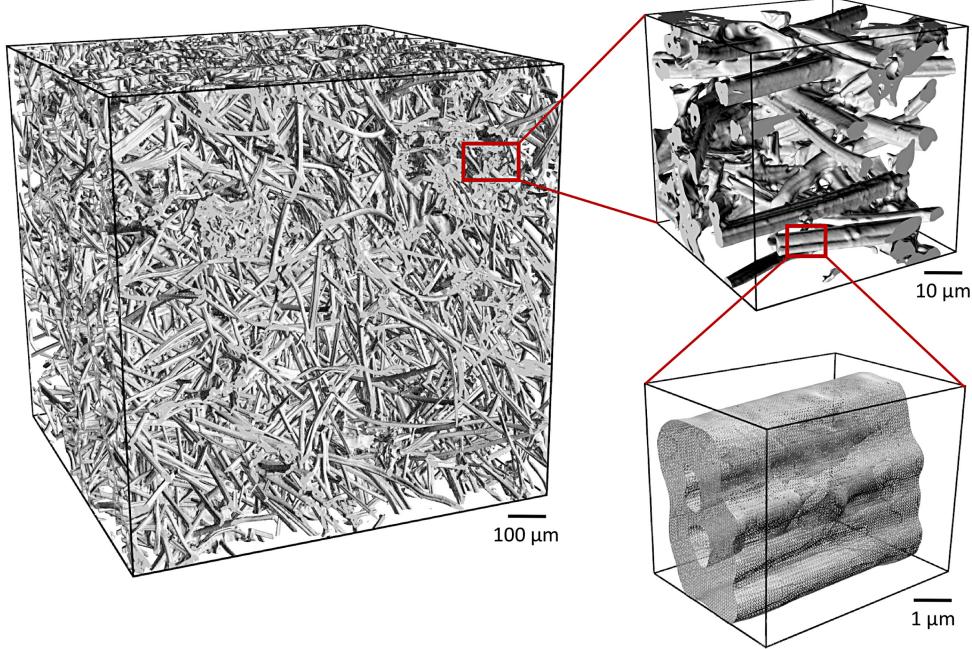


Figure 7: Surface renderings of a porous material X-ray micro-CT, using three different resolutions and domains, using PuMA v2.1. Credits to [9].

Considering the use of PuMA to acquire microstructures that could be obtained via X-ray micro-CT scans, then  $\ell_{vx}^{phy}$  or  $r_{vx}$  are the only free parameters that we can change, meaning that  $h = h(\ell_{vx}^{phy}, \xi)$  or  $h = h(r_{vx}, \xi)$ . The choice between manipulating the value of  $\ell_{vx}^{phy}$  or  $r_{vx}$  is free, since they are correlated by the fiber radius  $R_{phy}$ , so it really depends on the analysis being conducted. It is important to keep in mind that  $r_{vx}$  can only assume integer values, whereas  $\ell_{vx}^{phy}$  can assume any real value, as long as  $r_{vx}$  is rounded to the closest integer.

In addition, the function  $h$  could represent any property related to porous materials, but in our study  $h$  will be the thermal conductivity of the material. In addition, the off-diagonal terms of the thermal conductivity tensor (Equation 7) are at least one order of magnitude smaller than the diagonal terms, but in general they can be approximately 2 orders of magnitude smaller [11, 5]. Therefore, the present work will focus on analysing the diagonal terms  $k_{xx}$ ,  $k_{yy}$  and  $k_{zz}$ , as they are more crucial in any thermal response simulation or experiment.

Considering the REV from a mathematical point of view, we have that it is equivalent to the minimal physical domain volume for which we obtain a thermal conductivity measure  $k_{ii}$  that represents the bulk thermal conductivity of the material, being  $i$  the subscript to represent the  $x$ ,  $y$  or  $z$  direction. This is equivalent to stating that the search for the REV is the search for minimal value of  $D_{phy}$  for which  $h$  converges towards the bulk material property values.

However, the variance of the results obtained by PuMA must be taken into account as well when searching for the REV. Therefore, not only we must search for thermal conductivities that are close enough to the bulk values, but also we must minimize the distance between  $\text{Var}(h)$  and  $\eta(h)$ , where  $\text{Var}(h)$  is the variance of  $h$  computed with respect to the uncertainty of the problem (stochasticity and parameter uncertainties), and  $\eta(h)$  is a reference value for the variance of  $h$ .

Note that as we increase the resolution of the fibers by increasing  $r_{vx}$ , we decrease the voxel length  $\ell_{vx}^{phy}$ , and if we were to consider the limitations of X-ray micro-CT scans, meaning that  $d_{vx} = \text{constant}$ , then we actually decrease the value of  $D_{phy}$ . Thus, augmenting the resolution of the domain does not imply in better accuracy of the results in this case, as the physical domain will actually get smaller. The fact that  $d_{vx}$  is constant makes every change in  $r_{vx}$  imply in a change in  $D_{phy}$ , as  $D_{phy} \propto \frac{1}{r_{vx}}$ .

Additionally, since we want to minimize the value of  $|\text{Var}(h) - \eta(h)|$  but we cannot know

for certain if we are using the appropriate physical domain size (thus if we are indeed using the REV), we can converge our simulations results towards the wrong answer, due to the fact that the resolution of the fibers may not good enough. Hence, there is an urging need for a reference value.

In order to verify if the collected samples from PuMA are in the REV scale, we must check if the results obtained from the thermal simulations are close enough to the reference value. Since the results from the simulations are random, we must use the average, or mean value, of  $h$  in order to make comparisons, and we are going to denote  $\mu(\cdot)$  as the average, or arithmetic mean, of any set of data considered as argument. Therefore, we must minimize the variance of our results and we must minimize as well the value of  $|\mu(h) - \mu^{ref}(h)|$ , where  $\mu(h)$  is the average of  $h$ , while  $\mu^{ref}(h)$  is the reference value for the average of  $h$ .

Our problem consists in minimizing both  $|\mu(h) - \mu^{ref}(h)|$  and  $|\text{Var}(h) - \eta(h)|$  at the same time by changing  $r_{vx}$ , while satisfying some criteria. It is clear then that we find ourselves with a bi-objective optimization under uncertainty problem, with  $r_{vx}$  as the single free parameter. The summarized formulation of the problem is written as follows:

$$\begin{aligned}
 & \text{minimize: } |\mu(h) - \mu^{ref}(h)| \\
 & \text{minimize: } |\text{Var}(h) - \eta(h)| \\
 & \text{satisfying: } \ell_{vx}^{phy} \geq \ell_{min} \\
 & \text{satisfying: } \ell_{vx}^{phy} = \frac{R_{phy}}{r_{vx}} \\
 & \text{satisfying: } d_{vx} = \text{constant} \\
 & \text{by changing: } r_{vx}
 \end{aligned} \tag{15}$$

Two approaches will be analysed in sections 6 and 7. The first one is a brute force search for the REV, using a massive number of simulations and manually extracting the optimal values of  $r_{vx}$  in order to minimize both functions of interest, providing by consequence the respective value of  $D_{phy}$ , which will be considered as a first approximation of the REV. Later on, an optimization algorithm that seeks to reduce the number of calculations necessary to determine the REV is going to be used, by considering a probabilistic framework to study this bi-objective optimization under uncertainty problem. Both approaches requires the computation of the reference values, as well as determining the reference values for variances, here denoted by the Greek letter  $\eta$ .

In order to obtain the value of  $\mu^{ref}(h)$  and  $\eta(h)$  for the  $x$ ,  $y$  and  $z$  directions, a series of thermal simulations in PuMA were performed, using the maximal computational power possible, in order to use the highest value of  $r_{vx}$  possible, alongside with the correspondent value of  $\ell_{vx}^{phy}$  and with the highest value of  $d_{vx}$  possible. By doing so, the largest possible physical domain will be captured, and we are going to assume that this domain is able to represent the global behaviour of the material. This analysis is conducted in Section 6. Furthermore, the variance thresholds for the  $x$ ,  $y$  and  $z$  directions are going to be the variances related to the samples that are considered for computing the reference values  $\mu^{ref}(h)$ .

The platform used for all the massive computations of this work is the *Plateforme Fédérative pour la Recherche en Informatique et Mathématiques* (PlaFRIM) cluster, which has given the access for use due to the fact that this work has been conducted at Inria. It is important to note that PlaFRIM is an experimentation platform. Hence, PlaFRIM indicates in its documentation the search of a large production cluster for running large and long jobs, as PlaFRIM has its own limitations.

## 5 Introduction to constrained multi-objective optimization under uncertainty

### 5.1 Deterministic constrained multi-objective optimization

As basis for a thorough explanation on the concepts necessary to understand constrained multi-objective optimization under uncertainty problems and how it applies to this work with PuMA, we are going to rely strongly on chapter two of [28], as it is a very didactic work for those with no basis on the subject. Some changes in notation have been done in order to make the subject a little but more clear for the reader. The understanding of these concepts is crucial for solving the problem formulated in Section 4.3.

Consider a collection of input parameters as being a *Design of Experiment* (DoE), and every DoE is used to compute a number of output results. Using this data, we can compute statistical measures and consider them as *Objective Functions*  $f_i, \forall i \in \mathbb{N}^*$  or *Constraint Functions*  $g_i, \forall i \in \mathbb{N}^*$ , where the constraints take the form  $g_i \leq 0$ .

Every DoE has its own *Design Vector*  $\mathbf{x}$ , containing the output data measured from the inputs contained in the respective design. In addition, we are going to consider that every DoE resides in the *Design Space*  $\chi$ . For example, a DoE  $a \in \chi$  will have a design vector named  $\mathbf{x}_a$ , which contains the output data measured from the inputs of  $a$ . In the same manner, we can consider another DoE  $b \in \chi$  with design vector  $\mathbf{x}_b$ .

It is crucial to compute the objective functions and constraint functions using the design vectors as argument, so we are going to collect these statistical measures in vectors, which resides in the objective and constraint space. In this work, these vectors are called *Evaluated Vectors*, and the ones related to  $a$  and  $b$  are made explicit in Equation 16, where it is considered that there are  $m_1$  objective functions and  $m_2$  constraint functions.

$$\mathcal{A} = \begin{bmatrix} f_1(\mathbf{x}_a) \\ f_2(\mathbf{x}_a) \\ f_3(\mathbf{x}_a) \\ \vdots \\ f_{m_1-1}(\mathbf{x}_a) \\ f_{m_1}(\mathbf{x}_a) \\ g_1(\mathbf{x}_a) \\ g_2(\mathbf{x}_a) \\ g_3(\mathbf{x}_a) \\ \vdots \\ g_{m_2-1}(\mathbf{x}_a) \\ g_{m_2}(\mathbf{x}_a) \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} f_1(\mathbf{x}_b) \\ f_2(\mathbf{x}_b) \\ f_3(\mathbf{x}_b) \\ \vdots \\ f_{m_1-1}(\mathbf{x}_b) \\ f_{m_1}(\mathbf{x}_b) \\ g_1(\mathbf{x}_b) \\ g_2(\mathbf{x}_b) \\ g_3(\mathbf{x}_b) \\ \vdots \\ g_{m_2-1}(\mathbf{x}_b) \\ g_{m_2}(\mathbf{x}_b) \end{bmatrix} \quad (16)$$

This way, we have that  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^m$ , where  $m = m_1 + m_2$ . We are going to consider that  $\mathcal{A}, \mathcal{B} \in X \subset \mathbb{R}^m$ , where  $X$  is the set containing all evaluated vectors. By considering that we want to minimize multiple objective functions, then it is necessary to rank the designs in such a way that we can find the optimal designs.

Firstly, consider two vectors  $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$ . The classical Pareto dominance is given with specific notation by the following set of criteria:

$$\begin{aligned} \mathbf{w} \succ \mathbf{v} \ (\mathbf{w} \text{ dominates } \mathbf{v}) &\iff \forall j \in \llbracket 1, n \rrbracket, \mathbf{w}_j \leq \mathbf{v}_j \text{ and } \exists j \in \llbracket 1, n \rrbracket, \mathbf{w}_j < \mathbf{v}_j \\ \mathbf{w} \succ \mathbf{v} \ (\mathbf{w} \text{ strictly dominates } \mathbf{v}) &\iff \forall j \in \llbracket 1, n \rrbracket, \mathbf{w}_j < \mathbf{v}_j \\ \mathbf{w} \sim \mathbf{v} \ (\mathbf{w} \text{ is indifferent to } \mathbf{v}) &\iff \mathbf{w} \not\succ \mathbf{v} \text{ and } \mathbf{v} \not\succ \mathbf{w} \end{aligned} \quad (17)$$

This dominance rule can be extended to compare two designs  $a$  and  $b$  under the optics of objective and constraint functions  $f$  and  $g$ . Consider that  $\cdot_f$  and  $\cdot_g$  represents the vectors obtained by evaluating the design vectors using the objective functions or the constraint functions. Therefore, given a DoE  $a \in \chi$ , we have:

$$a_f = \begin{bmatrix} f_1(\mathbf{x}_a) \\ f_2(\mathbf{x}_a) \\ f_3(\mathbf{x}_a) \\ \vdots \\ f_{m_1-1}(\mathbf{x}_a) \\ f_{m_1}(\mathbf{x}_a) \end{bmatrix}, \quad a_g = \begin{bmatrix} g_1(\mathbf{x}_a) \\ g_2(\mathbf{x}_a) \\ g_3(\mathbf{x}_a) \\ \vdots \\ g_{m_2-1}(\mathbf{x}_a) \\ g_{m_2}(\mathbf{x}_a) \end{bmatrix} \quad (18)$$

This way, the evaluated vector  $\mathcal{A}$  simply writes as  $\mathcal{A} = \begin{bmatrix} a_f \\ a_g \end{bmatrix}$ . By using the Pareto dominance criteria defined in Equation 17, then the Pareto dominance under constraint criteria between the designs  $a$  and  $b$  is given by the following:

For two designs  $a, b \in \chi$  with respective evaluated vectors  $\mathcal{A}, \mathcal{B} \in X \subset \mathbb{R}^m$ :

$$\begin{aligned} \mathcal{A} \succ_c \mathcal{B} &\iff a \in \mathbb{A} \text{ and } a_f \succ b_f \text{ or } b \in \mathbb{F} \\ \mathcal{A} \succ\!\succ_c \mathcal{B} &\iff a \in \mathbb{A} \text{ and } a_f \succ\!\succ b_f \text{ or } b \in \mathbb{F} \\ \mathcal{A} \sim_c \mathcal{B} &\iff a_f \not\succ b_f \text{ and } b_f \not\succ a_f \end{aligned} \quad (19)$$

Where  $\mathbb{A}$  and  $\mathbb{F}$  stands for the *Admissible* and *Failure* sets. Considering that there are  $m_1$  objective functions,  $m_2$  constraint functions and that  $m_1 + m_2 = m$ , then the admissible and failure sets are defined as:

$$\begin{aligned} \mathbb{A} &= \{c \in \chi \mid \forall i \in [1, m_2], c_{g,i} \leq 0\} \\ \mathbb{F} &= \{c \in \chi \mid \exists i \in [1, m_2], c_{g,i} \not\leq 0\} \end{aligned} \quad (20)$$

Where  $c_{g,i}$  is the  $i$ th element of the vector  $c_g$ . By consequence of this definition,  $\mathbb{F}$  is the complement of  $\mathbb{A}$ , and  $\mathbb{F} \cup \mathbb{A} = \chi$ . In principle, the admissible set is the set containing all the designs in which all constraints are satisfied, while the failure set is the set containing all designs in which at least one constraint is not satisfied. One can note that the Pareto dominance defined in Equation 19 can be reduced to the classical Pareto dominance (Equation 17) just by removing the constraint functions  $g$  of the optimization problem.

Since all designs resides in the design space  $\chi$ , then we define the set  $\mathbb{P}(X)$  as the set containing all the *Pareto-Optimal* designs in  $\chi$  and we name it the *Pareto Front* of the problem with respect to  $X$ , the set containing the evaluated vectors. We can formally define the Pareto front in the constrained scenario as follows:

$$\mathbb{P}(X) = \{a \in \chi \mid \nexists b \in \chi, \mathcal{B} \succ_c \mathcal{A}\} \quad (21)$$

In Equation 21,  $\mathcal{A}$  and  $\mathcal{B}$  are the evaluated vectors of designs  $a$  and  $b$ , respectively. As a way to illustrate all these concepts, consider a bi-objective minimization problem with 5 designs  $a, b, c, d, e \in \chi$ , with respective evaluated vectors  $A, B, C, D, E \in X$ , as shown in Figure 8. Consider as well a constraint function  $g_1(\mathbf{x}) = f_2 - 1.9 \leq 0$ , meaning that  $X \subset \mathbb{R}^3$ . This way, we can note that  $a, d \in \mathbb{F}$ , while  $c, b, e \in \mathbb{A}$ . Furthermore, we have that  $\chi = \{a, b, c, d, e\}$ , and  $\mathbb{P}(X) = \{C, B\}$ , since they are the only evaluated vectors that are not being dominated, considering the Pareto dominance criteria in Equation 19. Therefore, the Pareto-optimal designs are  $c$  and  $b$ .

In summary, a deterministic constrained multi-objective optimization problem is described by the following:

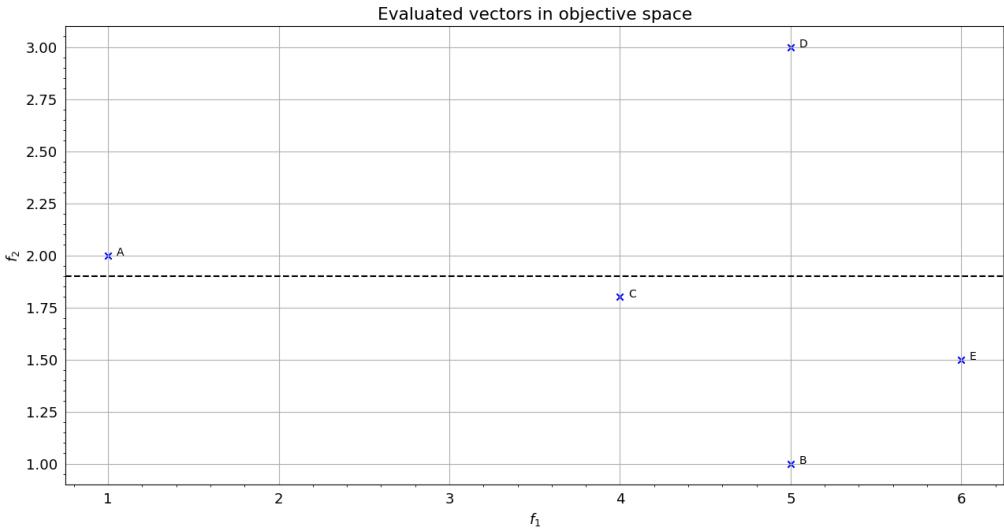


Figure 8: Simple example to illustrate the search for the Pareto-optimal designs.

$$\begin{aligned}
 & \text{minimize: } f(\mathbf{x}_a) \\
 & \text{satisfying: } g(\mathbf{x}_a) \leq 0 \\
 & \text{by changing: } a \in \chi
 \end{aligned} \tag{22}$$

In essence, changing the DoE means changing the input parameter values, in order to change the design vectors and thus change the evaluated vectors. The fundamental objective is to find the Pareto-optimal designs, given the optimization problem and context.

## 5.2 Optimization under uncertainty

Consider now that the objective functions  $f$  and constraint functions  $g$  depends on uncontrollable parameters, denoted in this work by  $\xi \in \Xi$ . These parameters are uncertain, and we are going to consider them as aleatoric uncertainties, instead of epistemic uncertainties. Therefore, we have that  $\forall a \in \chi$ , then  $f(\mathbf{x}_a)$  and  $g(\mathbf{x}_a)$  are random variables, meaning that we must adapt the general formulation of the constrained multi-objective optimization problem, presented in Equation 22.

Due to uncertainty, it is best to formulate the problem using statistical measures as objective and constraint functions. Therefore, consider  $\rho_f$  and  $\rho_g$  to be statistical measures to be considered as objective and constraints, respectively. Then, the optimization under uncertainty problem is given by:

$$\begin{aligned}
 & \text{minimize: } \rho_f(\mathbf{x}_a) \\
 & \text{satisfying: } \rho_g(\mathbf{x}_a) \leq 0 \\
 & \text{by changing: } a \in \chi
 \end{aligned} \tag{23}$$

When using statistical measures of the objective functions  $f$  as objectives, the problem is said to be a *Robust Design Optimization* (RDO), and the statistics  $\rho_f$  are called *Robustness Measures*. In a similar fashion, by using statistical measures of the constraint functions  $g$  as constraint functions, we find ourselves with a *Reliability-Based Design Optimization* (RBDO) problem, in which constraint statistics are denoted by  $\rho_g$  and called *Reliability Measures*. In this context, the evaluated vector is formally written as follows:

$$\forall a \in \chi, \mathcal{A} = \rho(\mathbf{x}_a) = \begin{bmatrix} \rho_f(\mathbf{x}_a) \\ \rho_g(\mathbf{x}_a) \end{bmatrix} \quad (24)$$

Meanwhile, the original functions  $f$  and  $g$  are gathered in a vector  $q \in \mathbb{R}^m$ , which stands for *Quantities of Interest* (QoI). Formally, we have that

$$\forall (a, \xi) \in \chi \times \Xi, q(\mathbf{x}_a, \xi) = \begin{bmatrix} f(\mathbf{x}_a, \xi) \\ g(\mathbf{x}_a, \xi) \end{bmatrix} \quad (25)$$

Since robustness and reliability measures are based on statistical measures, it can be any construction of statistical functions. The most common ones are expectation and variance, but other common choices are the maximum value, quantile and superquantile.

### 5.3 Probabilistic framework

Given the uncertainties related to the computations of  $f$  and  $g$ , then definitely  $\rho_f$  and  $\rho_g$  are not exact as well, but rather computed with a certain level of accuracy. Therefore, it is natural to consider that  $\forall a \in \chi, \rho(\mathbf{x}_a)$  is a random variable. Hence, we propose to estimate the statistical measures  $\rho$  with a random field  $\mathbf{P}$ , and we are going to explore the extension of the classical Pareto dominance under the scope of a probabilistic approach, since it is the most general way to analyse an optimization problem.

Consider two random vectors  $\mathcal{A} = \rho(\mathbf{x}_a)$  and  $\mathcal{B} = \rho(\mathbf{x}_b)$  that resides in the objective and constraint space, which are the evaluated vectors of designs  $a$  and  $b$ . When considering a way to determine the dominance relation between the two evaluated vectors, it is necessary make use of the probability  $P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ_c \mathcal{B}]$ , computed over the joint distribution of  $\mathcal{A}$  and  $\mathcal{B}$ .

When considering the constrained scenario, we have that

$$P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ_c \mathcal{B}] = \int_{b \in \mathbb{F}} \phi_{\mathcal{B}}(\mathbf{x}_b) d\mathbf{x}_b + \iint_{\substack{a \in \mathbb{A} \\ b \in \mathbb{A} \\ \mathcal{A} \succ \mathcal{B}}} \phi_{\mathcal{A}, \mathcal{B}}(\mathbf{x}_a, \mathbf{x}_b) d\mathbf{x}_a d\mathbf{x}_b \quad (26)$$

Where  $\phi$  stands for *Probability Density Function* (PDF). Thus,  $\phi_{\mathcal{A}, \mathcal{B}}$  is the joint PDF of  $\mathcal{A}$  and  $\mathcal{B}$ , While  $\phi_{\mathcal{B}}$  is the PDF of  $\mathcal{B}$ . Based on Equation 26, then the probabilistic constrained Pareto dominance rule can be formulated in the following way:

*For two designs  $a, b \in \chi$  with respective evaluated vectors  $\mathcal{A}, \mathcal{B} \in X \subset \mathbb{R}^m$  as explicit images of the designs in the objective and constraint random field  $\mathbf{P}$ :*

$$\begin{aligned} \mathcal{A} \succ_{c,\delta} \mathcal{B} &\iff P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ_c \mathcal{B}] \geq 1 - \delta \\ \mathcal{A} \succ\succ_{c,\delta} \mathcal{B} &\iff P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ\succ_c \mathcal{B}] \geq 1 - \delta \\ \mathcal{A} \sim_{c,\delta} \mathcal{B} &\iff \mathcal{A} \not\succ_{c,\delta} \mathcal{B} \text{ and } \mathcal{B} \not\succ_{c,\delta} \mathcal{A} \end{aligned} \quad (27)$$

The above formulation is the  $\delta$ -relaxed probabilistic constrained Pareto dominance. It considers the case of random evaluated vectors with infinite support [28], preventing the probability to be exactly equal to 1, and this approach has been proposed in [29]. This relaxation can be removed, and by doing so we end up with the following definition:

*For two designs  $a, b \in \chi$  with respective evaluated vectors  $\mathcal{A}, \mathcal{B} \in X \subset \mathbb{R}^m$  as explicit images of the designs in the objective and constraint random field  $\mathbf{P}$ :*

$$\begin{aligned} \mathcal{A} \succ_{c,0} \mathcal{B} &\iff P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ_c \mathcal{B}] = 1 \\ \mathcal{A} \succ\succ_{c,0} \mathcal{B} &\iff P_{\mathcal{A}, \mathcal{B}}[\mathcal{A} \succ\succ_c \mathcal{B}] = 1 \\ \mathcal{A} \sim_{c,0} \mathcal{B} &\iff \mathcal{A} \not\succ_{c,0} \mathcal{B} \text{ and } \mathcal{B} \not\succ_{c,0} \mathcal{A} \end{aligned} \quad (28)$$

## 5 INTRODUCTION TO CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

---

Note that the relaxed probabilistic Pareto dominance defined in Equation 27 provides us a very general framework for comparing random vectors in the objective and constraint space. However, this relaxation causes the Pareto front to contain many more non-dominated points, which means that we must determine a way to rank all the random vectors, as a manner of selecting the most optimal designs. For this, we are going to use the *Pareto-Optimal Probability* (POP).

The POP is a value calculated for all random evaluated vectors in order to quantitatively rank them all. It is formally defined as follows [28]:

$\forall a \in \chi$  with respective evaluated vector  $\mathcal{A} \in X \subset \mathbb{R}^m$  as explicit image of the design in the objective and constraint random field  $\mathbf{P}$ :

$$\text{POP}(\mathcal{A}) = P_X [\mathcal{A} \in \mathbb{P}(X)] = \int \phi_X(\mathbf{x}_a) \mathbb{1}_{\mathbb{P}(X)}(\mathcal{A}) d\mathbf{x}_a \quad (29)$$

Where  $\mathbb{1}_\Omega$  is the indicator function of set  $\Omega$ . This general definition is simply the probability for one random vector to be non-dominated, considering the joint PDF with respect to all other ones  $\phi_X$ . This is a very general formulation, and the computational cost is enormous, due to the fact that in real applications there could be hundreds of vectors to be compared, with unknown probability density functions. Therefore, an alternative definition for approximating the POP is to be proposed.

Consider we have  $N \in \mathbb{N}^*$  designs in  $\chi$ , which by consequence have  $N$  evaluated vectors in  $X$ . Let us denote the designs and evaluated vectors generically by  $a_i$  and  $\mathcal{A}_i$ , for  $i \in \llbracket 1, N \rrbracket$ . Therefore, we have that  $\chi = \{a_i\}_{i=1}^N$  and  $X = \{\mathcal{A}_i\}_{i=1}^N$ , and we define the POP of any evaluated vector as being

$$\text{POP}(\mathcal{A}_i) = P_X(\mathcal{A}_i \in \mathbb{P}(X)) = P_X \left[ \bigcap_{\substack{\mathcal{A}_j \in X \\ j \neq i}} \mathcal{A}_j \not\succ_c \mathcal{A}_i \right] \quad (30)$$

Where  $P_X$  is calculated considering the joint PDF of all elements of  $X$ . In order to avoid the complexity of the calculations involved in Equation 30 and the intersection between  $N - 1$  events that would result in high computational cost, there are ways to approximate the POP. A first approach would be to average out the individual domination comparison probabilities. By considering this idea, we can formulate the  $\text{POP}_{av}$  indicator as follows:

$$\text{POP}_{av}(\mathcal{A}_i) = \frac{1}{N-1} \sum_{\substack{\mathcal{A}_j \in X \\ j \neq i}} P_{\mathcal{A}_j, \mathcal{A}_i} [\mathcal{A}_j \not\succ_c \mathcal{A}_i] \quad (31)$$

Due to the probability being calculated only by performing one-to-one comparisons, the computational burden associated with Equation 31 is very small. However, this formulation is not good enough because we can have  $\text{POP}(\mathcal{A}_i) \neq 0$  even when  $\mathcal{A}_i$  gets dominated by another vector in  $X$ . Therefore, another formulation is proposed, by considering the minimal value of the one-to-one individual domination comparison probabilities between all evaluated vectors, which we denote as  $\text{POP}_{min}$ . It is written as follows:

$$\text{POP}_{min}(\mathcal{A}_i) = \min_{\substack{\mathcal{A}_j \in X \\ j \neq i}} (P_{\mathcal{A}_j, \mathcal{A}_i} [\mathcal{A}_j \not\succ_c \mathcal{A}_i]) \quad (32)$$

This formulation satisfies the following:

$$\exists j, \mathcal{A}_j \succ_{c,0} \mathcal{A}_i \implies \text{POP}_{min}(\mathcal{A}_i) = 0$$

Both approximations of the POP reduce the computational burden associated with Equation 30, however the  $\text{POP}_{min}$  approximation is considered to be the best, since it does not bring unexpected results for dominated individuals.

## 5.4 Coupling the optimization framework with PuMA

When considering the problem formulated in Section 4.3, we can see the parallel with the optimization framework vocabulary. In principle, we have two objective functions to be analysed, being them  $f_1 = |\mu(k_{ii}) - \mu^{ref}(k_{ii})|$  and  $f_2 = |\text{Var}(k_{ii}) - \eta(k_{ii})|$ , where  $i$  stands the subscript used to indicate the directions  $x$ ,  $y$  and  $z$ , and  $\eta$  stands for a user-defined threshold. No constraints are to be considered in this problem.

Furthermore, changing the  $r_{vx}$  value for each design is the same as stating that we are changing the designs in the design space  $\chi$ . In fact, a DoE  $a \in \chi$  is a collection of input information that must be provided for PuMA to run the simulations.

Two approaches are explored in the following sections, in order to apply the optimization framework on the study of the results obtained using PuMA. Firstly, we are considering a bi-objective minimization under uncertainty problem, but without the probabilistic approach. We are going to consider specific designs with the evaluated vectors being measured from two objective functions, without any constraints. Then, it is proposed to manually search for the Pareto front, thus giving a first estimation of the REV for a specific fiber geometry. As we will see, this is a simple approach but it requires a very large number of calculations, since we must test out all design inputs in order to correctly compare between multiple scenarios.

Afterwards, it is proposed to use an optimization algorithm named SAMATA, which will explore the optimization problem with a probabilistic framework, by considering that the same design input can generate multiple robustness and reliability measures  $\rho_f$  and  $\rho_g$ , which in principle will cause the Pareto front of the problem to be "blurred", as a single design input will generate many different evaluated vectors to be compared. Furthermore, the computations with this algorithm prioritize the most optimal designs, by using the POP metrics elaborated in Section 5.3, meaning that it is an efficient way to find the REV, by diminishing the needed computations and the number of designs considered.

## 6 Brute force search for the REV

At first, one could think that simply performing massive simulations on different sets of input parameters is sufficient for obtaining an appropriated estimation of the REV, since  $r_{vx}$  can assume only integer values and it is the only free parameter for artificially generating microstructures obtainable via X-ray microtomography scans using PuMA. However, this approach shows to be not as simple, and indeed it is very laborious, due to the elevated number of data points required to correctly determine the REV.

As means of demonstrating the results obtainable via this approach, this section is dedicated in showing the whole procedure, starting from the determination of the reference values for the thermal conductivities  $k_{xx}$ ,  $k_{yy}$  and  $k_{zz}$  and their respective variances, followed by the analysis of the results obtainable via X-ray micro-CT, where we fix the value of  $d_{vx}$  and compare the obtained results with the reference values.

### 6.1 Computing the reference values

#### 6.1.1 Input parameters

Before running any simulation, it is necessary to describe the version of PuMA used in this work, as well as the geometrical and computational input parameters chosen. Concerning the version used, the present work relies on a modified version of PuMA v2.1, that was made specifically for the von Karman Institute for Fluid Dynamics in order to run directly on the Linux terminal. This simpler version in C++ is able to compute only the thermal response simulations, and it is designed for research purposes, as it is easier to parallelize the computations. It uses the Explicit-jump finite difference solver, with periodic boundary conditions and the BiCGSTAB method for solving the linear systems. This setup will result in smallest time required for running simulations.

Now considering the input parameters, we have that no specific physical material was chosen and this work is supposed to be a proof of concept for future researchers to be based upon. Hence, the choice of geometrical parameters does not affect as much the qualitative analysis of the results. Nevertheless, as source of inspiration, the parameters were heavily based on the works [11, 5], which are based on fibrous porous materials with carbon fibers used for the solid phase.

Some input parameters such as  $r_{vx}$ ,  $d_{vx}$  and  $\ell_{vx}^{phy}$  depends on the analysis being conducted, while others are independent of the analysis, meaning that they are considered to be constant throughout the rest of the work. Thus, the following input parameters were chosen as constants:

- |  |  |
|--|--|
| • $R_{phy} = 5 \mu\text{m}$              | • $\Delta\theta_y = 90^\circ$                        |
| • $\sigma_{R_{phy}} = 0.625 \mu\text{m}$ | • $\Delta\theta_z = 20^\circ$                        |
| • $L_{phy} = 800 \mu\text{m}$            | • $\varepsilon = 0.89$                               |
| • $\sigma_{L_{phy}} = 500 \mu\text{m}$   | • tolerance = $1 \times 10^{-4}$                     |
| • $\Delta\theta_x = 90^\circ$            | • $k_{fiber} = 12 \text{ W}/(\text{m}\cdot\text{K})$ |

Where  $k_{fiber}$  is the intrinsic thermal conductivity of the fibers inside the microstructure. In a similar fashion,  $k_{void}$  will denote the thermal conductivity of the gas inserted in the void phase of the physical domain.

Further on, the tolerance in the input parameters is the accuracy of the numerical solver on PuMA. A smaller value of tolerance will result in more accurate results, however it will require more time. The default value of tolerance is set to  $1 \times 10^{-5}$ , so this choice has been made to reduce the time required for computations.

Another point to make is that the domain generated allows for fibers to intersect each other, which is more realistic, as a real porous microstructure can have this feature. If desired, the user can deactivate this feature, but it is not of our interest to do so.

All of these inputs can be chosen for other versions of PuMA, as means of comparison, since all the inputs made explicit here are used in any version of PuMA running on C++. However, up until the writing of this work, the python extension of PuMA v3.0 or above does not allow for fibers to have standard deviation inputs for length and radius.

Now, when it comes to the choice of  $r_{vx}$ ,  $d_{vx}$  and  $k_{void}$ , it is a matter of how to approach the calculation of the reference value and what gas to choose for the medium. At first, the choice of vacuum as medium was tested, since there was not enough data on this choice using PuMA and it could be interesting to check how PuMA responds to calculations with  $k_{void} = 0$ . Besides, having thermal conductivity data for a TPS in vacuum is highly interesting, since these types of materials are mostly used in space. However, several instability issues were experienced with PuMA.

While using PuMA with vacuum as medium, it was extremely hard to control all the data points gathered, because there were many instances where some results would be returned empty or simply would not return, meaning that there was no predictability on PuMA's behaviour.

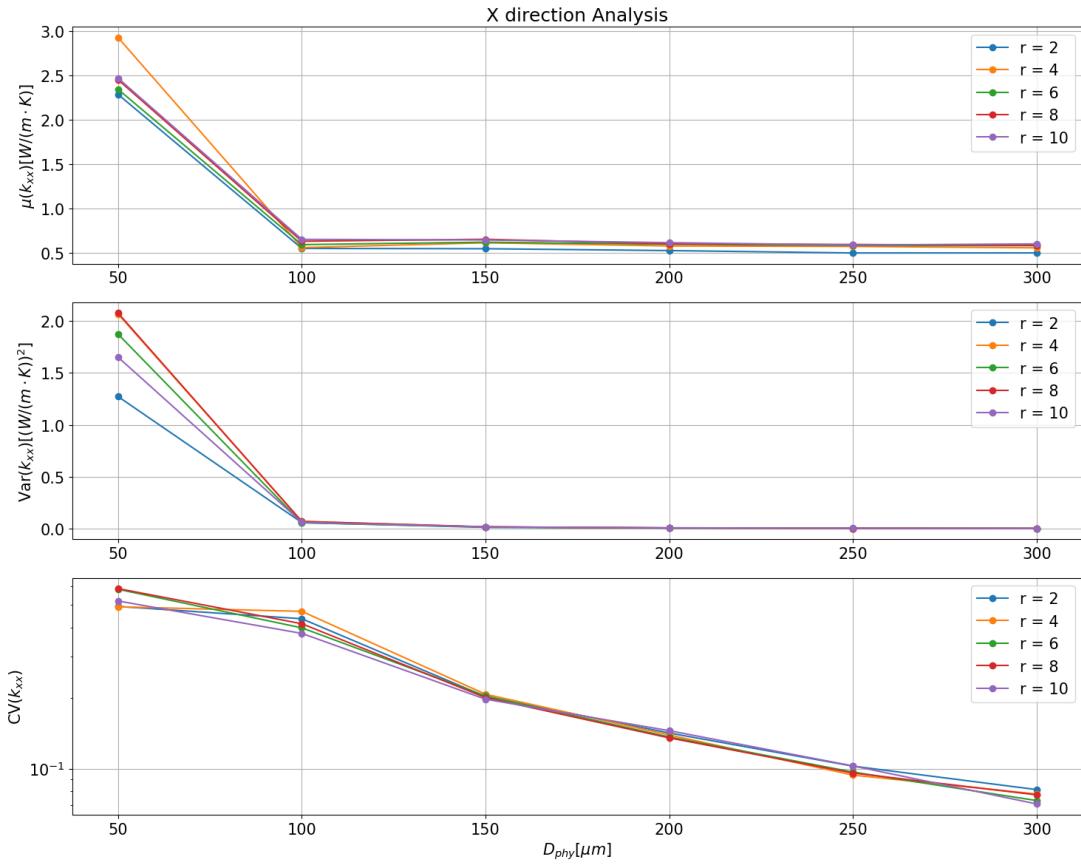
Therefore, knowing that the present work is a proof of concept, the value of  $k_{void} = 0.02587 \text{ W}/(\text{m}\cdot\text{K})$  has been chosen, as it is the value of thermal conductivity for air at a pressure of 1 bar, which is approximately equal to 1 atm, and a temperature of 20 °C. It is clear that the situations in which a TPS is tested are not the same as in this case, but in order to have a clear path of analysis to follow as guideline for further research, air in standard conditions has been chosen.

Considering the other inputs remaining, it is important to decide how to compare values in a manner that allows us to observe the influence of the refinement of the fiber on the final results of thermal conductivities. Thus, we must test different values of  $r_{vx}$  in order to comprehend the evolution of the thermal conductivities as functions of  $D_{phy}$ , meaning that  $d_{vx}$  must in fact be an input parameter that changes from simulation to simulation, depending on the combination of  $(r_{vx}, D_{phy})$  chosen.

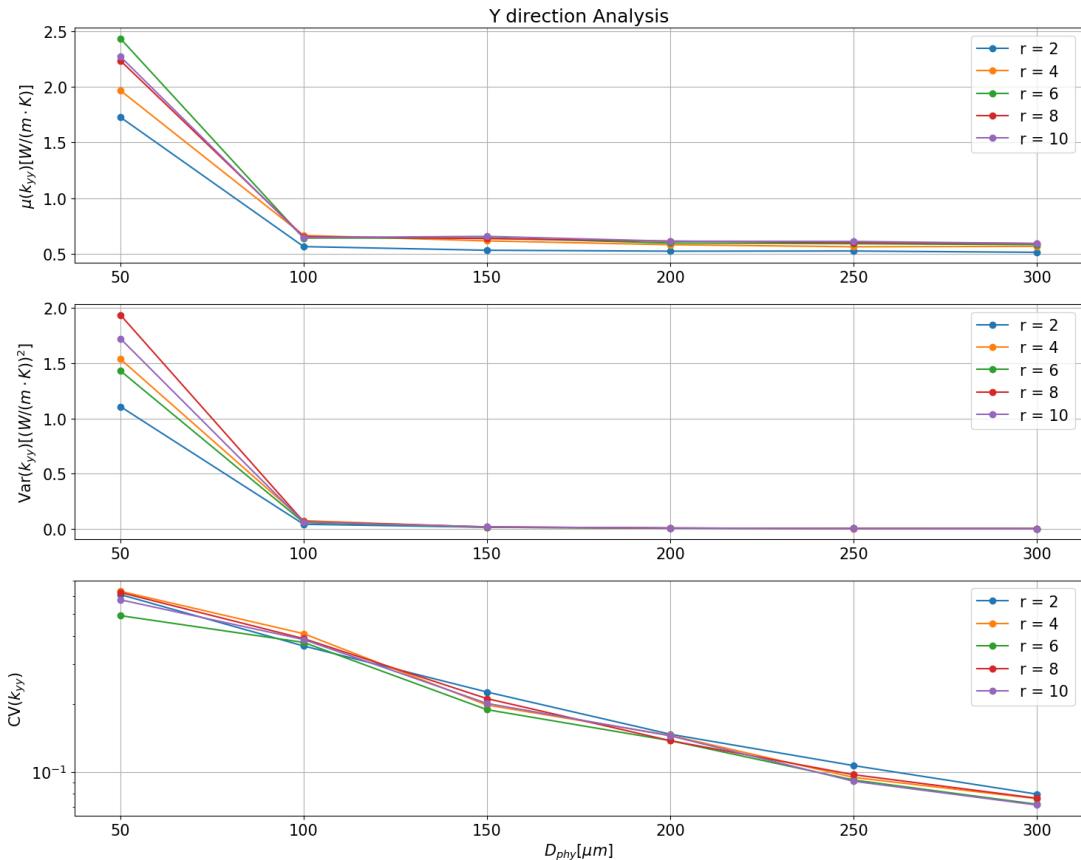
This way, by considering the values of  $r_{vx}$  tested by [11, 5], it has been decided that a comparison for  $r_{vx} \in \{2, 4, 6, 8, 10\}$  voxels had to be performed. In addition to this, we are going to compare the thermal conductivities for  $D_{phy} \in \{50, 100, 150, 200, 250, 300\} \mu\text{m}$ . The choice for  $D_{phy}^{max} = 300 \mu\text{m}$  has been motivated by the limitations encountered while using the PlaFRIM cluster, since for  $r_{vx} = (r_{vx})^{max} = 10$  we find  $(\ell_{vx}^{phy})^{min} = 0.5 \mu\text{m}/\text{voxel}$ , meaning that we have  $(d_{vx})^{max} = (D_{phy})^{max} / (\ell_{vx}^{phy})^{min} = 600$  voxels. When trying to reach higher values of  $D_{phy}$ , the results started to behave in a similar manner as when vacuum was considered as medium, meaning that this behaviour is due to limitations in computational power. With more computational power, the same analysis conducted here could be performed, and it would definitely produce results with higher accuracy, as we could pick larger physical domains.

### 6.1.2 Results obtained

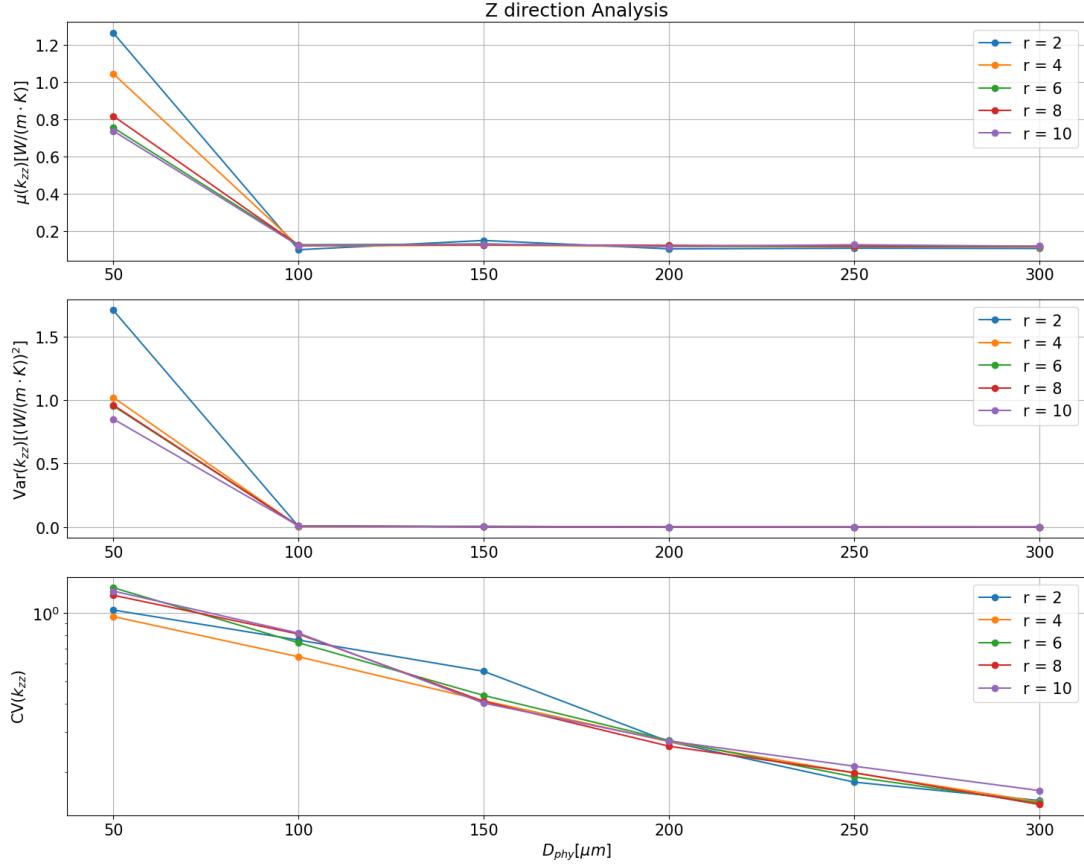
By using a script that runs in loop for all different configuration pairs of  $(r_{vx}, D_{phy})$  inside the list of possible combinations, it is possible to run PuMA as many times as we want per configuration, in order to build the data set. This way, 1000 simulation results were obtained for each configuration possible and the results are displayed in the following figure:



(a) Thermal response simulation results in the z direction for determining the value of  $\mu^{ref}(k_{xx})$ , using 1000 data points per configuration ( $r_{vx}, D_{phy}$ ).



(b) Thermal response simulation results in the z direction for determining the value of  $\mu^{ref}(k_{yy})$ , using 1000 data points per configuration ( $r_{vx}, D_{phy}$ ).



(c) Thermal response simulation results in the  $z$  direction for determining the value of  $\mu^{ref}(k_{zz})$ , using 1000 data points per configuration  $(r_{vx}, D_{phy})$ .

Figure 9: Thermal response simulation results for determining the reference values for thermal conductivities in all directions, using 1000 points per configuration  $(r_{vx}, D_{phy})$ .

In these results, we denote the arithmetic mean value of the thermal conductivity data in  $x$ ,  $y$  and  $z$  as  $\mu(k_{xx})$ ,  $\mu(k_{yy})$  and  $\mu(k_{zz})$ , respectively. In a similar manner, we denote the variance of the thermal conductivities calculated for each combination  $(r_{vx}, D_{phy})$  as  $\text{Var}(k_{xx})$ ,  $\text{Var}(k_{yy})$  and  $\text{Var}(k_{zz})$  for the  $x$ ,  $y$  and  $z$  direction, respectively. In this study, the variance is being calculated with a delta degree of freedom equal to 1, meaning that it is an unbiased estimator of the variance.

As  $D_{phy}$  increases, the thermal conductivity results follow an asymptotic behaviour, converging towards an estimation for the bulk value for the thermal conductivity that is different depending on  $r_{vx}$ . Meanwhile, every curve of variance converges towards zero, which was in accordance with what was said in Section 4.3, meaning that the thermal conductivities can converge to different values, even though their variances converge to zero, meaning that we could be converging to the wrong values of  $\mu^{ref}$ .

Moreover, Figure 9 presents curves of *Coefficient of Variation* (CV) for each direction, which is the ratio between the standard deviation and the mean value of a sample. Using the fact that the variance of a sample is equal to the standard deviation squared, we can calculate the CV for each combination  $(r_{vx}, D_{phy})$  using the data already collected..

The reason for the CV curves to have their vertical axes in logarithmic scale is due to the fact that the CV demonstrated an exponential decreasing while using linear scale for both axes. Therefore, one natural thought is that the CV and the values of  $D_{phy}$  are linearly correlated using this scale. In order to check this, we can make use of the correlation coefficient  $\text{Corr}(\cdot, \cdot)$ , which is a measure of how linearly correlated are two different measures.

Therefore, we must assume that

$$\text{CV}(k_{ii}) = a_i 10^{b_i D_{phy}} \Leftrightarrow \log_{10}(\text{CV}(k_{ii})) = \log(a_i) + b_i D_{phy}$$

In this equation,  $a$  and  $b$  are constants and the subscript  $i$  is used to indicate the direction, being therefore a mean of generalizing for the  $x$ ,  $y$  and  $z$  directions. It is important to mention that the CV is not only dependent on the direction of the thermal conductivity, but on the value of  $r_{vx}$  as well, due to the fact that increased values of  $r_{vx}$  provides us more accurate results, as the fibers are better rendered in the digitized domain.

By considering this, the values for correlation between  $\log_{10}(\text{CV}(k_{ii}))$  are displayed in Table 1, where not only we have all the directions, but all the possible values of  $r_{vx}$  being evaluated.

	Direction		
	<b>x</b>	<b>y</b>	<b>z</b>
$r_{vx}$	<b>2</b> -0.98142217	-0.99362371	-0.98770723
	<b>4</b> -0.97731947	-0.98686717	-0.99683969
	<b>6</b> -0.99083435	-0.99028229	-0.99208809
	<b>8</b> -0.98553687	-0.98974813	-0.98888798
	<b>10</b> -0.99406855	-0.99254697	-0.98164612

Table 1: Correlation coefficients  $\text{Corr}(\log_{10}(\text{CV}(k_{ii})), D_{phy})$  for all values of  $r_{vx}$ , being  $i$  the subscript to indicate the direction, being them  $x$ ,  $y$  and  $z$ .

As we can see, the correlation is strong, meaning that a linear regression would give a good prediction of CV values that were not captured by Figure 9. However, using the same figure we can see that the CV values still need to evolve in order to achieve lower values, meaning that it would be best to have samples using higher values of  $D_{phy}$ . If desired, it is possible predict at what value of  $D_{phy}$  we obtain a good enough CV value by using a linear regression, in order to increase the accuracy of the reference values obtained.

Considering that the higher the value of  $D_{phy}$  the best are the results obtained, then it is natural to choose  $D_{phy} = 300 \mu\text{m}$ . Additionally, as  $r_{vx}$  increases, we ought to have more accurate fibers, meaning that we are going to use the thermal conductivities and respective variances calculated at  $(D_{phy}, r_{vx}) = (300 \mu\text{m}, 10 \text{ voxels})$  as reference values of our study. Although  $\text{CV}(k_{zz})$  was higher for  $r_{vx} = 10$  voxels than for the other values, we have that  $\text{CV}(k_{xx})$  and  $\text{CV}(k_{yy})$  are the lowest when  $r_{vx} = 10$  voxels, and since in theory we must pick the higher value possible for  $r_{vx}$  and  $D_{phy}$ , then this result could be a consequence of different sources of error during the sampling.

By considering this, then we have the following:

$$\begin{aligned}
 \mu^{ref}(k_{xx}) &= 0.604182 \text{ W}/(\text{m}\cdot\text{K}) \\
 \mu^{ref}(k_{yy}) &= 0.593596 \text{ W}/(\text{m}\cdot\text{K}) \\
 \mu^{ref}(k_{zz}) &= 0.117677 \text{ W}/(\text{m}\cdot\text{K}) \\
 \text{Var}^{ref}(k_{xx}) &= 1.821063 \times 10^{-3} (\text{W}/(\text{m}\cdot\text{K}))^2 \\
 \text{Var}^{ref}(k_{yy}) &= 1.789927 \times 10^{-3} (\text{W}/(\text{m}\cdot\text{K}))^2 \\
 \text{Var}^{ref}(k_{zz}) &= 3.835413 \times 10^{-4} (\text{W}/(\text{m}\cdot\text{K}))^2
 \end{aligned} \tag{33}$$

Note that the notation for reference variance has changed, but in principle it is equal to what was stated in section 4.3, meaning that  $\text{Var}^{ref}(k_{ii}) = \eta(k_{ii})$ , where  $i$  indicates the direction  $x$ ,  $y$  or  $z$ .

For the rest of the study, these are going to be considered the reference values for thermal conductivities and variances, and *they will be considered as being constants*.

As the simulation results were gathered, it was observed that when  $D_{phy}$  and  $r_{vx}$  reached high values, there was not enough computational power to compute a large number of data points at once, meaning that the results had to be split in smaller groups. In addition, the version of PuMA used in this study had issues when generating random domains, meaning that many of the results obtained were exactly equal. In order to correct this, more data points should have been considered for each pair  $(r_{vx}, D_{phy})$ , and it would be advisable to use a more recent version of PuMA in a future work.

The higher the value of  $D_{phy}$  and  $r_{vx}$ , the lower will be the theoretical coefficient of variation, meaning that we would expect to find  $CV(k_{zz})$  getting lower and lower the higher the computational power available to perform computations and the number of data points taken. Definitely a better reference value could be computed if there was the possibility to compute the thermal conductivities for greater values of  $r_{vx}$  and  $D_{phy}$ .

In brief, it is not only a matter of number of samples, but also the computational power involved in the computations. With more power, more refined the digitized domains can be, and by consequence the thermal conductivity results would get closer to the bulk values, while the reference variance values would get closer to zero.

## 6.2 Considering the X-ray microtomography limitations

Now, we must generate samples with PuMA while considering the limitations imposed by real X-ray microtomography machines. Firstly, we must define the minimal voxel length that a microtomography machine is able to achieve, which we have called  $\ell_{min}$ .

When referring to conventional X-ray micro-CT machines, used in the medical or industrial sectors, generally there is a submillimeter range for the voxel length, meaning that we usually have voxel length  $\geq 100 \mu\text{m}$ . In contrast, a microtomography machine can present a micrometre range for the voxel length, meaning that we can have voxel length  $\geq 0.1 \mu\text{m}$  [30]. However, most commercially available machines microtomography machines present a voxel length between 1 and  $10 \mu\text{m}$  [31], while the most recent generation of microtomography machines present voxel length values as small as  $0.5 \mu\text{m}$  [32].

Thus, for the proceeding of the study, we are going to consider  $\ell_{min} = 0.75 \mu\text{m}/\text{voxel}$ , meaning that for  $R_{phy} = 5 \mu\text{m}$  we find  $r_{max} = 7$  voxels, wen rounding up the calculation. Therefore, considering the results obtained in the search for the REV, only the ones for  $r_{vx} \leq 7$  are considered to be obtainable via X-ray micro-CT scan.

### 6.2.1 Input parameters

For the samples obtained in Section 6.2, the inputs that were considered as constants in Section 6.1 are still considered constants, meaning that we need to discuss only the choice of  $r_{vx}$  and  $d_{vx}$  to run the simulations on PuMA. Here, we are considering that  $d_{vx} = \text{constant}$ , so it would be sufficient to choose any value  $d_{vx} \leq d_{vx}^{max} = 600$  voxels, but as means of comparison we are going to use three different values of  $d_{vx}$ , which will be considered as the values obtained from three different X-ray microtomography machines. Hence, we are choosing  $d_{vx} \in \{200, 250, 300\}$  voxels, meaning that the most powerful machine considered can only generate  $d_{vx} = (d_{vx})^{max}/2$ .

In reality, since this work is a proof of concept, there is no need to be alarmed by the values chosen, since the qualitative analysis remains valid. The principle here is to use results that would be obtained by X-ray micro-CT scans with less computational power than the cluster used to calculate the reference values. Finally, the values of  $r_{vx}$  will be the same as in section 6.1, meaning that  $r_{vx} \in \{2, 4, 6, 8, 10\}$  voxels.

In addition, the thermal conductivity of the void phase used is still  $k_{void} = 0.02587 \text{ W}/(\text{m}\cdot\text{K})$ , in order to have a valid comparison with the reference values computed beforehand.

### 6.2.2 Results obtained

By using another script that runs in loop for all different configuration pairs of  $(r_{vx}, d_{vx})$  inside the list of possible combinations, it is possible to run PuMA in order to build the data set for comparison with the reference values of thermal conductivities. The total number of data points per sample is still 1000 per configuration  $(r_{vx}, d_{vx})$ , and we are going to use the objective functions that were made explicit in the formulation of the problem, present in Section 4.3. Therefore, we denote the objective functions as being:

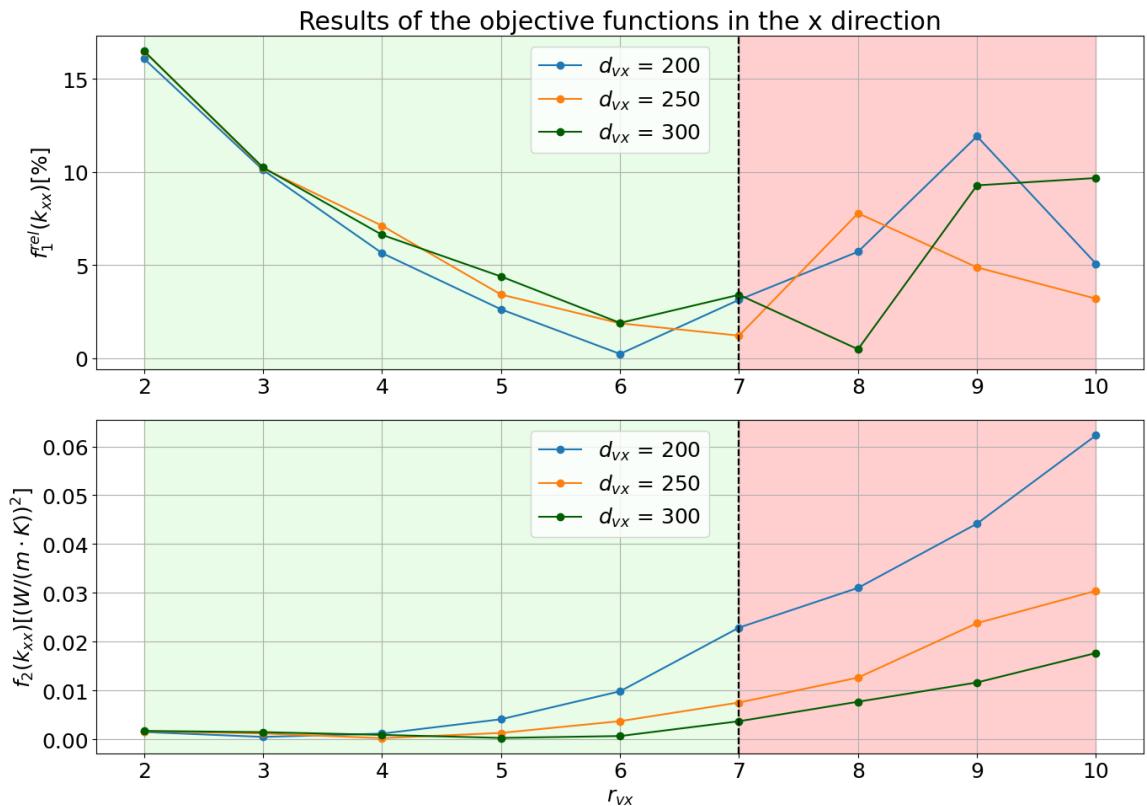
$$\begin{aligned} f_1(k_{ii}) &= |\mu(k_{ii}) - \mu^{ref}(k_{ii})| \\ f_2(k_{ii}) &= |\text{Var}(k_{ii}) - \text{Var}^{ref}(k_{ii})| \end{aligned} \quad (34)$$

Where  $i$  is denotes the directions  $x$ ,  $y$  or  $z$ . In order to visually compare the results obtained with the reference values, we are relying on the relative value for the objective function  $f_1$ , given by

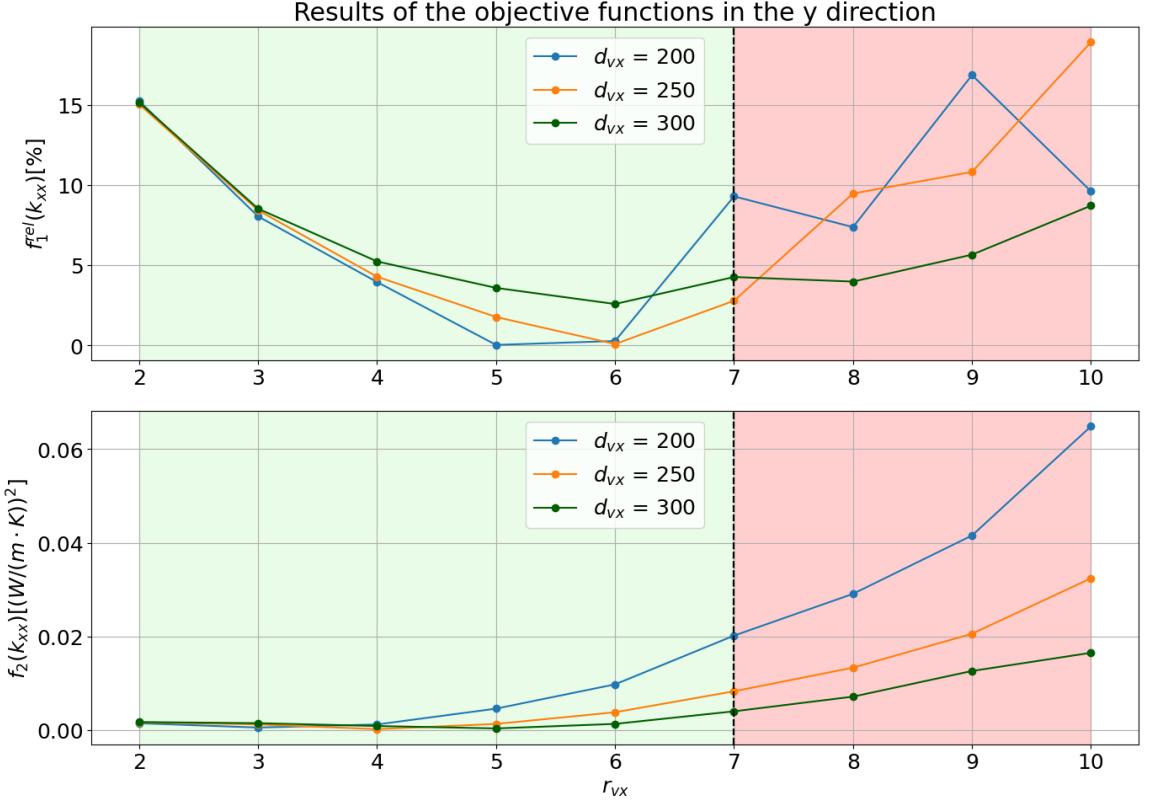
$$f_1^{rel}(k_{ii}) = \frac{|\mu(k_{ii}) - \mu^{ref}(k_{ii})|}{\mu^{ref}(k_{ii})} \quad (35)$$

Where  $i$  stands for the  $x$ ,  $y$  or  $z$  directions. Basically it expresses the relative error between the mean of the collected data points  $k_{ii}$  and the respective reference value in absolute terms. It is important to calculate this error in relative means in order to better understand how far are the results from the reference calculated.

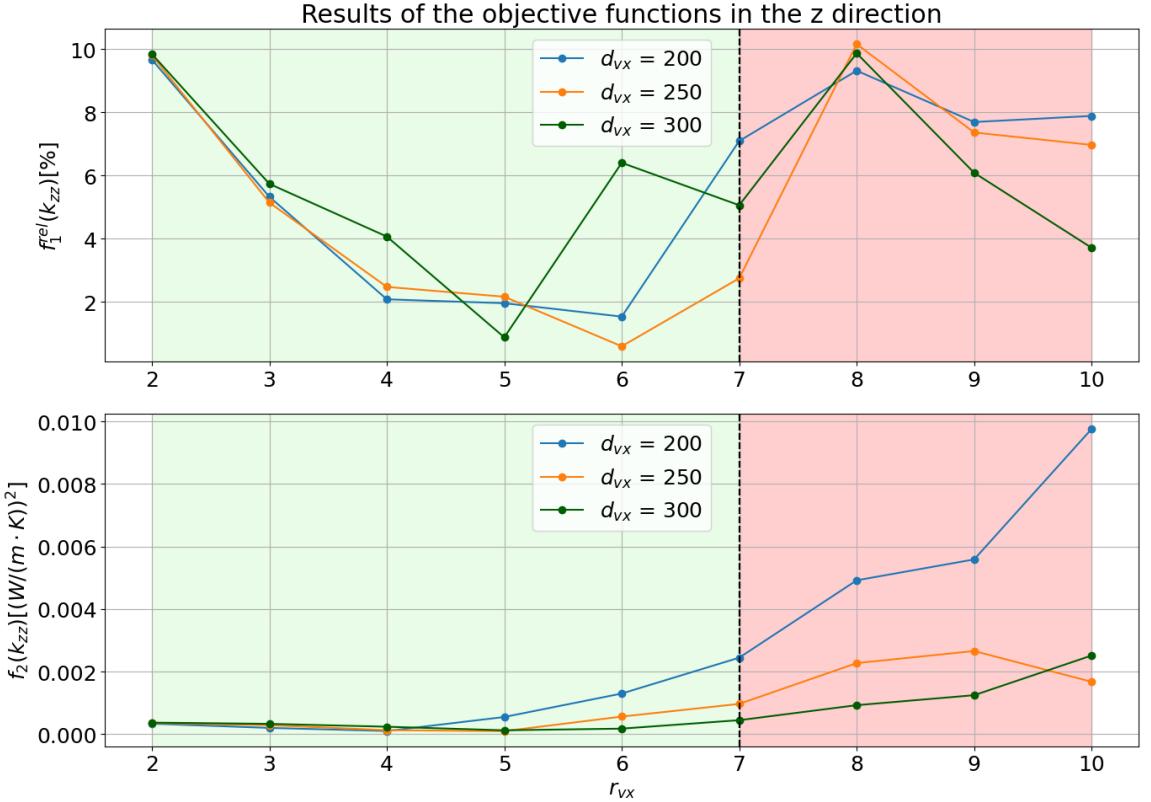
Figure 10 presents the results gathered for both objective functions for all configuration pairs  $(r_{vx}, d_{vx})$ .



(a) Objective functions results obtained in the  $x$  direction using 1000 samples per configuration  $(r_{vx}, d_{vx})$ .



(b) Objective functions results obtained in the y direction using 1000 samples per configuration ( $r_{vx}, d_{vx}$ ).



(c) Objective functions results obtained in the z direction using 1000 samples per configuration ( $r_{vx}, d_{vx}$ ).

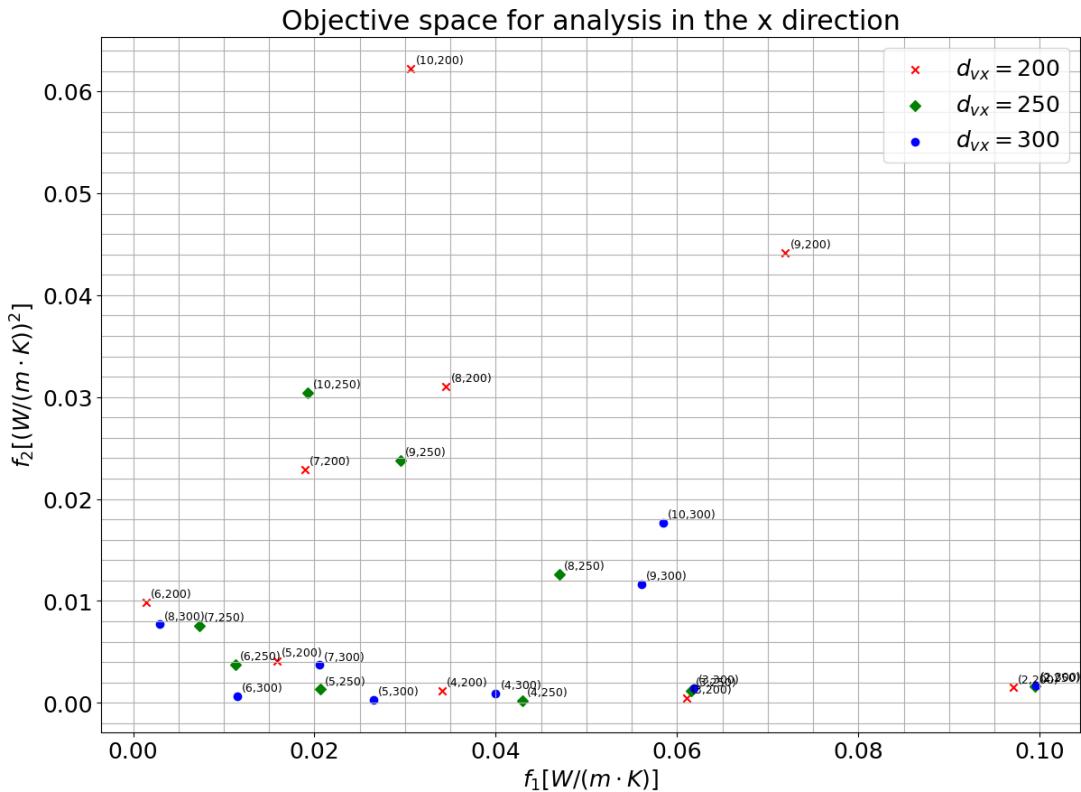
Figure 10: Objective functions results obtained in all directions, using 1000 points for each configuration pair ( $r_{vx}, d_{vx}$ ).

It is important to notice from Figure 10 that there are two zones in each direction plot, colored as green and red. The green zone indicates the results that are obtainable by considering  $\ell_{min} = 0.75 \mu\text{m}/\text{voxel}$ , and the red zone indicates the results that are not obtainable considering this limitation, but that were calculated using the PlaFRIM cluster computational resources as means of analysing the implications of using higher values of  $r_{vx}$ . The black dashed line simply indicates the separation between these two zones.

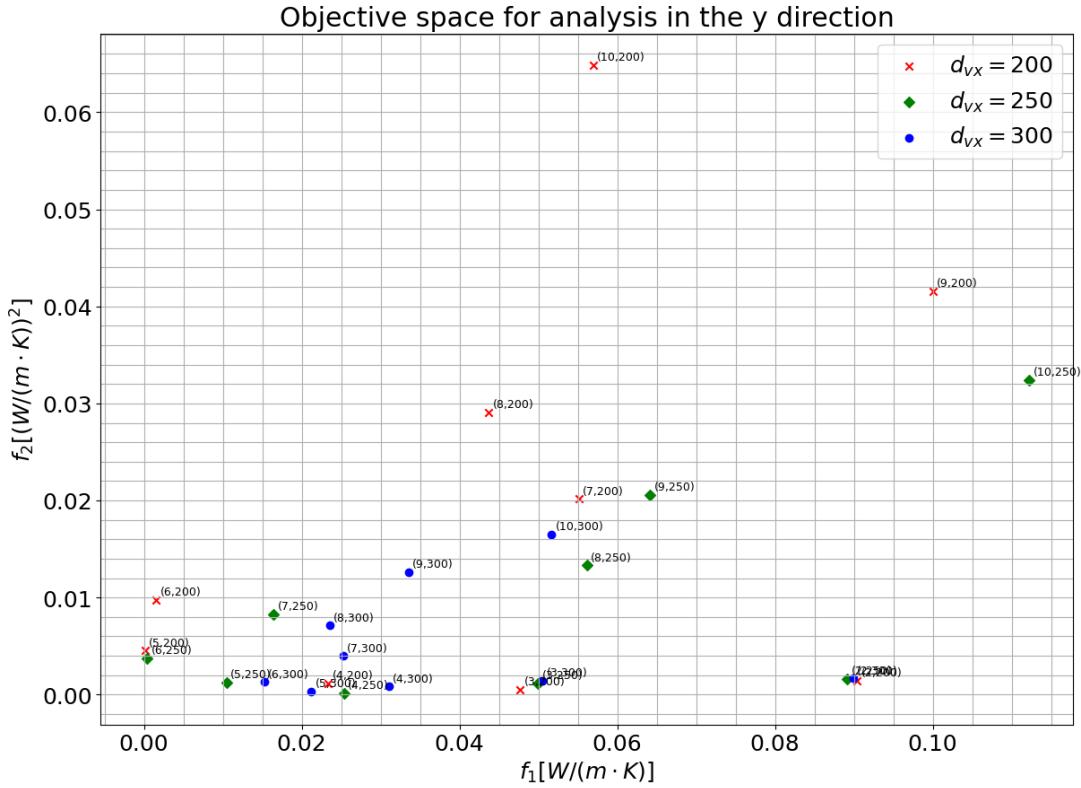
Considering the results obtained, it is clear that by fixing  $d_{vx}$  and increasing  $r_{vx}$ , the variance associated with each measurement increases, because we have that  $D_{phy}$  decreases, since  $D_{phy} \propto 1/r_{vx}$ . Additionally, as  $d_{vx}$  diminishes the variance increases even more, and this is due to a lack of voxels to correctly describe the microstructure.

From Figure 10 we can conclude as well that there is a point of minimal value for  $f_1$  for every curve. However, as  $r_{vx}$  increases,  $f_1$  starts to increase again, because the physical domain size is getting too small. This means that trying to reach for the highest resolution possible in the microtomography scan is not always the best option. In reality, it depends on the value of  $d_{vx}$ , which depends on the machine used.

The best approach is to use the optimization framework proposed in Section 5 and search for the Pareto front  $\mathbb{P}$  of this problem. Consider that all designs resides in the design space  $\chi$  and that all evaluated vectors are collected in the set  $X$ . Moreover, consider that all the evaluated vectors of the designs are named after the respective configuration pair  $(r_{vx}, d_{vx})$ , in order to identify the Pareto front  $\mathbb{P}(X)$ . In Figure 11 we can see all designs scattered in the objective space, using  $f_1$  in the horizontal axis and  $f_2$  in the vertical axis.



(a) Objective functions results obtained in the x direction using 1000 samples per configuration  $(r_{vx}, d_{vx})$ .



(b) Objective functions results obtained in the y direction using 1000 samples per configuration  $(r_{vx}, d_{vx})$ .

**Objective space for analysis in the z direction**

$f_1$	$f_2$	$d_{vx}$
0.005	0.005	(6,200)
0.005	0.005	(6,250)
0.005	0.007	(5,200)
0.005	0.007	(5,250)
0.005	0.008	(4,200)
0.005	0.008	(4,250)
0.005	0.009	(4,300)
0.005	0.010	(3,200)
0.005	0.010	(3,250)
0.005	0.011	(3,300)
0.005	0.012	(2,200)
0.005	0.012	(2,250)
0.005	0.013	(2,300)
0.005	0.014	(1,200)
0.005	0.014	(1,250)
0.005	0.015	(1,300)
0.005	0.016	(0.5,200)
0.005	0.016	(0.5,250)
0.005	0.017	(0.5,300)
0.005	0.018	(1,200)
0.005	0.018	(1,250)
0.005	0.019	(1,300)
0.005	0.020	(2,200)
0.005	0.020	(2,250)
0.005	0.021	(2,300)
0.005	0.022	(3,200)
0.005	0.022	(3,250)
0.005	0.023	(3,300)
0.005	0.024	(4,200)
0.005	0.024	(4,250)
0.005	0.025	(4,300)
0.005	0.026	(5,200)
0.005	0.026	(5,250)
0.005	0.027	(5,300)
0.005	0.028	(6,200)
0.005	0.028	(6,250)
0.005	0.029	(6,300)
0.005	0.030	(7,200)
0.005	0.030	(7,250)
0.005	0.031	(7,300)
0.005	0.032	(8,200)
0.005	0.032	(8,250)
0.005	0.033	(8,300)
0.005	0.034	(9,200)
0.005	0.034	(9,250)
0.005	0.035	(9,300)
0.005	0.036	(10,200)
0.01	0.001	(7,200)
0.01	0.002	(7,250)
0.01	0.003	(7,300)
0.01	0.	

Using the evaluated vectors in the objective space, we can obtain the Pareto front  $\mathbb{P}(X)$  for each direction analysis and for each value of  $d_{vx}$  used. In order to separate the results, consider that  $\mathbb{P}_x$ ,  $\mathbb{P}_y$  and  $\mathbb{P}_z$  stands for the Pareto front observed in the analysis of the  $x$ ,  $y$  and  $z$  directions, while all evaluated vectors are indicated only by the respective configuration pair  $(r_{vx}, d_{vx})$ . In order to present the results in a table, consider that we name the rows after the value of  $d_{vx}$ , meaning that we only need to give the correspondent values of  $r_{vx}$  to correctly identify the evaluated vectors in  $X$  for all designs in  $\chi$ . This way, the following table presents the Pareto-optimal values of  $r_{vx}$ :

Pareto front			
	$\mathbb{P}_x(X)$	$\mathbb{P}_y(X)$	$\mathbb{P}_z(X)$
$d_{vx}$	{3,4,5,6}	{3,4,5}	{4,5,6}
	{4,5,6,7}	{4,5,6}	{5,6}
	{5,6,8}	{5,6}	{5}

Table 2: Pareto fronts found for each direction analysis and separated between the  $d_{vx}$  values used. In order to fit the table in the page of this work, only the  $r_{vx}$  values are indicated for identifying each Pareto front.

From this table, we can roughly estimate the REV. But first, it is important to note that we have  $(r_{vx}, d_{vx}) = (8, 300)$  appearing in the Pareto front of the  $x$  direction, but since we must use  $r_{vx} \leq r_{max} = 7$  voxels, then this evaluated vector is dismissed.

For the estimation, we will fix a value of  $d_{vx}$  and a direction to analyse. From this, we must compute the range of  $D_{phy}$  values that corresponds to the  $r_{vx}$  values inside the Pareto front analysed, in order to have a maximum and minimum value for  $D_{phy}$  for each Pareto front. We are naming these maximum and minimum as  $D_{phy}^{max}$  and  $D_{phy}^{min}$ . After this, we must repeat the procedure for all other values of  $d_{vx}$ , in order to have all ranges of  $D_{phy}$  values for each Pareto front found in the direction analysed. After this, a way of approximating the final REV for the direction analysed would be to average out the limits of the ranges found for each  $d_{vx}$  value and then taking the arithmetic mean of these averages, in order to have a more general REV value, that would be a good approximation for any X-ray microtomography machine to use.

In order to illustrate this process, we are going to consider the process to find the REV in the  $x$  direction. Using the Pareto front  $\mathbb{P}_x(X)$  for every value of  $d_{vx}$  then we have  $D_{phy}|_{d_{vx}=200} \in [166.67, 333.33]$ ,  $\mu\text{m}$ ,  $D_{phy}|_{d_{vx}=250} \in [178.57, 312.50]$   $\mu\text{m}$  and  $D_{phy}|_{d_{vx}=300} \in [250.00, 300.00]$   $\mu\text{m}$ . Therefore, the average physical domain length for each  $d_{vx}$  is  $D_{phy}^{avg}|_{d_{vx}=200} = 250.00 \mu\text{m}$ ,  $D_{phy}^{avg}|_{d_{vx}=250} = 245.54 \mu\text{m}$  and  $D_{phy}^{avg}|_{d_{vx}=300} = 275.00 \mu\text{m}$ . By considering these averages on their own, we can see that they are somewhat close to each other. By taking the arithmetic mean between all these averages, we conclude that the optimal value  $(D_{phy}^{opt})_x$  for the physical domain length in the  $x$  direction analysis is equal to

$$(D_{phy}^{opt})_x = \frac{1}{3} \left[ D_{phy}^{avg}|_{d_{vx}=200} + D_{phy}^{avg}|_{d_{vx}=250} + D_{phy}^{avg}|_{d_{vx}=300} \right] \iff (D_{phy}^{opt})_x = 256.85 \mu\text{m}$$

By repeating this procedure in the  $y$  and  $z$  directions, we find the following results:

$$\begin{aligned} (D_{phy}^{opt})_x &= 256.85 \mu\text{m} \\ (D_{phy}^{opt})_y &= 267.36 \mu\text{m} \\ (D_{phy}^{opt})_z &= 245.83 \mu\text{m} \end{aligned} \tag{36}$$

Considering the results obtained, we can conclude a few points. Firstly, it has been shown only for the  $x$  direction that  $D_{phy}^{avg}$  has very close values between the results for all the  $d_{vx}$  value analysed, but this behaviour appears in the other directions as well. This means that the REV is not dependant on the machine used, but rather on the fiber's geometrical input parameters and the resolution of the domain.

Furthermore, the results obtained are close even when changing the direction analysed. In order to reduce the need to take multiple scans in a microtomography experimental campaign to analyse the different directions, one could simply average out all the  $D_{phy}^{opt}$  values found, in order to find an optimal  $D_{phy}$  in which good and reliable results would be found for all directions. By doing so, we find that

$$D_{phy}^{opt} = 256.52 \text{ } \mu\text{m} \quad (37)$$

### 6.3 Conclusion of the brute force search

This approach is the most simple approach and probably the one any researcher or experimentalist would try first. Indeed it provides good qualitative results for understanding the behaviour of PuMA and the computational power required for achieving a good performance.

As stated in Section 6.1, a higher computational power would be advisable for computing the thermal conductivities with higher values of  $D_{phy}$ , since it would best for the coefficient of variation to be smaller in all directions. Nevertheless, having a reference value with such small variation is good enough for evaluating the microtomography obtainable results.

Moreover, the results using the microtomography limitations made explicit that using the highest resolution possible is not the best approach. More often than not, having a higher voxel length value and a machine with less power can give good results.

In addition, since the evaluated vectors are not considered to be the results of a random field  $\mathbf{P}$  but rather a single vector for each design, then there is the probability that the Pareto fronts found are not indeed correct. The reason for this is that the points in the objective space should be rather a cloud of points for every design, and not only a single point, meaning that a probabilistic take on this problem will certainly provide more accurate results.

For instance, the Pareto front in the  $z$  direction found in Table 2 has only one evaluated vector, when actually the evaluated vector  $(r_{vx}, d_{vx}) = (6, 300)$  has almost made it into the respective Pareto front, meaning that due to uncertainties or errors with PuMA sampling we could be misled. If we were to consider the evaluated vectors as random vectors from a random field, thus generating clouds of points in the objective space, we could avoid misleading results.

In conclusion, there is the need to find an optimal value of  $r_{vx}$  to be used in the simulations, which by consequence implies in an optimal value for the voxel length selected by the experimentalist when performing an X-ray microtomography scan. By using this value, we were able to roughly estimate the optimal physical domain length for analysing the thermal response simulations on all directions. One could also average out all these results, in order to find an optimal value for running simulations on all directions, which would reduce the number of scans needed for a microtomography experimental campaign. Finally, this approach has led to a massive number of computations, so it is necessary to look for ways to reduce the number of calculations needed to estimate the REV.

## 7 Analysis conducted with the SAMATA algorithm

In this section, it is proposed to analyse the optimization problem formulated in Section 4.3 under the multi-objective optimization under uncertainty framework presented in Section 5.3. For such, we are going to present an optimization algorithm named SAMATA and solve the problem for the REV. What follows is a brief explanation that is heavily based on chapters 2 and 4 of [28].

### 7.1 Understanding the algorithm

The essence of the SAMATA algorithm is to tune the accuracy of each computation of the statistical measures  $\rho$  in order to give priority to the designs with highest POP measures. In principle, we compute the robustness and reliability measures with a few number of evaluations of the QoI vectors  $q$ , together with an estimate regarding their margin of error. The approximated measures are considered to be a random vector and we compare them and rank them using the POP metrics presented in Section 5.3. After this, the accuracy is tuned by improving the approximation error calculations only on the most promising designs. This leads to a major reduction in computation time, which is desirable. We name this approach the *Measure Approximation with Tunable Accuracy* (MATA).

Furthermore, we build a surrogate model on these robustness and reliability measure approximations in the design space, meaning that for any new DoE  $\alpha \in \chi$ , this surrogate will give predictions for the robustness and reliability measures, together with an estimation of the prediction error. We are denoting this as the *Surrogate-Assisting* (SA) model. In this case, when the prediction error is small enough, then we return the prediction to the optimization process, meaning that we don't have to estimate  $\rho$  through multiple evaluations of  $q$ .

The two methods coupled (SA and MATA) provide a very high efficiency, thus reducing the computational cost required to treat an optimization under uncertainty problem. In summary, the MATA technique permits to focus the computational power only on the most promising designs, while the SA strategy is a way to reduce the time required for computations, as it gives permission to bypass measure approximations and computations of the QoI  $q$  when we have already a good picture of the behaviour of  $\rho$ .

Furthermore, computations with this algorithm prioritize the most optimal designs, by using the POP metrics elaborated in this section, meaning that it is an efficient way to find the REV, by diminishing the needed computations and the number of designs considered.

The following figure presents a flowchart of the SAMATA algorithm. The notations presented in the figure are going to be discussed, in order to fully comprehend this algorithm.

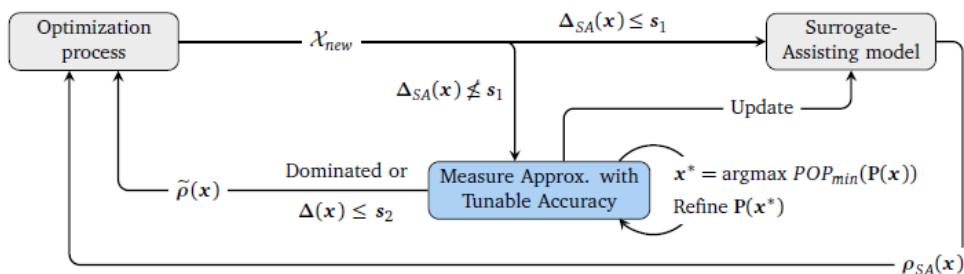


Figure 12: The SAMATA algorithm flowchart. Credits to [28].

For better comprehension of the algorithm, let us present the Surrogate-Assisting strategy and the Measure Approximation with Tunable Accuracy method used.

### 7.1.1 The Surrogate-Assisting strategy

Firstly, we consider the SA strategy. These types of strategies are very common practice to accelerate the optimization since we can bypass the high number of evaluations of  $q$  to obtain estimations of  $\rho$ . In this work, a surrogate model is built and constantly updated during the optimization process in order to predict the robustness and reliability measures  $\rho$  of the random evaluated vectors from the designs in  $\chi$ . In fact,  $\forall \mathbf{x}_i \in \chi$  that is visited during the optimization process, we consider that the evaluated vector  $\rho(\mathbf{x}_{\mathbf{x}_i})$  containing the robustness and reliability measures is an aleatoric approximation that resides in a random field  $\mathbf{P}$ , and we are going to name it  $\mathbf{P}(\mathbf{x}_i)$ . Then, the SA model considers the set  $\{(\mathbf{x}_i, \mathbf{P}(\mathbf{x}_i))\}$  as training set. When there is enough information collected, at a new design  $\mathbf{x}_{new}$ , the SA model may bypass the computation of the aleatoric approximation  $\mathbf{P}(\mathbf{x}_i)$  and returns directly the prediction  $\rho_{SA}(\mathbf{x}_{new})$ .

Considering the application of the algorithm, it is then necessary to make sure that the SA model is able to deal with aleatoric inputs, which can be considered to be affected by random noise in the input data. Indeed, the model should consider the noise for approximating its prediction error for each design. Thus, we make use here of heteroscedastic *Gaussian Processes* (GP).

To make it short, a GP is a non-parametric Bayesian regression technique used to model relationships between variables. It can be seen as a random process with Gaussian distribution on some training data, considering a priori a covariance function  $\mathbf{k}$  known. More information on GP can be seen in the first chapter of [28] and in the works [33, 34].

Here, it is proposed to consider the accuracy of the SA model in  $\mathbf{x}_{new}$  as a vector  $\Delta_{SA}(\mathbf{x}_{new})$ , so that  $\rho(\mathbf{x}_{new})$  gets redirected to the optimization process every time that  $\Delta_{SA}(\mathbf{x}_{new}) < s_1$ , where we consider  $s_1$  to be a threshold defined before running SAMATA. It is important to note that every time this condition is not met, then we must compute an approximation  $\mathbf{P}(\mathbf{x}_{new})$ , which requires a number of evaluations of  $q$ . Afterwards, the pair  $(\mathbf{x}_{new}, \mathbf{P}(\mathbf{x}_{new}))$  gets added to the training data, the surrogate is updated, and an estimated value  $\tilde{\rho}(\mathbf{x}_{new})$  gets returned to the optimizer. Usually the mean of  $\mathbf{P}(\mathbf{x}_{new})$  is used as  $\tilde{\rho}(\mathbf{x}_{new})$ .

It is clear that choosing smaller values of  $s_1$  results in an increased number of training data required before using the SA model. By consequence, this ensures a better accuracy of  $\rho_{SA}$ . However, it is important to look out if it is feasible to work with such small threshold, due to the increased computation time required to increase the training data. By using higher values of  $s_1$ , we can keep the number of evaluations of  $q$  low, meaning that we rely strongly on the SA model, since it gets called by the algorithm more times. For all these reasons, a loop using decreasing values of the threshold is used during the optimization to avoid bias introduced by an empirical choice of one threshold.

### 7.1.2 The Measure Approximation with Tunable Accuracy (MATA) technique

In this methodology, we are going to use the majority of the computational power available to compute the most promising designs. For such, we are going to use a second user-defined threshold, named  $s_2$ , that has the purpose of controlling the accuracy of these Pareto-optimal designs.

Let us denote by  $\chi_{new}$  and  $\chi_c$  the set containing the new designs to be explored by the optimization algorithm and the set of designs to compute, respectively. Furthermore, let us denote  $\chi_r \subseteq \chi_c$  the set containing designs that must be refined. The designs in  $\chi_r$  must follow two criteria:

1. They must belong to the Pareto front  $\mathbb{P}(\mathbf{P}(\chi_c))$
2. The threshold  $s_2$  has not yet been reached

Formally speaking, the definition of  $\chi_r$  is given as:

$$\chi_r = \{\mathbf{x} \in \chi_c \mid \Delta(\mathbf{x}) \leq s_2 \text{ and } \mathbf{P}(\mathbf{x}) \in \mathbb{P}(\mathbf{P}(\chi_c))\} \quad (38)$$

Where  $\Delta(\mathbf{x})$  is a chosen measure to describe the variability of  $\mathbf{P}(\mathbf{x})$ , being the standard deviation the most common measure used. Meanwhile,  $\mathbf{P}(\chi_c)$  corresponds to the set containing all images from the designs in  $\chi_c$ , meaning that  $\mathbf{P}(\chi_c) = \{\mathbf{P}(\mathbf{x})\}_{\mathbf{x} \in \chi_c}$ . It is important to highlight the fact that we must first compute the approximations  $\mathbf{P}$  and associated accuracy  $\Delta$ , and then assess the set  $\chi_r$ , in order to correctly define it after each optimization iteration.

Finally, it remains to determine which designs must have their probabilistic approximation  $\mathbf{P}$  refined. For this, it is proposed to use the POP concept that was introduced in Section 5.3, in a way that we can select the most promising designs to refine and concentrate our computational resources on them. Thus, the accuracy of the probabilistic evaluated vector  $\mathbf{P}$  is improved for the designs  $\mathbf{x}^*$  where we find the maximum values of  $\text{POP}_{min}$ . Formally, we have the following:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \chi_r}{\operatorname{argmax}} [\text{POP}_{min} (\mathbf{P}(\mathbf{x}))] \quad (39)$$

### 7.1.3 SA and MATA pseudo-code algorithms

The algorithms describing the functioning of the SA strategy and the MATA technique are illustrated in Algorithm 1 and Algorithm 2, respectively. Both algorithms were extracted from [28].

---

#### Algorithm 1: Algorithm of the SA strategy

---

```

1 Set initial threshold  $s_1$ 
2 Initialize the training set  $\mathbb{T} = \{(\mathbf{x}_{train_i}, \mathbf{P}(\mathbf{x}_{train_i}))\}_i$  empty
3 for  $i \in \llbracket 0, N_{threshold} \rrbracket$  do
4   Refine approximations in  $\mathbb{T}$  ▷Algorithm 2
5   Launch optimizer
6   while Optimization running do
7     Update the SA model on  $\mathbb{T}$ 
8     Get new designs  $\chi_{new}$  to visit
9      $\chi_{SA} = \{\mathbf{x} \in \chi_{new} \mid \Delta_{SA}(\mathbf{x}) \leq s_1\}$ 
10     $\chi_c = \chi_{new} \setminus \chi_{SA}$ 
11     $\forall \mathbf{x} \in \chi_{SA}$ , return  $\rho_{SA}(\mathbf{x})$  to the optimizer
12     $\forall \mathbf{x} \in \chi_c$ , compute  $\mathbf{P}(\mathbf{x})$  and return  $\tilde{\rho}(\mathbf{x})$  to the optimizer ▷Algorithm 2
13     $\forall \mathbf{x} \in \chi_c$ , add  $(\mathbf{x}, \mathbf{P}(\mathbf{x}))$  to  $\mathbb{T}$ 
14  Decrease  $s_1$ 
```

---

#### Algorithm 2: Algorithm of the MATA technique

---

```

1 Read threshold  $s_2$  and design set  $\chi_c$ 
2 Compute first approximations  $\mathbf{P}(\mathbf{x})$  or retrieve the ones that already exist
3 Compute the Pareto front  $\mathbb{P}(\mathbf{P}(\chi_c))$  with Equation 21
4 Compute  $\chi_r$  with Equation 38
5 while  $\chi_r$  is non-empty do
6   Find  $\mathbf{x}^*$  with Equation 39
7   Refine approximation  $\mathbf{P}(\mathbf{x}^*)$  with some evaluations of  $q$ 
8   Update  $\mathbb{P}(\mathbf{P}(\chi_c))$  and  $\chi_r$ 
9 return  $\mathbb{P}(\mathbf{x}) \forall \mathbf{x} \in \chi_c$ 
```

---

## 7.2 Adapting the algorithm

The SAMATA algorithm is a low-cost optimization under uncertainty algorithm, that considers the robustness and reliability measures as a random field  $\mathbf{P}$  over the design space  $\chi$ . Thus, in the following we are considering that it is possible to compute a number of realizations  $\mathbf{P}^{(i)}(\mathbf{x})$ ,  $\forall \mathbf{x} \in \chi$ , so that we can characterize  $\mathbf{P}(\mathbf{x})$ .

This way, we consider to take disjoint realizations, meaning that we draw realizations  $\mathbf{P}^{(i)}(\mathbf{x}_j)$  from the distribution  $\mathbf{P}(\mathbf{x}_j)$  at specific locations of interest, independently from the other ones. Hence, we must check the dominance relation between all pairs  $(i, j)$ , as depicted in the following equation:

$$P_{\mathbf{P}} [\mathbf{P}(\mathbf{x}_j) \not\succ_c \mathbf{P}(\mathbf{x}_k)] \approx \frac{1}{N_{samples}^2} \sum_{i=1}^{N_{samples}} \sum_{n=1}^{N_{samples}} \mathbb{1}_{\{\mathbf{P}^{(i)}(\mathbf{x}_j) \not\succ_c \mathbf{P}^{(n)}(\mathbf{x}_k)\}} \quad (40)$$

However, the computation of the POP comes with great computational cost when considering this formulation. Hence, it is proposed to use a Monte Carlo approximation, with a constant computational burden that can be picked to better adapt to the computational resources available. The Monte Carlo approximation is given in Equation 41

$$P_{\mathbf{P}} [\mathbf{P}(\mathbf{x}_j) \not\succ_c \mathbf{P}(\mathbf{x}_k)] \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \mathbb{1}_{\{\mathbf{P}^{(I_1^{(i)})}(\mathbf{x}_j) \not\succ_c \mathbf{P}^{(I_2^{(i)})}(\mathbf{x}_k)\}} \quad (41)$$

Where we consider to take realizations of  $I_1$  and  $I_2$ , where  $I_1, I_2 \sim \mathcal{U}([1, N_{samples}])$ , being  $\mathcal{U}([1, N_{samples}])$  the uniform distribution for picking a number between 1 and  $N_{samples}$ . Considering Equations 40 and 41, the  $POP_{min}$  metric is computed as

$$POP_{min} (\mathbf{P}(\mathbf{x}_i)) = \min_{\substack{\mathbf{x}_j \\ j \neq i}} (P_{\mathbf{P}} [\mathbf{P}(\mathbf{x}_j) \not\succ_c \mathbf{P}(\mathbf{x}_i)]) \quad (42)$$

Now, it is important to discuss the construction of the probabilistic approximation  $\mathbf{P}(\mathbf{x})$ , the refinement of the approximations of robustness and reliability measures and the technique used for constructing the SA model.

### 7.2.1 Random field surrogate based on a Kernel Density Estimation

At first, we are going to consider that we can construct the SA model by considering a Gaussian process. A prediction  $\mathbf{P}_{SA}(\mathbf{x})$  of  $\mathbf{P}$  must be constructed over the design space  $\chi$  by considering the empirical data collected, denoted by  $\mathbf{P}^{(i)}(\mathbf{x}_{SA})$ . In general, the GP can be described as

$$P_{SA}^{(i)}(\mathbf{x}_{SA}) = \mu_{SA}^{(i)}(\mathbf{x}_{SA}) + L_{SA} \mathbf{Y}^{(i)}, \quad \mathbf{Y} \sim \mathbf{N}(0, Id) \quad (43)$$

Where  $\mathbf{N}(0, Id)$  stands for a normal distribution with zero mean and the identity matrix  $Id$  as the covariance matrix. Furthermore,  $L_{SA}$  comes from a Cholesky decomposition performed on  $\Sigma_{SA}$ , where  $\Sigma_{SA}$  is the covariance matrix of the data in  $\mathbf{x}_{SA}$ , defined as the predictive covariance  $\Sigma_{GP}(\mathbf{x}_{SA})$ . Moreover,  $\mu_{SA}^{(i)}(\mathbf{x}_{SA})$  stands for the predictive mean. Further information can be read on chapter 4 of [28], but the main idea is that we consider that we can build an SA model out of a GP, and the specific notation and algorithm to compute it is not the main focus.

Now, we must formulate an SA model that approximates the true function  $\rho$  based on the random field  $\mathbf{P}$  computed at some design locations. Hence, it is proposed to construct a probabilistic SA model in which we compute random realizations of robustness and reliability measures, denoted by  $\mathbf{P}^{(i)}(\mathbf{x})$  considering a *Kernel Density Estimation* (KDE) strategy.

Here, the prediction  $\mathbf{P}_{SA}(\mathbf{x})$  of  $\mathbf{P}$  is going to be constructed over the design space  $\chi$  by considering the empirical data collected at specific training locations, denoted by  $\mathbf{P}^{(i)}(\mathbf{x}_{train})$ .

The training data denoted as  $\mathbf{P}^{(i)}(\mathbf{x}_{train})$  is considered to be a collection of sets of samples, where we can now compute a KDE distribution to describe it. Thus, we are going to construct the SA model in such a way that it agrees with the training KDE distribution at each training design, meaning that the KDE is a reference for the SA model.

Consider a given set of samples  $\mathbf{P}^{(i)}(\mathbf{x}_{train})$  at a training design  $\mathbf{x}_{train}$  and a specific smoothing kernel function, then the KDE must provide an approximation of the continuous density  $\phi_{\mathbf{x}_{train}}$ . The smoothing kernel function allows the diffusion of discrete information locally and it provides a non-parametric approximation of the PDF of  $\mathbf{P}(\mathbf{x}_{train})$ . Thus, by considering an equally weighted KDE, we have the following:

$$\phi_{\mathbf{x}_{train}}^{KDE}(\rho) = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} \text{ker}\left(\rho, \mathbf{P}^{(i)}(\mathbf{x}_{train})\right) \quad (44)$$

In Equation 44 we have that  $\text{ker}$  is the previously mentioned kernel function, verifying  $\forall \rho', \int_{\mathbb{R}^n} \text{ker}(\rho, \rho') d\rho = 1$ . This means that  $\phi_{\mathbf{x}_{train}}^{KDE}(\rho)$  also sums to 1, where in this formulation  $\rho$  stands for a dummy variable.

For simplicity reasons, we are going consider the kernel function to be a non-correlated Gaussian distribution, meaning that we must determine the standard deviations. In this context, the standard deviations are referred as the kernel bandwidths  $\mathbf{h}$ . There is a wide variety of options when it comes to calibrating  $\mathbf{h}$ , but in this work we are going to use the Silverman's bandwidth [35] on normalized data, meaning that the values  $\rho$  are normalized to  $\bar{\rho}$ , which is given by:

$$\forall \mathbf{x}_{train}, \forall j, \bar{\rho}_j = \frac{\rho_j - \min_i \left( (\mathbf{P}_j^{(i)}(\mathbf{x}_{train})) \right)}{\max_i \left( (\mathbf{P}_j^{(i)}(\mathbf{x}_{train})) \right) - \min_i \left( (\mathbf{P}_j^{(i)}(\mathbf{x}_{train})) \right)} \quad (45)$$

In addition, we are going to denote  $\text{ker}_{\mathbf{h}}$  the Gaussian kernel of standard deviation  $\mathbf{h}$ , so that:

$$\text{ker}_{\mathbf{h}}(\rho, \rho') = \frac{1}{\mathbf{h}\sqrt{2\pi}} \exp\left(-\frac{\|\rho - \rho'\|_2^2}{2\mathbf{h}^2}\right) \quad (46)$$

The kernel function used in Equation 44 is in reality  $\text{ker}^*(\rho, \rho') = \text{ker}_{\mathbf{h}^*}(\bar{\rho}, \bar{\rho}')$ , where the selected standard deviation or kernel bandwidth  $\mathbf{h}^*$  comes from the Silverman's rule of thumb, which is given by:

$$\mathbf{h}^* = \left(\frac{4}{n(dim + 2)}\right)^{\frac{1}{dim+4}} \quad (47)$$

Note that we considered the normalized Gaussian distribution for computing  $\mathbf{h}^*$ , meaning that the standard deviations are considered to be 1, while  $n$  and  $dim$  in Equation 47 stands for the number of samples and the number of dimensions of the data collected, respectively. Other choices could be made to optimize the choice of the bandwidth, such as using the *Mean Integrated Squared Error* (MISE) or cross-validation methods, however the Silverman's rule of thumb for selecting the bandwidth has been chosen in order to avoid turning the approach too expensive, in computational costs. It is a good estimation when considering Gaussian distributions, and it demonstrates good performance.

### 7.2.2 Constructing the SA model

By considering the KDE technique explored in Section 7.2.1, it is possible to construct a set of non-parametric PDF estimations  $\phi_{\mathbf{x}_{train}}^{KDE}(\rho)$  at each training design  $\mathbf{x}_{train}$ , by using the disjoint

realizations  $\mathbf{P}^{(i)}(\mathbf{x}_{train})$ . It is therefore possible to build a general approximation of the random field  $\mathbf{P}$  over the complete design space  $\chi$ .

In this work, we are relying on the *Metropolis-Hastings* (MH) algorithm to apply the computed KDE densities on the realizations of a heteroscedastic GP, thereby achieving the desired smoothness for  $\rho$ .

The MH algorithm can accept or reject the realizations of this GP, depending on the accordance with the KDE densities. As way to quantify this, the probability of acceptance for going from realization  $\mathbf{P}^{(i)}$  to  $\mathbf{P}^{(j)}$  is given by:

$$\alpha = \frac{\phi_{target}^{(j)} \phi_{proposal}^{(i)}}{\phi_{target}^{(i)} \phi_{proposal}^{(j)}} \quad (48)$$

Note that the probability  $\alpha$  can be seen as the ration between  $p^{(j)} = \frac{\phi_{target}^{(j)}}{\phi_{proposal}^{(j)}}$  and  $p^{(i)}$ , where each  $p^{(j)}$  quantifies how interesting  $\mathbf{P}^{(j)}$  is.

The proposal distribution comes from the heteroscedastic GP, while the targed density refers to the computed KDE. Hence, we are going to say that  $\phi_{target}^{(i)} = \phi_{KDE}^{(i)}$ , where  $\phi_{KDE}^{(i)}$  is given by the following:

$$\phi_{KDE}^{(i)} = \prod_{j=1}^{N_{train}} \phi_{\mathbf{x}_{train_j}}^{KDE} \left( \mathbf{P}^{(i)}(\mathbf{x}_{train_j}) \right) \quad (49)$$

Meanwhile,  $\phi_{proposal}^{(i)} = \phi_{GP}^{(i)}$ , where  $\phi_{GP}^{(i)}$  is defined by the following:

$$\phi_{GP}^{(i)} = \phi_{Gauss} \left( \mathbf{P}^{(i)}(\chi_{train}), \mu_{GP}(\chi_{train}), \Sigma_{GP}(\chi_{train}) \right) \quad (50)$$

Where  $\chi_{train}$  denotes the set containing all designs  $\mathbf{x}_{train}$ . In addition,  $\phi_{Gauss}$  denotes the PDF of the multivariate normal distribution of the GP, which has been discretized on the training designs  $\mathbf{x}_{train}$ . We have that  $\mu$  stands for the mean, while  $\Sigma$  stands for the covariance matrix. Thus, it is writen as follows:

$$\phi_{Gauss}(\rho, \mu, \Sigma) = \frac{\exp \left( -\frac{1}{2} (\rho - \mu)^T \Sigma (\rho - \mu) \right)}{\sqrt{(2\pi)^{N_{train}} |\Sigma|}} \quad (51)$$

By using  $\phi_{GP}$  as proposal distribution, we end up penalizing smooth functions when  $\phi_{GP}$  is high, favoring thus the rougher functions with lower value of  $\phi_{GP}$ . Hence, it is proposed to replace the joint computation of  $\phi_{GP}$  with an uncorrelated one  $\phi'_{GP}$ , and this can be done by simply setting all the off-diagonal values of  $\Sigma_{GP}(\chi_{train})$  to zero in Equation 50, meaning that  $\Sigma_{GP}(\chi_{train})$  becomes a diagonal matrix.

In addition, we are going to tune  $\phi'_{GP}$  as the proposal PDF, to guarantee that the constructed probabilistic SA model relapses to a heteroscedastic GP when all KDE densities are Gaussian. This can be done by considering  $\alpha = 1$  when all KDE densities are Gaussian.

This means that each density corresponds to the Gaussian noise associated to the specific location used to construct the heteroscedastic GP. Therefore, we have the following:

$$\phi_{noise}^{(i)} = \phi_{Gauss} \left( \mathbf{P}^{(i)}(\chi_{train}), \mu_{noise}(\chi_{train}), \Sigma_{noise}(\chi_{train}) \right) \quad (52)$$

Where  $\mu_{noise}$  and  $\Sigma_{noise}$  are the mean and the diagonal covariance matrix containing all the input variances, respectively. Thus, by setting  $\phi_{proposal} = \phi_{noise}$ , we have

$$\alpha = \frac{\phi_{KDE}^{(j)} \phi_{noise}^{(i)}}{\phi_{KDE}^{(i)} \phi_{noise}^{(j)}} \quad (53)$$

Which verifies  $\phi_{KDE} = \phi_{noise} \implies \alpha = 1$ , meaning that we relapse back to the simple heteroscedastic GP, which was of our interest.

### 7.2.3 Adapting the SAMATA algorithm to a non-parametric uncertainties scenario

At last, we are going to present the final considerations for the SAMATA algorithm used to tackle the problem formulated in Section 4.3. We are dealing with non-parametric uncertainties, meaning that the values  $q(\mathbf{x}, \xi)$  are not directly accessible, but the individual realizations  $q(\mathbf{x}, \xi^{(i)})$  are available to be drawn, without any previous information available regarding  $\xi^{(i)}$ . Hence, we denote  $\mathbf{Q}(\mathbf{x})$  the random value that comes from the function  $q$  evaluated at a given design  $\mathbf{x}$ , and only samples  $\mathbf{Q}^{(i)}(\mathbf{x})$  are available to be drawn, at a given cost.

By using the KDE-based SA model and a technique for executing the sampling-based Measure Approximation (MA), we can adjust the SAMATA algorithm. The technique chosen is the bootstrap strategy for sampling robustness and reliability measures  $\rho$  and refine current approximations by computing additional samples.

By using the classic bootstrap scheme, we can estimate the distribution of a statistic without any previous information on the distribution. Given a set of samples  $\mathcal{Q}(\mathbf{x}) = \{\mathbf{Q}^{(i)}(\mathbf{x})\}_{i=1}^{N_{samples}}$  at the design  $\mathbf{x}$ , the robustness and reliability measures can be empirically estimated as  $\tilde{\rho}(\mathbf{x}) = H[\mathcal{Q}(\mathbf{x})]$ , where  $H$  stands for expectation, a variance or a quantile. In this work, we consider it to be the expectation, so we denote it as  $\tilde{\rho}(\mathbf{x}) = E[\mathcal{Q}(\mathbf{x})]$ .

The bootstrap method consists in re-sampling from the set  $\mathcal{Q}(\mathbf{x})$  with replacement. Each element from a given bootstrap re-sampling  $\mathcal{Q}_{bs}^{(i)}(\mathbf{x})$  is denoted as  $\mathbf{Q}^{(I)}(\mathbf{x}) \in \mathcal{Q}(\mathbf{x})$ , where the index  $I$  follows  $I \sim \mathcal{U}([1, N_{samples}])$ . We denote  $\{\mathbf{P}^{(i)}(\mathbf{x})\}_i = \left\{E\left[\mathcal{Q}_{bs}^{(i)}(\mathbf{x})\right]\right\}_i$  the set of statistics, and it can be considered as a collection of *Independent and Identically Distributed* (i.i.d) realizations of the robustness and reliability measures.

When adding a small deviation for each sampled value, we obtain the smoothed bootstrap, which allows to smooth out the estimated distribution  $\mathbf{P}$ . Numerically, this is equivalent to computing realizations from a KDE of  $\mathbf{Q}$  during the bootstrap re-sampling step. Hence, the values  $\mathbf{P}^{(i)}$  are defined as statistics calculated over a re-sampled set of computed values  $E\left[\mathcal{Q}_{smooth}^{(i)}(\mathbf{x})\right]$ , in which every element from the re-sampled set  $\mathcal{Q}_{smooth}^{(i)}(\mathbf{x})$  is in reality a realization from the KDE density  $\psi_{\mathbf{x}}^{KDE}$ , built on samples  $\mathcal{Q}(\mathbf{x})$ .

This way, the adjusted SAMATA algorithm is given in Algorithm 3, for performing optimization under uncertainty with non-parametric uncertainties. The algorithm was extracted from [28].

## 7.3 Coupling SAMATA with PuMA

Now, it is time to tackle the optimization problem formulated in Section 4.3. Under the light of the results obtained in Section 6, it is proposed to change the formulation in order to obtain a more general analysis. For now, we are going to consider that we are not limited to a specific value of  $d_{vx}$ , but rather that we are limited to a range of values for  $D_{phy}$  and  $\ell_{vx}^{phy}$ . The reason for doing so is that sampling from a range of real values instead of only integer values is more adequate when considering the SAMATA algorithm, so we can interpret this change in formulation as a change of perspective.

Now we are not limited to analysing three machines, as in Section 6, but rather any machine that can generate a value  $d_{vx}$  that is equal to the rounded value of  $D_{phy}/\ell_{vx}^{phy}$ , and by changing  $\ell_{vx}^{phy}$  freely, we are going to round up the value of  $r_{vx}$ , computed with  $R_{phy}/\ell_{vx}^{phy}$ . This change was done in order to observe if it was a better option to change the physical domain length

**Algorithm 3:** Adjusted SAMATA algorithm

---

```

1 Loop over values of  $s_1$  and  $s_2$ 
2 while Optimization running do
3   Get new designs  $\chi_{new}$  to visit
4   Initialize  $\chi_c$  empty
5    $\forall \mathbf{x} \in \chi_{new}$ , Compute  $\mathbf{P}_{SA}$  using a (GP)       $\triangleright$ Equation 43 as reference
6   Compute  $\chi_{SA}^0 = \{\mathbf{x} \in \chi_{new} \mid \Delta_{SA}(\mathbf{x}) \leq s_1\}$  and  $\chi_c^0 = \chi_{new} \setminus \chi_{SA}^0$ 
7   Compute  $\chi_r^0$  from  $\chi_c^0$  using Equation 38
8   count = 0
9   while  $\chi_r^{count} \neq \emptyset$  do
10    Compute  $\forall \mathbf{x} \in \chi_{new}$ ,  $POP_{min}(\mathbf{x})$  with respect to all designs using
        Equation 32 and Equation 41
11    Find  $\mathbf{x}^*$  from  $\chi_r^{count}$  using Equation 39
12    if  $\mathbf{x}^* \notin \chi_c$  then
13      Add  $\mathbf{x}^*$  to  $\chi_c$ 
14      Compute  $N_{first}$  realizations  $\mathbf{Q}^{(i)}(\mathbf{x}^*)$ 
15    else
16      Compute  $N_{ref}$  realizations  $\mathbf{Q}^{(i)}(\mathbf{x}^*)$ 
17    Draw bootstrap realizations  $\mathbf{P}^{(i)}(\mathbf{x}^*)$  as presented in Section 7.2.3
18     $\forall \mathbf{x} \in \chi_{new} \setminus \chi_c$ , update  $\mathbf{P}_{SA}$ 
19    Compute  $\chi_c^{count+1}$ ,  $\chi_{SA}^{count+1}$  and  $\chi_r^{count+1}$ 
20    count = count + 1
21  Return  $E[\mathbf{P}(\mathbf{x})]$  to the optimizer for all  $\mathbf{x} \in \chi_{new}$ 
22  Return all measure approximations  $\mathbf{P}(\mathbf{x})$  and their associated POP values

```

---

and voxel length combination, in order to get closer to the reference values 33, or if the optimal physical domain length  $D_{phy}$  was independent on the power of the machine analysed.

Since this work is a proof of concept, this analysis has been conducted only regarding the  $z$  direction, as it is considered the *insulating direction* in our study, and it is usually the most critical when it comes to TPS applications.

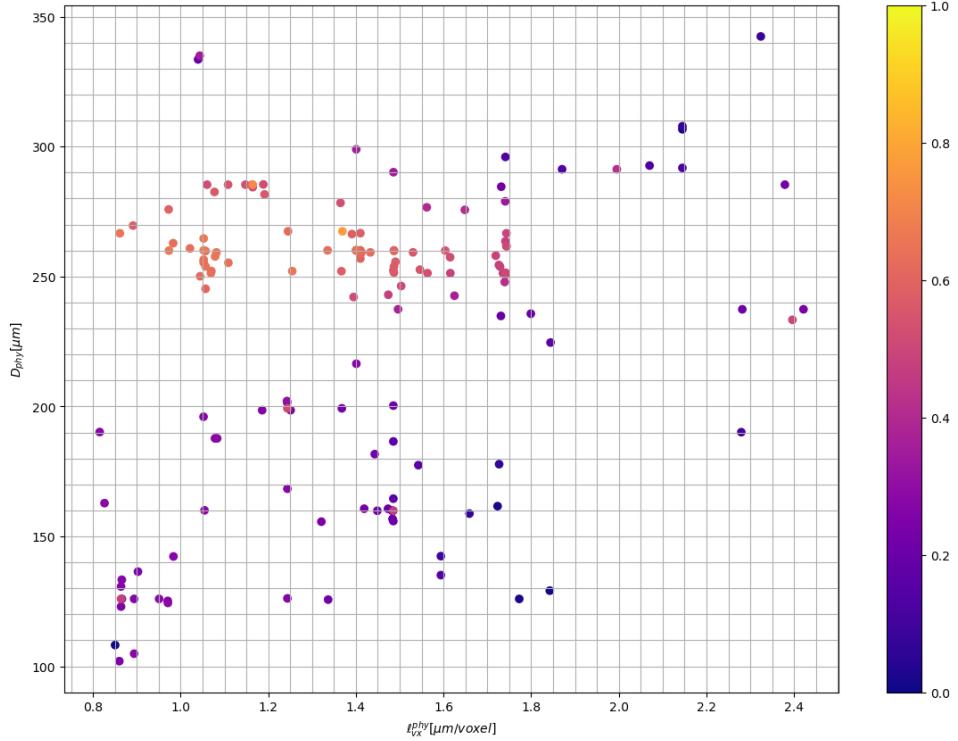
By considering many combinations of thresholds  $s_1$  and  $s_2$  for the SAMATA algorithm, it has been found a good computational time for running the code when  $s_1$  and  $s_2$  are equal and belong to the set  $\{1, 0.8, 0.6, 0.4\}$ . We reduce the values of the thresholds in order to see if we obtain good results even not relying so strongly on the SA model, and being more rigorous with the MATA section of the algorithm. For running the search of optimal designs, we are going to consider that  $(D_{phy}, \ell_{vx}^{phy}) \in [100, 350] \times [0.75, 2.5]$ , where the units of  $D_{phy}$  and  $\ell_{vx}^{phy}$  are in  $\mu\text{m}$  and  $\mu\text{m}/\text{voxel}$ , respectively. This way, we have  $r_{vx} \in [2, 7]$  and we can check even if it is better to have higher values of  $D_{phy}$  than the one used for capturing the reference values, but with lower resolution.

Thus, the new formulation of the problem is the following:

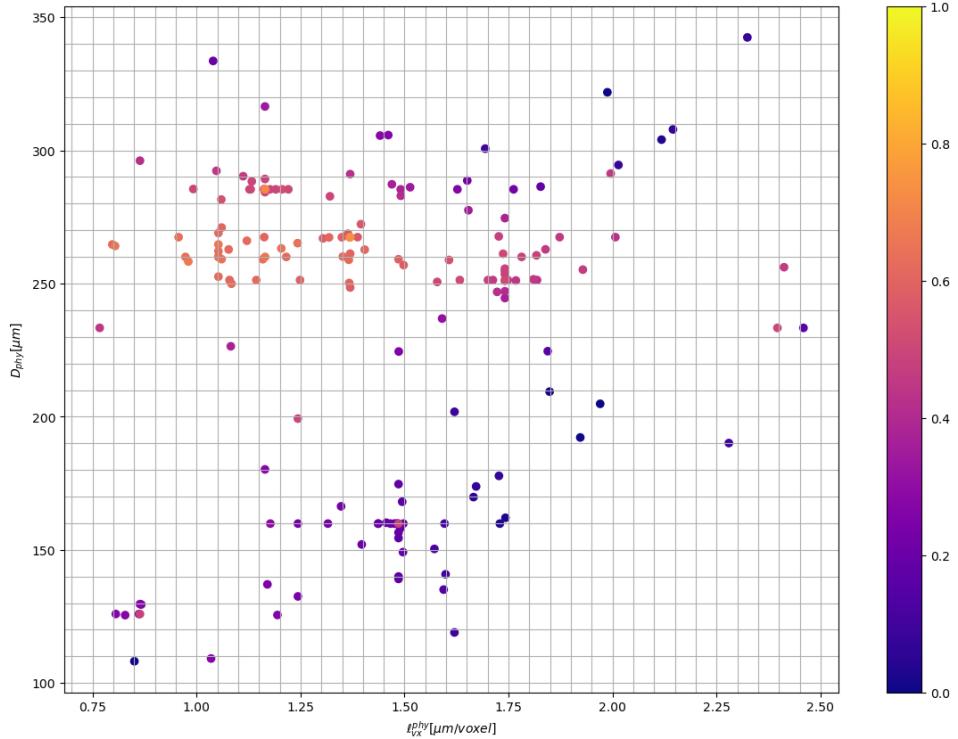
$$\begin{aligned}
& \text{minimize: } f_1(k_{zz}) = |\mu(k_{zz}) - \mu^{ref}(k_{zz})| \\
& \text{minimize: } f_2(k_{zz}) = |\text{Var}(k_{zz}) - \text{Var}^{ref}(k_{zz})| \\
& \text{by changing: } (D_{phy}, \ell_{vx}^{phy}) \in [100, 350] \times [0.75, 2.5]
\end{aligned} \tag{54}$$

Furthermore, concerning the bootstrap strategy of the adapted SAMATA algorithm, it was

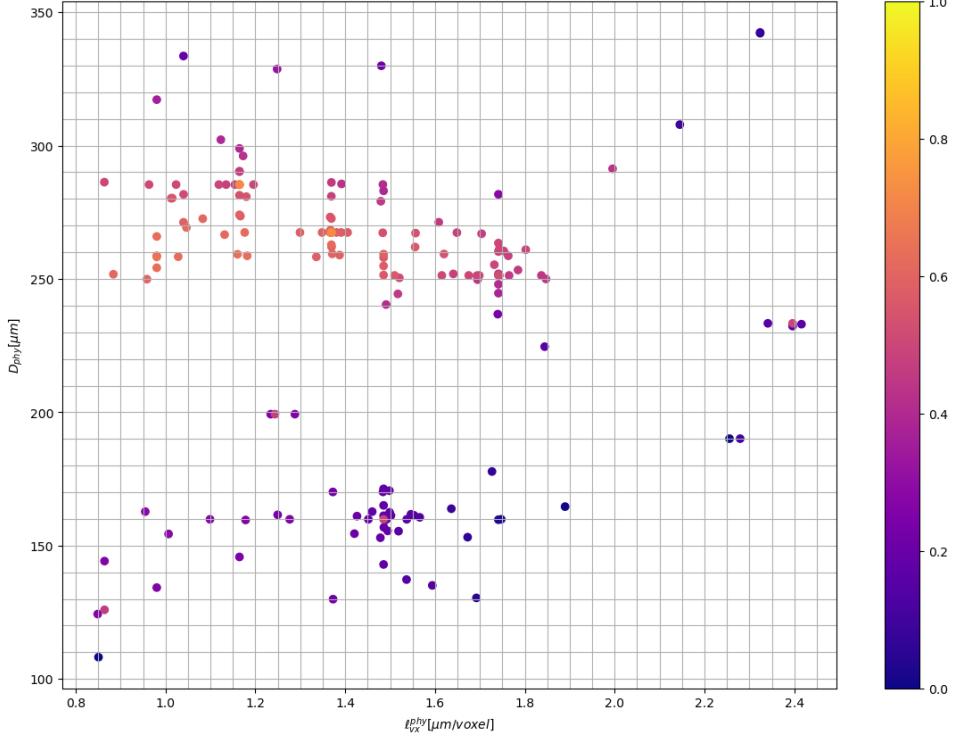
considered  $N_{first} = 20$  and  $N_{ref} = 10$ , while it was considered  $N_{samples} = 1000$  for the KDE procedure. After running the algorithm, the following results were gathered, illustrating the  $\text{POP}_{min}$  values for the various designs studied by the algorithm, using a color scale.



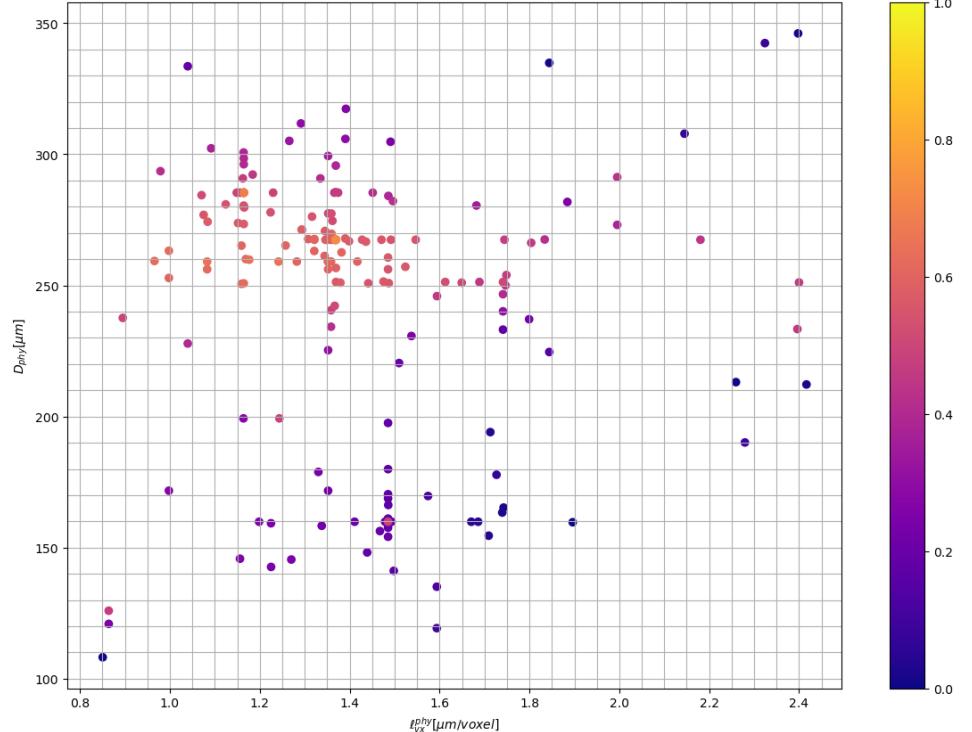
(a)  $\text{POP}_{min}$  values for  $s_1 = s_2 = 1$  for all designs  $(D_{phy}, \ell_{vx}^{phy})$  studied in the  $z$  direction.



(b)  $\text{POP}_{min}$  values for  $s_1 = s_2 = 0.8$  for all designs  $(D_{phy}, \ell_{vx}^{phy})$  studied in the  $z$  direction.



(c)  $\text{POP}_{\min}$  values for  $s_1 = s_2 = 0.6$  for all designs  $(D_{phy}, \ell_{vx}^{phy})$  studied in the  $z$  direction.



(d)  $\text{POP}_{\min}$  values for  $s_1 = s_2 = 0.4$  for all designs  $(D_{phy}, \ell_{vx}^{phy})$  studied in the  $z$  direction.

Figure 13: Comparison between  $\text{POP}_{\min}$  results obtained for various values of thresholds  $s_1$  and  $s_2$ , considering  $(D_{phy}, \ell_{vx}^{phy}) \in [100, 350] \times [0.75, 2.5]$ .

We can observe from these results that the designs with  $D_{phy} \in [250, 270] \mu\text{m}$  shows the highest values of  $\text{POP}_{min}$ , meaning that they are the most optimal designs, and we can observe that this optimal physical domain length extends for different resolutions. This means that for the various  $d_{vx}$  values found at each optimal design, we still find that the best designs fall into the same band of physical domain length, demonstrating that the REV does not depend on the power of the microtomography machine. This way, we can conclude that selecting the correct resolution to be able to capture a physical domain length that falls within this band in a given X-ray micro-CT machine is the best option for conduction a microtomography experimental campaign.

Furthermore, we can observe that the optimal physical domain length found in Section 6 is fairly close to fit the bandwidth of optimal results found in the current section. However, using a probabilistic framework to conduct the optimization was a better option due to the fact that we are dealing with an optimization under uncertainty problem, and we have managed to drastically reduce the number of computations necessary to reach conclusions. In the brute force search it was necessary to compute 1000 thermal conductivity values for each combination  $(r_{vx}, d_{vx})$ , summing up to a total of 27000 thermal response simulations to obtain responses in all directions, while using the SAMATA algorithm reduced it to 2186 simulations when analysing one direction exclusively. This number is definitely not a fixed number of required simulations, since we are considering a probabilistic framework, but the order of magnitude remains the same, and it highlights a reduction of approximately 90% in the number of simulations needed to analyse one direction. Assuming we would need 2000 simulations for each direction, we would still end up with 6000 simulations in total, which represents a reduction around 75%.

Finally, with this approach it was possible to assess the optimal physical domain length for microtomography machines with  $d_{vx}$  values ranging from 40 to 467 voxels, depending on the combination selected for  $(D_{phy}, \ell_{vx}^{phy})$ , with a substantial reduction of the computational cost.

## 8 Conclusion

This study dealt with the intricacies of porous materials and their characterization via X-ray microtomography. Considering the limitations imposed by real machinery, it is thus crucial to perform a robust optimization to determine the best setup to obtain reliable results, specially when characterizing materials to be used in TPS for space applications.

Due to the fact that the fibers in a porous material are oriented in a random manner, it was mandatory to formulate a mathematical problem of optimization under uncertainty and to treat it with the correct tools in order to find the REV. For doing so, the framework for determining reference values for the thermal conductivities in all directions had to be built in advance, in order to compare them with the results gathered using the X-ray microtomography machines limitations.

All microstructures were artificially generated using the PuMA software, and it has proven to be an excellent software for a microtomography experimental campaign, as it eliminates the necessity to perform multiple scans in real machines in order to get a grasp of what is the correct scale and resolution to use.

Throughout this study, we observed that the trade-off between resolution and physical domain size played a significant role in the determination of the most optimal results. Considering the brute force search for the REV, it has been found that the optimal physical domain length  $D_{phy}$  for all directions was approximately the same, and indeed we have determined that the Pareto-optimal designs were not the ones with highest resolution, which means that seeking for the highest resolution possible regardless of the physical domain size analysed is not the best approach for an experimental campaign.

Furthermore, similar results have been found for optimal physical domain size while performing the optimization analysis using the SAMATA algorithm, but with more reliability once we are considering a probabilistic framework instead of a deterministic one. We were able to reduce the number of computations and time required drastically by incorporating the optimization algorithm, which focused computational power on the most promising designs, eliminating the need to compute the thermal conductivities for all possible designs multiple times.

Indeed, by considering the probabilistic framework to tackle the optimization problem, we were able to determine that the optimal physical domain length  $D_{phy}$  lies within a not so wide band of values, independently on the  $d_{vx}$  value fixed, which means that the REV is not dependent on  $d_{vx}$ . In addition, by reducing the thresholds  $s_1$  and  $s_2$ , we were able to narrow the band of optimal results, at a cost of increased time required to complete computations.

Definitely the results obtained in this study are strongly dependant on the reference values computed in Section 6, and it is clear that with increased computational resources it is possible to compute better reference values, as we could reduce the voxel length and increase  $d_{vx}$  even further. The present work is a proof of concept, to show the results obtained by following the methodology used, which relies heavily on the UQ framework. Coupling the UQ tools with the PuMA software is a novel approach to improve experimental campaigns, and further research could be done by experimenting with a different optimization methodology or model and by using the most recent version of PuMA available.

## 9 References

- [1] Vasavada, Ashwin R: *Mission overview and scientific contributions from the mars science laboratory curiosity rover after eight years of surface operations.* Space Science Reviews, 218(3):14, 2022.
- [2] Witze, Alexandra: *A month on mars: what nasa's perseverance rover has found so far.* Nature, 591(7851):509–511, 2021.
- [3] Motion, Re entry: *Returning from space: Re-entry*, 2016.
- [4] Smith, Marcia S: *Nasa's space shuttle columbia: Synopsis of the report of the columbia accident investigation board.* Technical report, 2003.
- [5] Orero, J.B. Rico: *Computational study on the effect of the microstrucutre on macroscopic properties for carbon fiber felts.* Master's thesis, 2020.
- [6] Prabhakar, Subrahmanyam: *High-fidelity aerothermal engineering analysis for planetary probes using dotnet framework and olap cubes database.* International Journal of Aerospace Engineering, 2009, January 2009.
- [7] Arenas, Jorge and Malcolm Crocker: *Recent trends in porous sound-absorbing materials.* Sound vibration, 44:12–17, July 2010.
- [8] Baird, Emily and Gavin Taylor: *X-ray micro computed-tomography.* Current Biology, 27(8):R289–R291, 2017, ISSN 0960-9822. <https://www.sciencedirect.com/science/article/pii/S0960982217301318>.
- [9] Ferguson, Joseph C, Francesco Panerai, Arnaud Borner, and Nagi N Mansour: *Puma: The porous microstructure analysis software.* SoftwareX, 7:81–87, 2018.
- [10] Brune, Andrew J, Thomas K West IV, and Serhat Hosder: *Uncertainty quantification of planetary entry technologies.* Progress in Aerospace Sciences, 111:100574, 2019.
- [11] Girault, Florian and Pietro Marco Congedo: *Uncertainty Characterization of ablative materials for atmospheric reentry of spacecrafts using PuMA.* PhD thesis, Inria Saclay-Île de France, 2021.
- [12] Lorensen, William E and Harvey E Cline: *Marching cubes: A high resolution 3d surface construction algorithm.* ACM siggraph computer graphics, 21(4):163–169, 1987.
- [13] Long, Zhongjie and Kouki Nagamune: *A marching cubes algorithm: Application for three-dimensional surface reconstruction based on endoscope and optical fiber.* Information (Japan), 18:1425–1437, April 2015.
- [14] O'Sullivan, James D. B., Julia Behnson, Tobias Starborg, Andrew S. MacDonald, Alexander T. Phythian-Adams, Kathryn J. Else, Sheena M. Cruickshank, and Philip J. Withers: *X-ray micro-computed tomography (ct): an emerging opportunity in parasite imaging.* Parasitology, 145(7):848–854, 2018.
- [15] Ferguson, Joseph C, Federico Semeraro, John M Thornton, Francesco Panerai, Arnaud Borner, and Nagi N Mansour: *Update 3.0 to “puma: The porous microstructure analysis software”,(pii: S2352711018300281).* SoftwareX, 15:100775, 2021.
- [16] NASA: *Porous Microstructure Analysis (PuMA) software*, 2017. <https://github.com/nasa/puma>, visited on June 16, 2023.

- [17] Math2Market: *GeoDict*. <https://www.geodict.com/>, visited on June 16, 2023.
- [18] Synopsys: *Simpleware*. <https://www.synopsys.com/simpleware.html>, visited on June 16, 2023.
- [19] Cooper, Samuel J, Antonio Bertei, Paul R Shearing, JA Kilner, and Nigel P Brandon: *Taufactor: An open-source application for calculating tortuosity factors from tomographic data*. SoftwareX, 5:203–210, 2016.
- [20] Wiegmann, A and A Zemitis: *Ej-heat: A fast explicit jump harmonic averaging solver for the effective heat conductivity of composite materials*. 2006.
- [21] Wiegmann, Andreas and Kenneth P Bube: *The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions*. SIAM Journal on Numerical Analysis, 37(3):827–862, 2000.
- [22] Brigham, E Oran and RE Morrow: *The fast fourier transform*. IEEE spectrum, 4(12):63–70, 1967.
- [23] Mugler, DH and RA Scott: *Fast fourier transform method for partial differential equations, case study: The 2-d diffusion equation*. Computers & Mathematics with Applications, 16(3):221–228, 1988.
- [24] Vorst, Henk A Van der: *Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems*. SIAM Journal on scientific and Statistical Computing, 13(2):631–644, 1992.
- [25] Bergman, T.L., A.S. Lavine, and F.P. Incropera: *Fundamentals of Heat and Mass Transfer, 7th Edition*. John Wiley & Sons, Incorporated, 2011, ISBN 9781118137253. <https://books.google.com/books?id=5cgbAAAAQBAJ>.
- [26] Kolditz, Olaf: *Finite Volume Method*, pages 173–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, ISBN 978-3-662-04761-3. [https://doi.org/10.1007/978-3-662-04761-3\\_8](https://doi.org/10.1007/978-3-662-04761-3_8).
- [27] Liu, Renwei, Dongjie Wang, Xinyu Zhang, Wang Li, and Bo Yu: *Comparison study on the performances of finite volume method and finite difference method*. Journal of Applied Mathematics, 2013, 2013.
- [28] Rivier, Mickael: *Low-cost methods for constrained multi-objective optimization under uncertainty*. PhD thesis, Institut Polytechnique de Paris, 2020.
- [29] Khosravi, Faramarz, Michael Borst, and Jürgen Teich: *Probabilistic dominance in robust multi-objective optimization*. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–6. IEEE, 2018.
- [30] Withers, Philip J, Charles Bouman, Simone Carmignato, Veerle Cnudde, David Grimaldi, Charlotte K Hagen, Eric Maire, Marena Manley, Anton Du Plessis, and Stuart R Stock: *X-ray computed tomography*. Nature Reviews Methods Primers, 1(1):18, 2021.
- [31] Christiansen, Blaine A: *Effect of micro-computed tomography voxel size and segmentation method on trabecular bone microstructure measures in mice*. Bone reports, 5:136–140, 2016.
- [32] Tjong, Willy, Galateia J Kazakia, Andrew J Burghardt, and Sharmila Majumdar: *The effect of voxel size on high-resolution peripheral computed tomography measurements of trabecular and cortical bone microstructure*. Medical physics, 39(4):1893–1903, 2012.

- [33] Le, Quoc V, Alex J Smola, and Stéphane Canu: *Heteroscedastic gaussian process regression*. In *Proceedings of the 22nd international conference on Machine learning*, pages 489–496, 2005.
- [34] Schulz, Eric, Maarten Speekenbrink, and Andreas Krause: *A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions*. Journal of Mathematical Psychology, 85:1–16, 2018.
- [35] Silverman, Bernard W: *Density estimation for statistics and data analysis*. Routledge, 2018.
- [36] Ankenman, B., B.L. Nelson, and J. Staum: *Stochastic Kriging for Simulation Metamodeling*. Operations Research, 58(2):371–382, 2010.
- [37] Rasmussen, C.E. and C.K.I. Williams: *Gaussian Processes for Machine Learning*. the MIT Press, 2006, ISBN 026218253X.
- [38] Wang, W. and X. Chen: *The effects of estimation of heteroscedasticity on stochastic kriging*. 2016.