POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Statistical Analysis of Spherical Harmonics Representations of Soil Particles

GABRIEL LABERGE
Département de mathématiques et de génie Industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques Appliquées

Août 2020

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Statistical Analysis of Spherical Harmonics Representations of Soil Particles

présenté par **Gabriel LABERGE**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Richard LABIB, président
Serge PRUDHOMME, membre et directeur de recherche
Marc LAFOREST, membre et codirecteur de recherche
Bruno BLAIS, membre

DEDICATION

To everyone who pushed me
and helped me reach my full potential

ACKNOWLEDGEMENTS

First of all, I want to express my sincerest gratitude to my supervisors Prof. Serge Prudhomme and Prof. Marc Laforest. Their constant and constructive feedback allowed me to reach my full potential and to develop my rigorous thinking. Moreover, they knew when to let me loose, which sparked my inner curiosity and passion for research.

I am grateful to Dr. Varvara Roubtsova, Dr. Mohamed Chekired, and Dr. Paul Labbé from Institut de Recherche d'Hydro-Québec, who made the research possible by granting us access all the datasets of soil particles.

I also wish to thank all my co-workers from the Department of Mathematics and Industrial Engineering for insightful our conversations. Indeed, many technical mathematical details were made clearer through our long and heated debates.

Additionally, I am much obliged to my girlfriend Jennifer, not only for her unconditional love and support, but also for the time she generously spent helping me improve my writing abilities.

Finally, I must thank my whole family who were present through out my whole academic path. Everything I am today, I owe them.

RÉSUMÉ

Grâce aux avancées en micro-tomographie par rayons-X, il est désormais possible d'obtenir des représentations en 3D haute résolution de milliers de particules échantillonnées depuis diverses sources géologiques. La représentation plus précise des particules pourrait éventuellement permettre d'obtenir des simulations numériques plus fidèles des comportements de matériaux granulaires par la méthode des éléments discrets (DEM, Discrete Element Method en anglais). Cependant, l'accès à des descriptions fines demande aussi de développer de nouveaux outils numériques pour la caractérisation géométrique et l'analyse statistique d'ensembles de particules. Ce mémoire se concentre sur la modélisation géométrique des particules de sol par la représentation de leur surface à l'aide de la décomposition en harmoniques sphériques. Plus précisément, nous discutons de l'utilisation des représentations en harmoniques sphériques pour développer un modèle statistique permettant de générer des assemblages virtuels de particules à partir des données de plusieurs centaines de grains. La haute dimension de tels ensembles de données a longtemps été une complication majeure, mais avec les récentes avancées en apprentissage automatique dans l'analyse des mégadonnées, il y a espoir que ces nouveaux algorithmes puissent surmonter cette limitation.

ABSTRACT

Advancements in X-ray micro-computed tomography allow one to obtain high resolution 3D representations of particles collected from multiple geological sources. The representational power enabled by this new technology could allow for more accurate numerical simulations of granular materials using the celebrated *Discrete Element Method* (DEM). However, access to realistic representations of particles requires the development of more advanced geometrical and statistical characterization techniques. This thesis focuses on the use of the Spherical Harmonics decomposition of soil particles to model the surface of the particles. More precisely, we discuss the application of the Spherical Harmonics decomposition of particles to develop generative models of virtual assemblies that are calibrated based on datasets made of hundreds of grains. For long, the high dimensionality of the data has been a major challenge to the developpement of such statistical models. However, with recent advances of machine learning algorithms in the context of Big Data, there is hope that these new techniques can be utilized to overcome this limitation and obtain very accurate generative models of assemblies.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT vi

TABLE OF CONTENTS vii

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF SYMBOLS AND ACRONYMS xiv

LIST OF APPENDICES xvi

CHAPTER 1 INTRODUCTION 1
- 1.1 Scientific Context 1
- 1.2 Objectives 2
- 1.3 Literature Review 4
- 1.4 Contributions 6
- 1.5 Organization of the Thesis 8

CHAPTER 2 DISCRETE REPRESENTATIONS OF PARTICLES 9
- 2.1 Surface Triangular Mesh Representation (STM) 9
- 2.2 Spherical Harmonics Representation (SH) 12
  - 2.2.1 Fourier Series 12
  - 2.2.2 Spherical Harmonics 14
  - 2.2.3 Implementation 18
  - 2.2.4 Geometrical Interpretation 26
- 2.3 Preliminary Results 30
- 2.4 Summary 31

CHAPTER 3 GEOMETRICAL CHARACTERIZATION 35

3.1 Preliminary Results of Differential Geometry 35
3.1.1 Surface and Volume 35
3.1.2 Inertia and Semi-Axes 36
3.1.3 Curvature 37
3.2 Classical Shape Descriptors 40
3.2.1 Elongation and Flatness Indexes 40
3.2.2 Sphericity 41
3.2.3 Roundness 41
3.2.4 Convexity 42
3.3 Manufactured Star-Shaped Particles 42
3.3.1 Revolution Ellipsoid 43
3.3.2 Smooth Cubes 44
3.4 Asphalt Particle 48
3.5 Summary 48

CHAPTER 4 PROBABILISTIC MODELING 51
4.1 Data Samples and Storage 51
4.2 Principal Component Analysis 58
4.2.1 Affine Changes of Basis 58
4.2.2 Reconstruction Loss 60
4.2.3 Geometrical Intuition 61
4.3 Clustering 63
4.3.1 Silhouette 66
4.3.2 Cluster Compactness 67
4.4 Statistical Model 69
4.4.1 Simplification Hypotheses 69
4.4.2 Multivariate Gaussian 72
4.4.3 Kernel Density Estimation 73
4.5 Summary 78

CHAPTER 5 NUMERICAL RESULTS 80
5.1 Manufactured Particle Populations 80
5.1.1 PCA 80
5.1.2 Clustering 81
5.1.3 Statistical Model 84
5.2 Real Particle Population 86
5.2.1 PCA 88

5.2.2 Clustering 89
5.2.3 Statistical Model 92
5.3 Summary 99

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS 101
6.1 Summary of Work 101
6.2 Future Research 102

REFERENCES 104

APPENDICES 110

LIST OF TABLES

Table 2.1	Amount of usable and defective files for each population.	11
Table 4.1	Sampling distributions of ellipsoids.	53
Table 4.2	Sampling distributions of superquadrics.	53
Table 5.1	Cummulative variance of manufactured particles.	81
Table 5.2	Cummulative variance of real particles.	89
Table 5.3	Amount of rejected hypothesis of independence.	95

LIST OF FIGURES

Figure 1.1	Diagram of the main objective.	2
Figure 1.2	Diagram of the sub-objectives.	3
Figure 2.1	STM representation of a river particle.	10
Figure 2.2	Common defects in STM representations.	11
Figure 2.3	Extreme defect in the STM representations.	11
Figure 2.4	Example of the Gibbs phenomenon.	15
Figure 2.5	Spherical coordinates.	16
Figure 2.6	Slices of a star-shaped and non star-shaped particle.	16
Figure 2.7	Examples of Spherical Harmonics.	18
Figure 2.8	Interpolation of the grain surface.	19
Figure 2.9	Interpolation of two asphalt particles.	20
Figure 2.10	Gauss points distribution for various meshes.	22
Figure 2.11	Discretization errors of $\|Y_{15}^{0}\|^2$	22
Figure 2.12	Discretization errors of $\|Y_{20}^{10}\|^2$	23
Figure 2.13	Discretization errors of $\|Y_{10}^{10}\|^2$	23
Figure 2.14	Convergence of SH representations.	26
Figure 2.15	Energy decay for three particles.	27
Figure 2.16	SH modes as perturbations of the sphere.	28
Figure 2.17	Vizualization of the spherical harmonics.	29
Figure 2.18	Comparison of STM and SH representations.	33
Figure 2.19	More comparisons of STM and SH representations.	34
Figure 3.1	Principal frame of a paticle.	38
Figure 3.2	Example of parametric curve.	38
Figure 3.3	Example of tangent circle.	39
Figure 3.4	Curvature of a surface.	40
Figure 3.5	Four classes of ellipsoids.	41
Figure 3.6	Comparison of sphericity and roundness.	43
Figure 3.7	Convergence study on a revolution ellipsoid.	45
Figure 3.8	Parametrization of smooth cubes.	46
Figure 3.9	Convergence study on a smooth cube.	46
Figure 3.10	Curvature of a smooth cube.	47
Figure 3.11	Roundness of smooth cubes.	47
Figure 3.12	Asphalt particle under study.	49

Figure 3.13 Convergence study on a asphalt particle. 49
Figure 3.14 Largest inscribed sphere for STM and SH representations. . . 50
Figure 3.15 Curvature of an asphalt particle. 50
Figure 4.1 Diagram of the generative process. 52
Figure 4.2 STM representations of manufactured populations. 54
Figure 4.3 Additional STM representations of manufactured populations. 55
Figure 4.4 SH representation of manufactured populations. 56
Figure 4.5 SH representations of river particles. 57
Figure 4.6 Orthonormal change of basis in $\mathbb{R}^2$. 59
Figure 4.7 Example of PCA. 62
Figure 4.8 Nearest neighbors in SH coefficients space. 65
Figure 4.9 Unit balls in SH coefficients space. 68
Figure 4.10 Ideal clustering result. 71
Figure 4.11 Example non-multivariate Gaussian with Gaussian marginals. 73
Figure 4.12 Intuition behind KDE. 75
Figure 4.13 Selection of the bandwidth with cross-validation. 75
Figure 4.14 Intuition behind $\mathcal{M}$-KDE. 78
Figure 5.1 Mean of prolates. 82
Figure 5.2 Principal modes of prolates. 82
Figure 5.4 Selection of N_c for GM. 83
Figure 5.5 Clusters predicted by GM on manufactured particles. 84
Figure 5.6 Independence hypothesis on the box population. 85
Figure 5.7 Real and virtual boxes. 87
Figure 5.8 Real and virtual diamonds. 88
Figure 5.9 PCA mean and modes on the river population. 90
Figure 5.10 Selection of N_c for GM on river particles. 91
Figure 5.11 Clusters predicted by GM on river particles. 91
Figure 5.12 River particles from both clusters. 92
Figure 5.13 Shapes descriptors of the river particles from both clusters. . . 93
Figure 5.14 Virtual particles where the x_i are independent. 94
Figure 5.15 Independence test on river particles. 95
Figure 5.16 Bandwidth selection on river particles. 96
Figure 5.17 Exploration of the neighborhood of a river particle. 97
Figure 5.18 Comparison of KDE and $\mathcal{M}$-KDE. 99
Figure 5.19 Manifold hypohtesis applied on river particles. 99
Figure C.1 Independence test for a Uniform distribution. 117

Figure C.2 Independence test for linear relationship. 118
Figure C.3 Independence test for quadratic relationship. 118
Figure C.4 Independence test for independent Gaussians. 119
Figure C.5 Independence test for correlated Gaussians. 119
Figure C.6 Independence test for square-shaped distribution. 120
Figure D.1 Distance between two arbitrary river particles. 123
Figure D.2 Min/max distances from a query point to river particles. . . . 124

LIST OF SYMBOLS AND ACRONYMS

Notation	Definition
Chapter 2 page 9	
N_v	Number of vertices in triangulation of surface of grain
N_f	Number of faces in triangulation of surface of grain
$\mathbf{V}$	$N_v \times 3$ matrix of all nodes in STL file
$\mathbf{K}$	$N_f \times 3$ Connectivity matrix in STL file
$\mathbf{N}$	$N_f \times 3$ Matrix of the normals of every face in STL file
$\mathbf{A}(i,j)$	Indexing the ith row and jth column of the matrix $\mathbf{A}$
$L^2(\Omega)$	Hilbert space of squared integrable functions on domain Ω
$\langle \cdot\,, \cdot \rangle_\Omega$	Scalar product in $L^2(\Omega)$
$\|\cdot\|_\Omega^2$	Norm induced by the scalar product
θ	Azimuth angle from spherical coordinates $0 \leq \theta < 2\pi$
ϕ	Polar angle from spherical coordinates $0 \leq \phi \leq \pi$
$r(\theta, \phi)$	Radius of the particle in spherical coordinates
S^2	Unit sphere $S^2 = \{\boldsymbol{x} \in \mathbb{R}^3 : \ \|\boldsymbol{x}\| = 1\}$
$Y_\ell^m(\theta, \phi)$	Spherical harmonics
c_ℓ^m	Spherical harmonics coefficients
ℓ	First spherical harmonics index
m	Second spherical harmonics index
$\ell_{\max}$	Maximal value of ℓ used in SH representation
ℓ_0	Frequency at which the Lanczos sigma factor starts being applied
$E(\ell)$	Total energy of the frequency ℓ
$\mathcal{I}$	Linear interpolation operator of a particles surface
Ω_i	ith element of the mesh of the unit sphere
N_g	Number of Gauss points per mesh element
$\langle f \rangle$	Mean value of a function on the unit sphere $\langle f \rangle = \frac{1}{4\pi} \iint_{S^2} f(\theta, \phi) dS$
Chapter 3 page 35	
S	Surface Area
V	Volume
$\mathbf{I}$	Inertia tensor
$\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$	Principal axes

Notation	Definition
a, b, c	Ellipsoids semi-axes $a < b < c$
κ_1, κ_2	Principal curvatures $\kappa_1 < \kappa_2$
H	Mean Curvature
K	Gauss Curvature
EI	Elongation index
FI	Flatness index
Chapter 4 page 51	
d	Maximal number of SH coefficients considered $d = (\ell_{\max} + 1)^2$
N_s	Number of samples in a dataset
$c_i^{(j)}$	ith SH coefficient of the jth particle in a dataset
$\widehat{c}_i^{(j)}$	Normalized SH coefficients.
$\widehat{\mathbf{C}}$	$(d-1) \times N_s$ coefficient matrix, $\widehat{\mathbf{C}}(i,j) := \widehat{c}_i^{(j)}$
m	Number principal components selected by PCA ($m < d-1$)
$\boldsymbol{q}_i$	Data's ith principal axis $i = 1, 2, ...m$
$x_i^{(j)}$	ith principal components of the jth particle
$\mathbf{Q}$	$m \times N_s$ principal components matrix, $\mathbf{Q}(i,j) := x_i^{(j)}$
$X_k(\theta, \phi)$	New orthonormal perturbation of the sphere induced by PCA
$R(\mathbf{Q}, \boldsymbol{\mu})$	L_2 Reconstruction loss
$\widehat{w}_i^{(j)}$	Reconstruction of $\widehat{c}_i^{(j)}$
$\mathrm{CV}(i)$	Cummulative variance of to the ith principal component
$\|\cdot\|_F^2$	Frobenius norm of a matrix
$\|\cdot\|_2^2$	Euclidean norm
$\|\cdot\|_{\mathbf{A}}^2$	Mahalanobis norm
N_c	Number of clusters
K_i	ith cluster, $i = 1, 2, 3, ..., N_c$
$s(j)$	Silhouette of the jth particle
$\mathrm{I}(N_c)$	Average cluster inertia for N_c clusters
$\mathrm{BIC}(N_c)$	The Bayesian information criterion
$K(\cdot)$	Kernel function used in Kernel Density Estimation (KDE)
h	KDE bandwidth
$\mathbf{I}$	Identity matrix
$\mathbf{1}$	Column vector containing only ones
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Mutivariate Gaussian distribution
$n_k(j) \subset \{1, 2, \ldots, N_s\}$	List of the indices of the k nearest neighbors of the jth particle

LIST OF APPENDICES

Appendix A Proofs 110
Appendix B Formulas for Spherical Harmonics 112
Appendix C Correlation and Independence Tests 117
Appendix D Norms in $\ell^2(\mathbb{N})$ 121

CHAPTER 1 INTRODUCTION

1.1 Scientific Context

It is well understood that the macroscopic behavior of granular materials results from a cascade of phenomena that originates at the scale of the geometry of the particles. Many experimental studies show the effect of particle morphology on such characteristics as the packing density, stiffness, compressibility, or critical state [1–3]. The Discrete Element Method (DEM) [4], first introduced by Cundall and Strack in the 1980's, was developed with the prospect of numerically simulating granular materials by considering the interactions between all particles in an assembly. In today's day and age, the exponentially increasing power of computers has made it possible to simulate with DEM assemblies of thousands, and even millions of particles, and therefore one can envision simulating properties at the macroscopic scale. Recent numerical studies based on DEM have confirmed that there is a relationship between particle morphology and macroscopical behavior [5–8]. Yet, those studies have only identified partial relations between the means of certain particle characteristics and macroscopic behavior.

One of the major limitations of the DEM is the necessity to simplify the shape of the particles to reduce computational costs. Commonly used to approximate particles are spheres, ellipsoids, superquadrics, or polyhedrons. To obtain meaningful correlations between the shape of grains and DEM responses, the geometrical features used to characterize particle morphology must uniquely determine the particles. Simply put, the complexity of useful geometrical characterization tools is driven by the complexity of the shapes considered in DEM simulations. The widespread use of simplified shapes like spheres, ellipsoids and superquadrics has favored the classical shape descriptors as geometrical characterization tools i.e. the volume, aspect ratios, convexity, sphericity, convexity, and roundness [9]. Although these features are easily interpretable and completely determine spheres, ellipsoids, they are only partially correlated with DEM simulations.

The use of more complex geometries in DEM requires the development of more advanced geometrical characterization techniques. Originally, the geometrical features of particles were calculated using 2D images obtained by microscopy [10]. Though cheap, these methods do not provide a complete picture in 3D of the morphology of the grains. With the recent arrival of inexpensive X-ray micro-computed tomography (μCT), 3D representations of thousands of particles collected from various sources are now made available [11]. With access to this data, it is not only possible to compute classical shape descriptors more accurately [12], but

Figure 1.1 Overall generative process, as a black box, to generate an arbitrary large collection of virtual particles from a small sample of real particles.

it has also been possible to develop more powerful geometrical characterization tools that can allow one to use DEM for simulating real soils, rather than an idealization of soils.

1.2 Objectives

One of the new exciting applications enabled by μCT datasets of particles is the generation of *virtual* assemblies which share the same complex geometries as real particles collected from a given soil. The principal objective of this thesis shall be stated as:

> **Main Objective** *Develop a generative process to create random collections of virtual particles that are both realistic and geometrically similar to real particles collected from various geological sources. These particles shall be completely determined by their geometrical properties.*

The overall process, as illustrated in Figure 1.1 involves several requirements, which are now clarified. By virtual particles, it is meant that said particles must be generated by a computer algorithm and must not be a simple copy of a real particle. The randomness of the generating process is key to enable the construction of 10K-100K particles from soil samples of only 1K particles. Indeed, the available soil samples are usually incomplete and much smaller than what is required to run large scale DEM simulations. A probabilistic process can hopefully solve the issue of generating arbitrarily large collections of representative grains by filling out the gaps between the available particles.

Moreover, it is required that the virtual particles be realistic and geometrically similar to real particles. The notion of *realistic* particles is really subjective, although aberrant shapes can easily be detected by the naked eye. Being *geometrically similar* is a notion that is not trivial to formalize mathematically and will be studied thoroughly in this document.

The final requirement is that the virtual particles be determined by their geometrical characteristics. Basically, two particles that are obviously different cannot share the same geometrical characteristics as it would lead to some ambiguity when correlating geometry to DEM responses.

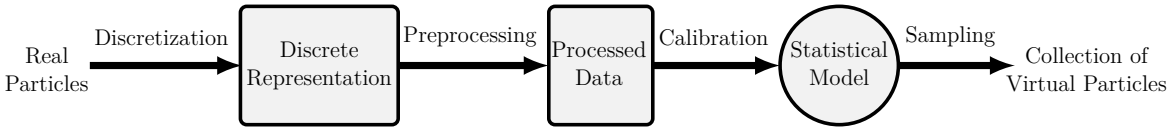


Figure 1.2 Sequence of steps to generate a collection of virtual particles from a sample of real particles.

The black box illustrated in Figure 1.1 can be decomposed into several steps which are shown in Figure 1.2. In the diagram, boxes represent stored data, the circle is a model and the arrows are the various mandatory steps. The first step is referred to as a *discretization*, which takes real particles collected from a soil, and compute their discrete representation in order to store them in a computer. The chosen discrete representation must satisfy two constraints: consistency and completeness. Consistency refers to the requirement that every particle should be represented using a fixed number of features that measure specific morphological quantities, i.e. the same finite-dimensional subspace. This enables meaningful comparisons between particles in a dataset as well as statistical analysis. Completeness implies that it is possible to reconstruct any particle to arbitrary precision by choosing sufficiently many features. This property ensures that the particles are uniquely defined by their geometrical features, which is one of the constraints of our main objective.

The following step of the generative process is to find representations of the data which allow for efficient statistical analysis. This is referred to as *preprocessing* in Figure 1.2. Such techniques can include normalization and dimensionality reduction for example.

Once the raw data has been transformed, the final step is to use it to calibrate a *statistical model*, shown as a circle in Figure 1.2. After being trained, the model is used to sample an arbitrarily large number of virtual particles. In this thesis, we shall mainly be interested in statistical models that come from the field of machine learning.

Once all steps of the generating process have been implemented, we shall use manufactured particles with simple geometries, i.e. ellipsoids and superquadrics, to apply validation tests of the various models. This is critical since the long term objective is to fully automate the full process described in Figure 1.2. To do so, we need an absolute confidence in every part of the black box. Manufactured particles are useful to validate models as their exact geometries and statistical distributions can be controlled.

1.3 Literature Review

Micro-computed tomography grants access to high-resolution 3D representations of real particles. These representations are usually discretized using voxels or surface meshes, which are useful for particle visualization using computer software. However, as explained earlier, we are looking for representations of particles that are consistent and complete. It is unlikely that all particles obtained by μCT would have the same number of voxels or nodes so these representations are not consistent. Even if some particles had the same number of nodes, consistency would still not be achieved since the ith node need not describe the same aspect of the particles across the dataset. A popular consistent representation of particles are the classical shape descriptors, which are often used for statistical analysis. Unfortunately, these descriptors do not allow for a complete representation unless one is restricted to work on spheres or ellipsoids. To develop consistent and complete representations of complex particles, researchers have recently been looking for mappings from the unit sphere onto the surface of the particles. When such maps are found, the surfaces can be decomposed into a basis of functions, e.g. spherical harmonics or spherical wavelets. Decomposing the surfaces in a fixed basis introduces representations with the desired two properties. Two different mappings from the unit sphere to the particle surface are currently employed in the literature. They shall be referred to as radial parametrization and surface parametrization.

The radial parametrization method was first introduced by Garboczi [13] and was explored in depth in [14–19]. This approach only applies to so-called star-shaped particles, that is to say particles for which there exists an interior point from which the segment connecting to any other point in the particle lies entirely within the particle [20]. This definition generalizes the notion of convexity as every convex domain is also star-shaped. Because it can be difficult to identify the existence of such a point, we usually only verify if a domain is star-shaped by examining segments originating from its center of mass. When particles are identified as star-shaped, the distance from the center of mass to each point on the surface is a well-defined function on the unit sphere, which can be decomposed in terms of spherical harmonics.

The surface parameterization approach was introduced by Brechbühler et al. [21] and later expanded by Shen and his collaborators [22, 23]. The method was applied to soil particles in the following studies [24–26]. It is based on the principle that any closed surface without holes is topologically equivalent to a sphere. In practice, this topological equivalence is interpreted as stating that there is a bijective and continuous map between points on the unit sphere and points on the surface of the particle. This mapping is not unique and must be determined using a constrained optimization algorithm which minimizes the distortion [21]. Once the mapping is found, it can be approximated using three spherical harmonics decompositions,

one for each component of the parametrization.

Each method has strengths and weaknesses. The radial parametrization is appealing for its natural geometric interpretation. It is indeed easy to roughly guess the point on the surface of the particle that one obtains through the mapping from a given point on the unit sphere. This property is lost when using a surface parametrization since the mapping from the unit sphere to the surface is distorted. Another difference is that a radial parametrization is only well defined on star-shaped particles while a surface parameterization can be applied to every particle that is topologically equivalent to a sphere, which is satisfied by a larger class of particles. The larger versatility of the surface parameterization method comes at a price though since it requires the use of three spherical harmonics decompositions instead of one. Moreover, there is no proof that the surface parametrization is any way unique, and hence may not permit a good comparison of different particles. For the sake of simplicity, the radial parameterization is utilized in this work but most methodologies presented here could also be applied to the surface parameterization if one uses the latter instead of the former.

Both radial and surface parameterizations allow the computation of classical shape descriptors of the particle: volume, aspect ratios, sphericity, and roundness. Garboczi [13] showed how to compute the shape descriptors using a radial parametrization while Zhou et al. [24] showed how to compute the same descriptors using a surface parametrization.

Once the spherical harmonics representations of particles are obtained using the radial parametrization, they can be used to calibrate a statistical model from which virtual particles can be sampled. The most daunting aspect of this task is the high dimensionality of the data, which is typically on the order of several hundred degrees of freedom. It is a well established fact that the complexity of calibrating a statistical model grows exponentially with the dimension of its feature space. This is known as the so-called *curse of dimensionality* [27–29]. Two different statistical models currently dominate the geology literature. Though they differ in some aspects, they are both based on the Principal Components Analysis (PCA) to reduce the dimensionality of the data and to decorrelate the different features.

The first statistical model consists of sampling the principal components of the data from independent normal distributions [17, 19, 24]. Doing so is only theoretically justifiable when the data follows a multivariate Gaussian distribution. Unfortunately, the assumptions of independence in those studies were mainly based on crude visual observations, rather than through statistical hypothesis testing. We suspect that the three studies were not amenable to validation due to the fact that the sample sizes were all extremely small, i.e. 12, 20, and 100.

The second model requires transforming the marginals of each variable into normal distribu-

tions with a Nataf transform [18, 25]. The PCA is then applied on the transformed variables instead of the original ones. As before, the principal components are fitted with normal distributions and sampled independently to generate new particles. This method's main hypothesis is that, by transforming the marginals into normals, the joint distribution should become more similar to a multivariate Gaussian. The issue with this reasoning is that having normal marginals is not a strong enough statement to imply that the data follows a multivariate normal distribution. In fact, it is easy to construct joint distributions which are not multivariate normals but whose marginals follow normal distributions.

To improve the current state of the art, it is important to apply a more rigorous inference of the data distribution based on hypothesis testing. It is also primordial to experiment with generative models which assume looser hypotheses. With the recent success of machine learning algorithms on high-dimensional applications such as clustering, pattern recognition and image generation, there is hope that some of those techniques can be applied to the specific task of this project. In fact, a number of novel generative models on high-dimensional data which range from non-parametric to neural-networks have actually been developed in the past two decades [28, 30].

Several applications of virtual particles sampled from probabilistic models currently exists in the literature. An efficient contact model between SH representations of particles has been developped using the *extent overlap box* [31]. This contact model was implemented within the code Anm which models composite structures of mortar or concrete [32]. In Anm, various virtual particles are suspended in a unit cell with periodic boundary to model a composite material, and the contact algorithm ensures that no overlap between the suspended grains is observed. Other recent studies have used virtual particles in DEM packing simulations. To reduce the cost of dynamically computing contacts between all particles, the SH representations are approximated with sphere clumps [18, 33].

1.4 Contributions

At a conceptual level, one can summarize our contributions by saying that we made use of modern machine learning techniques to generate large datasets of virtual particles. Clustering algorithms are employed to identify subpopulations of particles which share geometrical properties. Partitioning the data in this manner holds the promise of requiring simpler statistical models to capture the geometrical patterns of each subpopulation. To the best of our knowledge, this is the first time these techniques have been used on spherical harmonics representations of particles. Generative models which go beyond the multivariate Gaussian and the Nataf transforms have also been studied. More precisely, our application of the Ker-

nel Density Estimation (KDE) algorithm shows promise on low-dimensional manufactured populations and with a modified version called Manifold-KDE, we obtain encouraging results on high-dimensional data of real particles.

Among other important results is the empirical demonstration that the SH representations of real particles are concentrated near low dimensional manifolds. The possible origin of such manifolds is a point dealt in the thesis. We propose that understanding them can yield geological insight on the data and lead to better generative models exploiting those low dimensional structures. Such models would include, but are not restricted to, Manifold-KDE, Variational Auto-Encoders, Generative Adversarial Networks, and $\mathcal{M}$-flows [28, 34, 35].

Underlying the other conclusions to this research, are the careful and numerous verification processes introduced at each step of the methodology, many of which are lacking from the literature. Some verification analysis original to this thesis include the quadrature study on the unit sphere, the effects of the Lanczos filtering, and the verification of statistical hypothesis tests on simple bivariate distributions.

The final innovation discussed in the thesis is the introduction of particularly useful series of validations based on different families of manufactured particles, i.e. ellipsoids and superquadrics. Working with said particles allows to validate of the statistical models since the exact geometries and distributions of the particles can be controlled.

1.5 Organization of the Thesis

Here is how the content will be divided between chapters:

- Chapter 2 focuses on the discretization step in Figure 1.2. More precisely, the surface triangular mesh (STM) and the spherical harmonics (SH) representations, as well as their computation, are described in great detail. Convergence studies are also presented.
- Chapter 3 tackles the challenge of defining the geometrical resemblance between particles. Indeed, the main requirement of the generative model is that virtual particles be similar to the original particles. The notion of resemblance is frequently defined in terms of the following classical descriptors: volume, aspect ratios, sphericity, and roundness. All algorithms implemented for their calculation are extensively verified with the use of simple manufactured particles.
- Chapter 4 describes in detail the preprocessing, calibration, and sampling steps as shown in Figure 1.2. More precisely, the Principal Component Analysis (PCA), KMean, Gaussian Mixture (GM), and Kernel Density Estimation (KDE) algorithms are discussed.
- Chapter 5 presents some numerical results that demonstrate the efficacity of the generative process to create virtual particles when starting from real or manufactured particles. Synthetic or manufactured particles are used to validate the appropriateness of the statistical models while real particles illustrate the performance of the overall process in the case of real-life applications.
- Chapter 6 provides some concluding remarks about our work and explores possible avenues for future work.

CHAPTER 2 DISCRETE REPRESENTATIONS OF PARTICLES

The first step in the development of the generative process is to obtain a discrete representation of real particles collected from a given geological source, see Figure 1.2. Ideally, the discrete representation should have specific properties which allow for the statistical analysis of samples of particles. To that end, this chapter introduces the mathematical definitions of two representations of particles: the surface triangular mesh (STM) and the spherical harmonics (SH) representations, which are both used during the discretization process. First, the STM representation of particles is discussed, the definition of a surface tessellation is provided, and defective tessellations are investigated. Secondly, the SH representation is introduced and its numerical computation is detailed extensively. The chapter concludes with a visual comparison of four arbitrary particles in both representations.

2.1 Surface Triangular Mesh Representation (STM)

With the advent of inexpensive X-ray micro-computed tomography (μCT), it has been possible to characterize a host of materials and objects. This non-invasive technology enables a resolution at the micrometer scale of bones, archeological artifacts, biological materials, and grains. The characterization is commonly given with a triangulation of the surface of the particle. To make subsequent text lighter, the surface triangular mesh representation shall be referred to as the STM representation. Figure **??** illustrates the STM representation of a grain picked from a river bed.

The STL format (Stereolithography) [36], is often used to store the necessary information to describe such representations. More concretely, an STL file contains N_v vertices stored in a two-dimensional set $\mathbf{V} = ((x_i, y_i, z_i))_{1 \leq i \leq N_v}$. The set $\mathbf{V}$ can be seen as a $N_v \times 3$ matrix whose rows represent the nodes in 3D space. The STL file also lists the N_f triangles forming the (hopefully) closed surface. A connectivity matrix $\mathbf{K}$ of size $N_f \times 3$ ties the local vertices of the faces to the global vertices in $\mathbf{V}$. The file also contains normals on each face stored in a matrix $\mathbf{N}$ of size $N_f \times 3$. To index elements of $\mathbf{V}$, $\mathbf{N}$ and $\mathbf{K}$, we will use a matlab-like notation $\mathbf{N}(i, j)$, $\mathbf{V}(i, j)$ and $\mathbf{K}(i, j)$ where the indices i and j start at 1. For example, $\mathbf{V}(i, j)$ represents the jth coordinate of the ith vertex. Also, $\mathbf{K}(i, j)$ outputs the global index of the jth local vertex of the ith face. Finally, to index a whole row or column, we will use the ":" symbol so the first vertex of the file would be obtained with $\boldsymbol{V}(1, :) \in \mathbb{R}^3$ for example. This notation is standard throughout the document when indexing the elements of a matrix.

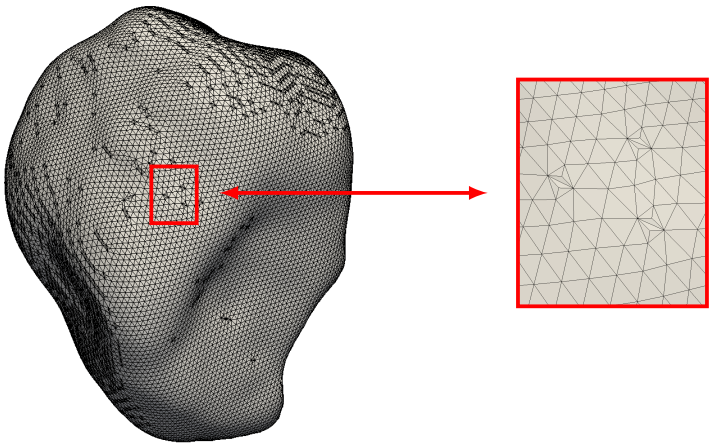
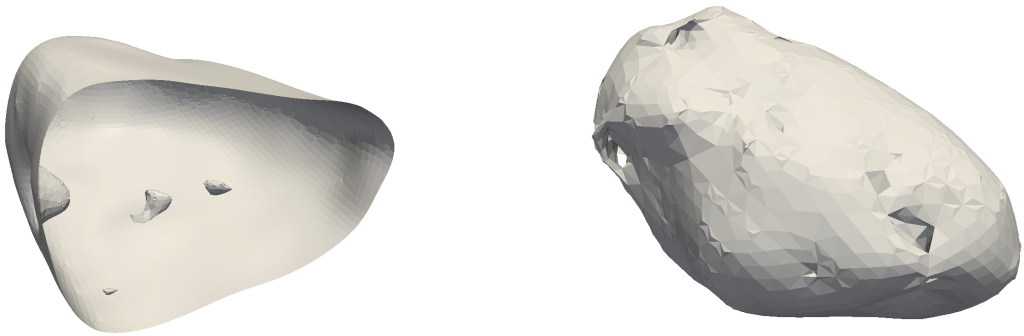


Figure 2.1 STM representation of a river particle.

In differential geometry, surface triangulations are assumed to be consistent in the sense that i) each triangle (2-simplex) is formed of 3 different segments (1-simplex), ii) each segment is formed of two nodes (0-simplex) and iii) each $n-1$-simplex is the intersection of exactly two n-simplices. Without these conditions, the triangulation will not correspond to the triangulation of a 2-manifold, but the necessity of such conditions was the result of decades of debates, as detailed in the respected book of Imre Lakatos [37].

The STM representations of four distinct populations of particles obtained with micro-computed tomography were made available to us by Hydro-Québec. Those populations include asphalt, river, rouge and margelle particles. Unfortunately, the surface triangulations obtained by μCT are prone to disrespect the consistency conditions described earlier. The most common errors are the presence of 2-simplexes suspended inside the particle and holes on the particle surface. Both defects result in some edges that are not shared between two 2-simplexes, which violates the third condition of consistent triangulations. Figure 2.2 shows examples of two such defects. The most extreme defect was having multiple particles glued together, see Figure 2.3. Though the triangulation may be consistent on such examples, they obviously represent non-physical particles.

Automating the removal of such defects is a rather complicated task in computer graphics and is beyond the scope of this thesis. Hence, we have chosen to simply eliminate surface triangulations that contain too many defects. In practice, we observe that grains with defects are not more or less complicated than those without, indicating that these are caused by the micro-tomography, and that the geometrical results we would deduce would still be valid, even after the exclusion of certain particles. The selection process can be applied to the four populations of particles described earlier and the results are reported in Table 2.1.



(a) 2-Simplexes suspended inside the volume

(b) Holes on the surface

Figure 2.2 Two common defects on STM representations.

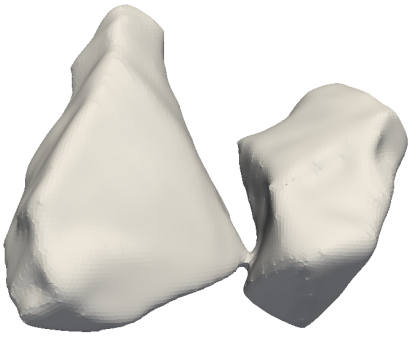


Figure 2.3 Extreme defect in the STM representations.

Table 2.1 – Amount of usable and defective files for each population.

	Asphalt	**River**	**Rouge**	**Margelle**
usable	301	1005	264	892
defective	1	53	52	0
total	302	1058	316	892

The STM representation of grains is convenient for visualization with computer software, however, it is not well adapted to statistical analysis. The reason being that, to analyze samples of multiple particles, one needs a representation that is consistent and complete. By consistent, we mean that each particle is represented with the same number of features

which all characterize the same geometric information. By complete, we impose that, given the set of all features, one should be able to reconstruct the associated particle to some arbitrary accuracy. When working on STM representations, one could be tempted to use the nodes as features. However, the STL files do not have the same number of nodes. Moreover, comparing the ith node from two different STL files does not yield any insight, since we could always perform rotations. For these two reasons, the STM representation is not consistent. The first important challenge to this research is to find a consistent and complete representation of soil particles. A classical approach would be to use features such as volume, aspect ratios, sphericity, convexity, and roundness. Though this representation is consistent, it is not complete, i.e. only simple artificial grains are uniquely determined by those few quantities, while the particles we study typically have a rich geometry and cannot be fully reconstructed when only given their classical shape descriptors.

2.2 Spherical Harmonics Representation (SH)

In order to find a consistent and complete representation, we need to explore the concepts of Fourier series and Spherical Harmonics. The general Fourier Series are ubiquitous in science and engineering but the theory of Spherical Harmonics appears mostly in specialized applications. For this reason, we begin this section by introducing the Fourier series in the general sense and afterwards specify how they can be apply to the representation of 3D particles, which will naturally lead to the formulation of the Spherical Harmonics. Most of the content of Section 2.2.1 is based on the first and fourth chapters in [38].

2.2.1 Fourier Series

The theory attempts to describe functions belonging to the space of squared integrable functions

$$L^2(\Omega) = \Big\{ f : \Omega \to \mathbb{R} \ \text{ such that } \ \int_\Omega f^2(x)dx \leq \infty \Big\}, \tag{2.1}$$

which is a Hilbert space equipped with the following inner product and norm

$$\langle f, g \rangle_\Omega = \int_\Omega f(x)g(x)dx, \tag{2.2}$$

$$\|f\|_\Omega^2 = \langle f, f \rangle_\Omega = \int_\Omega f^2(x)dx. \tag{2.3}$$

For simplicity, we assume here that Ω is an open bounded domain with a piecewise smooth

boundary. A set $\{u_i\}_{i\in\mathbb{N}}$ of functions in $L^2(\Omega)$ is called an *orthonormal system* if

$$\langle u_i, u_j\rangle_\Omega = \delta_{ij}, \quad \forall i,j \in \mathbb{N}, \tag{2.4}$$

An orthonormal system $\{u_i\}_{i\in\mathbb{N}}$ forms an *orthonormal basis* of $L^2(\Omega)$ if and only if its span is a dense subset, i.e. for every $f \in L^2(\Omega)$ there exists a sequence $(c_i)_{i\in\mathbb{N}} \in \mathbb{R}^\infty$ such that

$$\lim_{n\to\infty} \left\| f - \sum_{i=1}^{n} c_i u_i \right\|_\Omega \to 0. \tag{2.5}$$

A natural consequence of the structure of Hilbert spaces is that the coefficients $(c_i)_{i\in\mathbb{N}}$, which satisfy (2.5), are unique and are obtained through the L^2 projection

$$c_i = \langle f, u_i\rangle_\Omega. \tag{2.6}$$

The coefficients c_i are called the *components* or the *Fourier coefficients* of f and each contribution $c_i u_i$ is referred to as a *mode*. We note that in practice, many different orthonormal bases for $L^2(\Omega)$ may exist, and so many sequences of Fourier coefficients may be associated to the same function f. Though convergence in $L^2(\Omega)$ is always guaranteed, pointwise convergence can be achieved under sufficient smoothness assumptions on f. According to Theorem 2.5 in [39], the following holds for one-dimensional domains,

$$f \in C^1(\Omega) \implies f(x) = \sum_{i=1}^{\infty} c_i u_i(x), \ \forall x \in \Omega, \tag{2.7}$$

which could be relaxed to continuous piecewise smooth functions, i.e. those corresponding to the class of functions provided by the STM representation. An important property of orthonormal bases is *Parseval's equality*

$$\|f\|_\Omega^2 = \sum_{i=1}^{\infty} \langle f, u_i\rangle_\Omega^2 = \sum_{i=1}^{\infty} c_i^2 \leq \infty. \tag{2.8}$$

This equality sheds light on a strict constraint over the coefficients

$$(c_i)_{i\in\mathbb{N}} \in \ell^2(\mathbb{N}) = \Big\{ \boldsymbol{c} \in \mathbb{R}^\infty \ \text{ such that } \ \sum_{i=1}^{\infty} c_i^2 \leq \infty \Big\}. \tag{2.9}$$

To summarize, using an orthonormal basis defines a bijective isometry between the space of functions $L^2(\Omega)$ and the space of coefficients $\ell^2(\mathbb{N})$, the latter being easier to work with from a statistical point of view than the former. In practice, one must truncate $(c_i)_{i\in\mathbb{N}}$ to obtain a finite amount of features to store in a dataset. Let $n_{\max}$ be the maximal number of features selected, the L^2 error of the representation would be, using (2.8),

$$\begin{aligned}\left\| f - \sum_{i=1}^{n_{\max}} c_i u_i \right\|_\Omega^2 &= \|f\|_\Omega^2 - \sum_{i=1}^{n_{\max}} c_i^2 \\ &= \sum_{i=n_{\max}+1}^{\infty} c_i^2,\end{aligned} \tag{2.10}$$

which shows that the truncation error is closely related to the decay rate of the coefficients $(c_i)_{i\in\mathbb{N}}$. According to Theorem 2.6 in [39], the decay rate of the Fourier coefficients is linked to the regularity of the function f,

$$|c_i| \leq C\, i^{-k-1}, \tag{2.11}$$

where k is the regularity of f, i.e. the number of continous derivatives. This result is only valid in one dimension, however, similar results can be derived in higher dimensions. The decay rate $k+1$ according to the regularity of f which is an intuitive observation. Very irregular functions require a high frequency modes to faithfully represent their steep variations so their coefficients have a smaller decay rate.

A well known truncation artefact of the Fourier series is the Gibbs phenomenon, which occurs when truncating the representation of a function with discontinuities. For example, let $\Omega = [-1, 1]$ and f be a step function over Ω. Figure 2.4 shows the Fourier series of f using a cosine orthonormal basis $\cos(x)$. Adding more modes improves the approximation but the largest overshoot converges to a constant 18% overshoot. Common approaches to reduce Gibbs phenomenon consist of applying a high frequency filter to the components which reduces the ripples at the cost of having a larger L^2 error.

2.2.2 Spherical Harmonics

Having described the Fourier Series in the general sense, we now specialize to their application to represent 3D particles, which will lead to the definition of the Spherical Harmonics. As

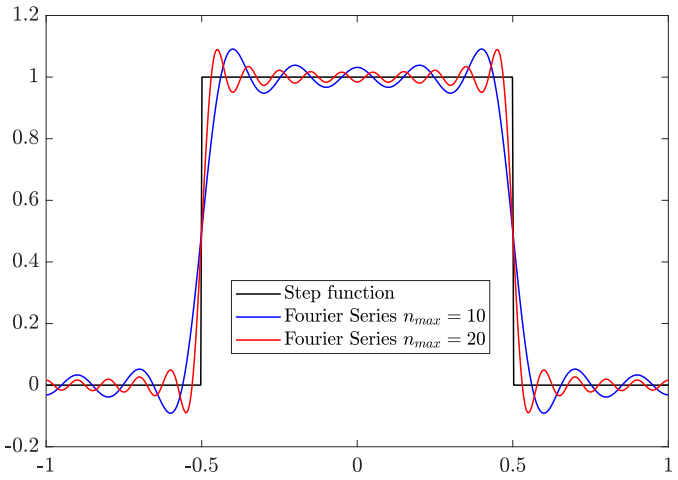


Figure 2.4 The Gibbs phenomenon generates an overshoot (and undershoot) that is equal to 18% of the strength of the discontinuity. This overshoot does not decrease as the number of modes increases.

discussed in the previous chapter, finding a continuous map from the unit sphere

$$S^2 := \Big\{ \boldsymbol{x} \in \mathbb{R}^3 : \sum_{i=1}^{3} x_i^2 = 1 \Big\}, \tag{2.12}$$

to the particle surface is the first step to develop a consistent and complete representation of realistic soil particles. Two different mappings were explained in Section 1.3 and the radial parametrization has been selected. To define this parametrization, the spherical coordinate system of $\mathbb{R}^3$ must be introduced,

$$\boldsymbol{r} = (r\ \cos\theta \sin\phi, r\ \sin\theta \sin\phi, r\ \cos\phi), \tag{2.13}$$

where $\theta \in [0, 2\pi[$ is called the azimuth angle and $\phi \in [0, \pi]$ is called the polar angle, see Figure 2.5. Since there is a bijection between S^2 and the angles (θ, ϕ), a function $f(\theta, \phi)$ defines a function on the unit sphere. To define arbitrary grains as functions over the unit sphere, the class of *star-shaped* particles must be considered. A particle is said to be star-shaped if it contains a fixed point such that any segment connecting this point to another interior point lies within the particle [20]. This definition is a more relaxed version of convexity where only one interior point is arbitrary, which implies that convex particles constitute a subset of star-shaped particles. If one uses the fixed point as the origin of a spherical coordinate system, then a star-shaped particle can be represented with a function $r(\theta, \phi)$ defining the

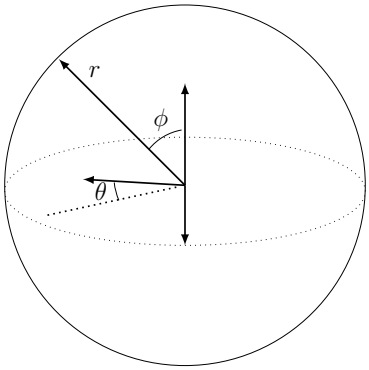
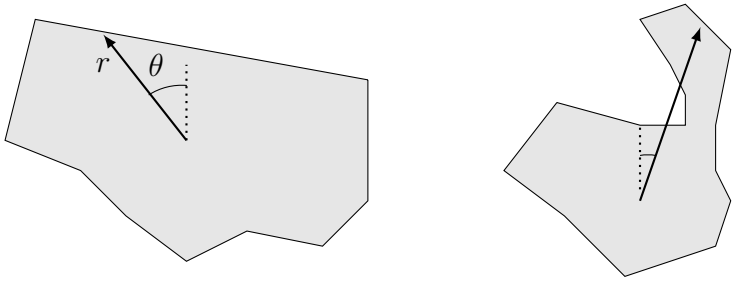


Figure 2.5 Spherical coordinates.



Star-Shaped Non Star-Shaped

Figure 2.6 Slices of a star-shaped and non star-shaped particle.

distance from the origin to the surface in every direction. Figure 2.6 illustrates examples of star-shaped and non star-shaped particles.

Every particle must be represented with a unique function $r(\theta, \phi)$, however, for any given function, pre-composition with an isometry of the sphere will generate a new map. This can be solved by consistently measuring the angles θ and ϕ with respect to the principal axes of the particle, which are explained in Section 3.1.2. Moreover, even the center used in the definition of the star-shaped domain is not necessarily unique. The common solution is to use the center of mass, though it is not always one of the points with respect to which the particle is star-shaped (but it is reasonable). The Hilbert space $L^2(S^2)$ can now be introduced with respect to the the following inner product

$$\langle f, g \rangle_{S^2} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi)\, g(\theta, \phi)\, \sin\phi \, d\phi d\theta. \tag{2.14}$$

The spherical harmonics on S^2 correspond to the eigenfunctions of the surface Laplacian ∇^2. Since the Laplacian is a compact self-adjoint operator, its eigenfunctions form an orthonormal

basis of $L^2(S^2)$ [40] and elliptic regularity implies that the eigenfunctions are smooth. They are given by

$$Y_\ell^m(\theta,\phi) = \begin{cases} \sqrt{2}C(\ell,|m|)\, P_\ell^{|m|}(\cos\phi)\, \cos(m\theta), & m>0, \\ C(\ell,0)\, P_\ell^0(\cos\phi), & m=0, \\ \sqrt{2}C(\ell,|m|)\, P_\ell^{|m|}(\cos\phi)\, \sin(-m\theta), & m<0, \end{cases} \tag{2.15}$$

where

$$C(\ell,|m|) = \sqrt{\frac{(2\ell+1)\,(\ell-|m|)!}{4\pi\,(\ell+|m|)!}},$$

is a normalization constant and $P_\ell^{|m|}$ are the associated Legendre polynomials. The spherical harmonics are indexed by two the indices ℓ and m, which are associated with frequencies in the ϕ and θ *directions* ans satisfy

$$\begin{aligned} &\ell = 0,1,2,3\ldots \\ &m = -\ell,-\ell+1,\ldots,-1,0,1,\ldots,\ell-1,\ell. \end{aligned} \tag{2.16}$$

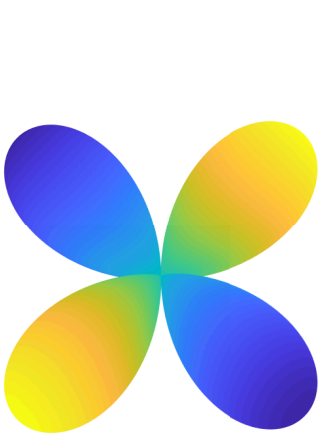
Examples of harmonics are shown in Figure 2.7. Since the spherical harmonics form a basis, any square integrable surface function $r(\theta,\phi)$ can be represented in term of its Fourier coefficients (2.6), which will be referred from now on as its SH coefficients and will be denoted c_ℓ^m. To obtain a dataset of multiple particles using a finite number of features, one must truncate the sequence to an index $\ell = \ell_{\max}$. Therefore, only $(\ell_{\max}+1)^2$ different SH coefficients c_ℓ^m are considered, which provides an approximation of the true function $r(\theta,\phi)$

$$r(\theta,\phi) \approx \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} c_\ell^m Y_\ell^m(\theta,\phi). \tag{2.17}$$

It was previously discussed that the L^2 truncation error is related to the decay of the Fourier coefficients. The energy associated with the frequency ℓ characterizes this decay in the framework of the spherical harmonics [18, 24, 26]

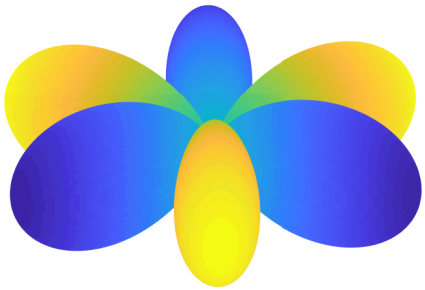
$$E(\ell) = \sqrt{\sum_{m=-\ell}^{m=\ell} \left|c_\ell^m\right|^2}, \tag{2.18}$$

where the relation between the decay rate of the energy and the regularity of $r(\theta,\phi)$ is a lot more complicated than in (2.11). We refer to the first example of [41] for additional insight. Harmonics are in $\mathcal{C}^\infty(S^2)$ and the expression of their first and second derivatives are provided

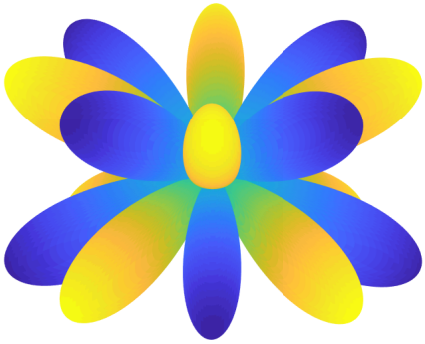


(a) $\ell = 2, m = -1$

(b) $\ell = 3, m = 0$



(c) $\ell = 3, m = 3$



(d) $\ell = 5, m = 4$

Figure 2.7 Absolute value of the real valued harmonics for different values of ℓ and m. The colormap shows which values are positive (blue) and negative (yellow).

in appendix B.

2.2.3 Implementation

We now go through the numerical computation of the SH coefficients when given the STM representation of a particle. The main steps are the interpolation of the surface and the discretization of the L^2 projection. Error sources and convergence are discussed afterwards.

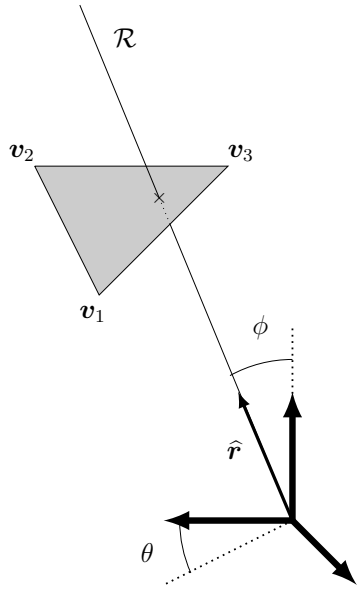


Figure 2.8 Interpolation of the grain surface.

Linear interpolation: Computing the SH representation requires the ability to compute the particle surface function $r(\theta, \phi)$ in arbitrary directions. The objective is therefore to compute, for each direction $\widehat{\boldsymbol{r}} \in S^2$, the intersection point between the triangulated surface and the ray

$$\mathcal{R} := \{\boldsymbol{r} \in \mathbb{R}^3 \,|\, \boldsymbol{r} = \alpha\, \widehat{\boldsymbol{r}}, \alpha \in \mathbb{R}^+\}. \tag{2.19}$$

Since the surface is described by a collection of triangles, the problem consists in finding the unique triangle that intersects the ray $\mathcal{R}$. The three vertices of the ith triangle, with i belonging between 1 and N_f, are obtained as

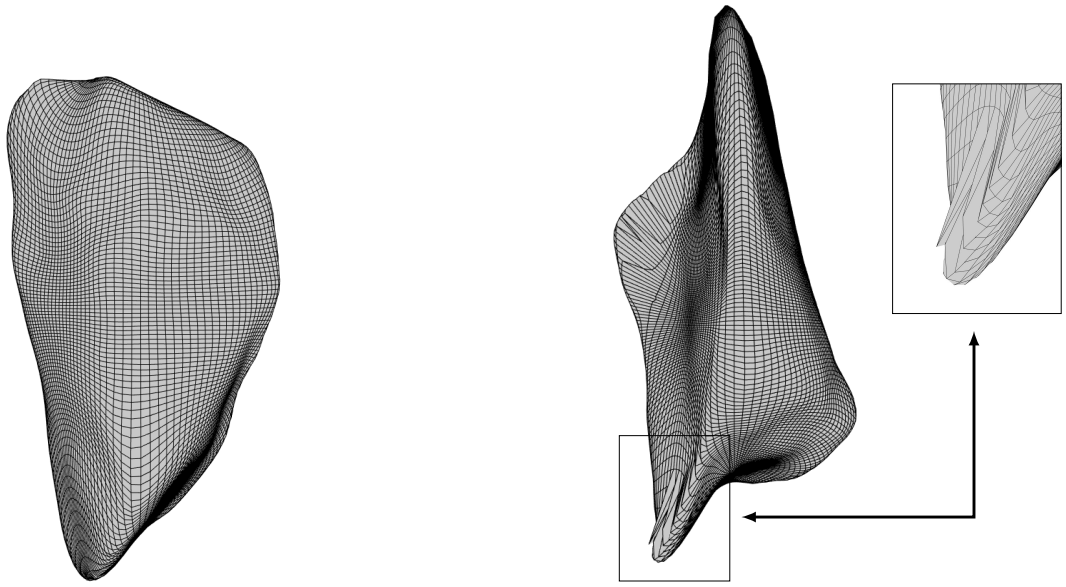
$$\boldsymbol{v}_1 = \mathbf{V}(\mathbf{K}(i,1),:),\ \boldsymbol{v}_2 = \mathbf{V}(\mathbf{K}(i,2),:),\ \boldsymbol{v}_3 = \mathbf{V}(\mathbf{K}(i,3),:).$$

To identify the unique triangle pierced by the ray $\mathcal{R}$, we need to verify each triangle individually by computing:

1. The intersection point between the ray and the plane generated by the triangle,
2. Whether or not the intersection point is located inside the triangle.

The technique is illustrated in Figure 2.8. Once the pierced triangle is identified, the contact point distance from the origin is chosen as the interpolated value $\mathcal{I}(\theta, \phi)$.

The technical details of the algorithm are omitted but they involve computing the barycentric coordinates of each triangle. Figure 2.9 demonstrates results of the interpolation on two



(a) $\mathcal{I}(\theta, \phi)$ of star-shaped particle

(b) $\mathcal{I}(\theta, \phi)$ of non star-shaped particle

Figure 2.9 Interpolation of two asphalt particles.

asphalt particles. It appears that the surface of the star-shaped particle from Figure 2.9(a) is faithfully approximated by linear interpolation. However, some issues occur when trying to interpolate the surface of a non star-shaped particle, see Figure 2.9(b). By zooming at the bottom of the particle, very abrupt variations in the interpolation can be observed. These variations are attributed to the non star-shapedness of the particle, which causes the ray $\mathcal{R}$ to pierce the surface multiple times. In those instances the interpolation algorithm struggles to assign distances from the origin in a smooth manner. These strong irregularities can cause large ripples on the surface of the SH representation and a very slow decay of the SH coefficients.

Gauss Quadrature: Assuming now that the function $\mathcal{I}(\theta, \phi)$ could be computed for all (θ, ϕ) on the sphere, one must estimate the integral $\langle r, Y_\ell^m \rangle_{S^2}$ by a quadrature rule over the unit sphere. Several schemes are possible but the most common one in the literature is the Gauss Quadrature involving between 14,000 and 60,000 Gauss points sampled in the $\theta - \phi$ plane [13, 14, 20]. This scheme is simple since is does not require discretizing the sphere into patches. Despite the importance of the quadrature on the values of the SH coefficients,

comparisons with other schemes are rarely, if ever, made.

Rather than simply selecting a method of approximation, this thesis will attempt to establish guidelines for this process. We begin by defining a partition of the sphere as a collection of N_e subsets called elements $\Omega_i \subset S^2$ such that

$$\bigcup_{i=1}^{N_e} \Omega_i = S^2 \quad \text{and} \quad \Omega_i \cap \Omega_j = \varnothing \text{ for } i \neq j. \tag{2.20}$$

The most straightforward partition of the sphere is to use rectangles in $[0, 2\pi[\times]0, \pi[$ such that each element of the partition is described by

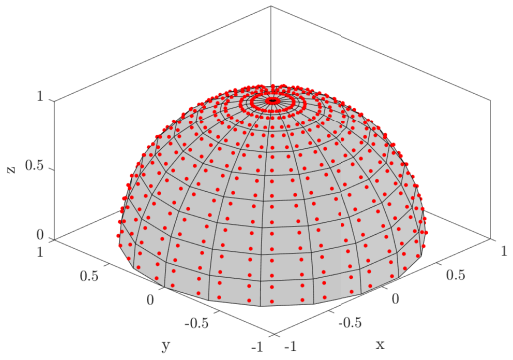
$$\Omega_i := \{\boldsymbol{r} \in S^2 \mid \theta_i \leq \theta < \theta_i + \Delta\theta_i \,,\, \phi_i \leq \phi < \phi_i + \Delta\phi_i\}, \tag{2.21}$$

where (θ_i, ϕ_i) is a lower left corner of the rectangle and $\Delta\theta_i$ and $\Delta\phi_i$ may depend on i. The three partitions we consider are

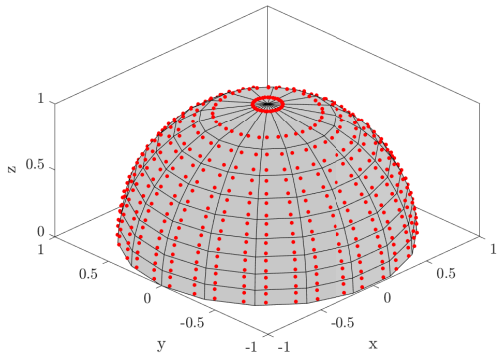
1. The *uniform mesh* with constant $\Delta\theta$ and $\Delta\phi$, independent of i;
2. The *semi-uniform mesh* with constant $\Delta\theta$, but $\Delta\phi$ dependent on ϕ;
3. The *igloo mesh* where the area of Ω_i is constant over all elements.

Since the elements are rectangular, the Gauss points inside them can be computed by a tensor-product of the one-dimensional Gauss points [42]. Figure 2.10 shows the distributions of the elements and Gauss points on the upper half of the unit sphere with 4 Gauss points per element.

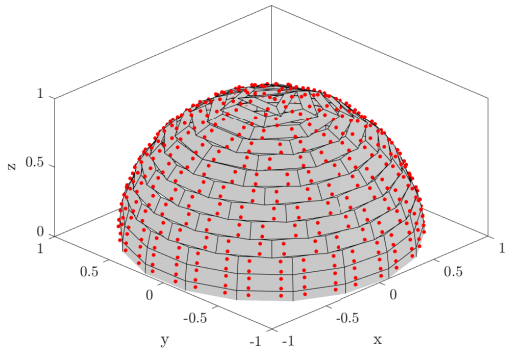
The uniform mesh is very straightforward to understand and implement, however the number of quadrature points is probably suboptimal because they become concentrated near the poles. This concentration of points is easily explained by the vanishing of the Jacobian when looking at the mapping from the $\theta - \phi$ plane to the sphere. The second mesh is uniform in θ but not in ϕ. It is designed to counterbalance the fact that the points get more concentrated at the poles, at least with respect to ϕ. The idea is to generate a uniform grid with respect to $\alpha \in [0, 1]$ and generate the non-uniform grid for ϕ by mapping $\phi = \operatorname{acos}(1 - 2\alpha)$. This mapping is primarily used in statistics to sample points uniformly from S^2. However, for this application, it seems like an over-correction because there are much fewer points in the vicinity of the poles. The third mesh generates elements of same area, producing a partition of the sphere with the appearance of an igloo.



(a) Gauss points on uniform mesh.



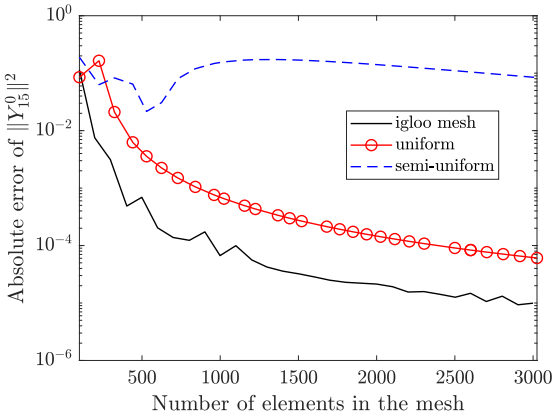
(b) Gauss points on semi-uniform mesh.



(c) Gauss points on igloo mesh.

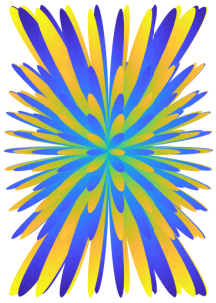
Figure 2.10 Gauss points distribution for various meshes.

(a) Absolute value of Y_{15}^0

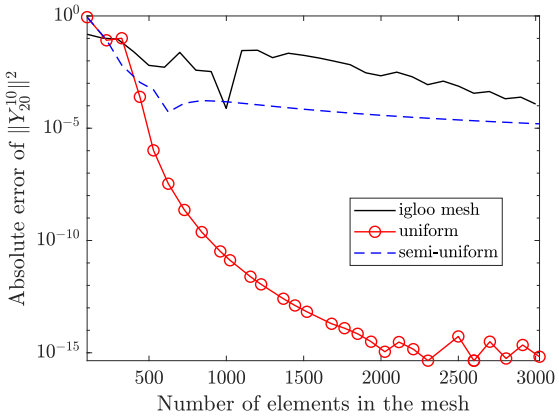


(b) Convergence

Figure 2.11 Discretization errors for the approximation of $\|Y_{15}^0\|^2$ over three meshes.

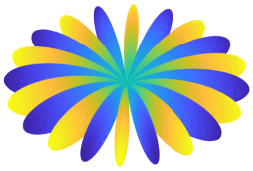


(a) Absolute value of Y_{20}^{10}

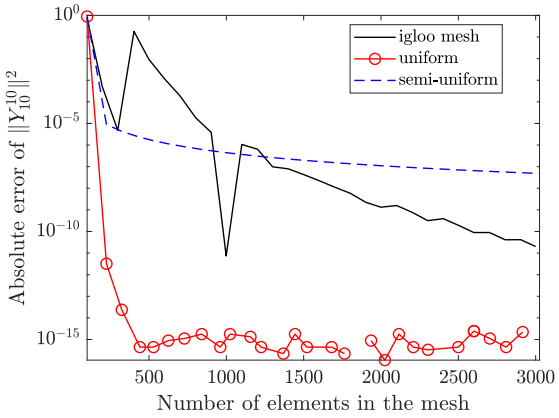


(b) Convergence

Figure 2.12 Discretization errors for the approximation of $\|Y_{20}^{10}\|^2$ over three meshes.



(a) Absolute value of Y_{10}^{10}



(b) Convergence

Figure 2.13 Discretization errors for the approximation of $\|Y_{10}^{10}\|^2$ over three meshes.

To select the best mesh, the following integral

$$\|Y_\ell^m\|_{S^2}^2 = \int_0^{2\pi} \int_0^{\pi} |Y_l^m(\theta, \phi)|^2 \sin\phi \; d\phi d\theta, \tag{2.22}$$

which is known to equal unity for all ℓ and m (2.4) can be evaluated using all three partitions. The partitions that yields the smallest absolute error of (2.22) with a fixed number of elements and Gauss points per element can be selected. High values of ℓ and m are used because their associated harmonics are prone to abrupt variations, which makes them harder to integrate. Figures 2.11, 2.12, and 2.13 illustrate the absolute errors of the integrals of Y_{15}^0, Y_{10}^{20}, and Y_{10}^{10}

when considering $9 = 3 \times 3$ Gauss points per element and various numbers of elements. We observe that, on these examples, the uniform mesh is better or equivalent than the two other meshes. For this reason it is selected as the mesh used to compute all our integrals over the unit sphere.

Given a function $r(\theta, \phi)$, the coefficients of the SH expansion are approximated as:

$$\begin{aligned}
c_l^m &= \langle r, Y_\ell^m \rangle_{S^2} \\
&= \iint_{S^2} r(\theta,\phi)\, Y_l^m(\theta,\phi) dS \\
&= \sum_{i=1}^{N_e} \iint_{\Omega_i} r(\theta,\phi)\, Y_l^m(\theta,\phi) d\Omega_i \\
&= \sum_{i=1}^{N_e} \int_{\theta_i}^{\theta_i+\Delta\theta_i} \int_{\phi_i}^{\phi_i+\Delta\phi_i} r(\theta,\phi)\, Y_l^m(\theta,\phi) \sin\phi \; d\phi d\theta \\
&\approx \sum_{i=1}^{N_e} \int_{\theta_i}^{\theta_i+\Delta\theta_i} \int_{\phi_i}^{\phi_i+\Delta\phi_i} \mathcal{I}(\theta,\phi)\, Y_l^m(\theta,\phi) \sin\phi \; d\phi d\theta \\
&= \sum_{i=1}^{N_e} \int_{\theta_i}^{\theta_i+\Delta\theta_i} \int_{\phi_i}^{\phi_i+\Delta\phi_i} f(\theta,\phi)\, d\phi d\theta \qquad (\text{with } f(\theta,\phi) := \mathcal{I}(\theta,\phi)\, Y_m^\ell(\theta,\phi)\, \sin\phi) \\
&\approx \sum_{i=1}^{N_e} \frac{\Delta\phi_i \Delta\theta_i}{4} \sum_{j=1}^{N_g} \sum_{k=1}^{N_g} \omega_j \omega_k f\Big(0.5\,\Delta\theta_i(\xi_j+1) + \theta_i \;,\; 0.5\,\Delta\phi_i(\eta_k+1) + \phi_i\Big),
\end{aligned} \tag{2.23}$$

where N_g is the number of Gauss points in each direction, ω_j and ω_k are the Gauss weights, and ξ_j and η_k denote the one-dimensional coordinates of the Gauss points in the interval $[-1, 1]$.

Convergence and errors: Before studying the convergence of the SH representation, it is necessary to investigate the various sources of errors involved in approximating a particle by a SH representation. The three principal errors are

1. Truncation of the Fourier series at $\ell_{\max}$;
2. Surface interpolation of particles which are not star-shaped;
3. Numerical integration of the SH coefficients.

Since discretization errors of the integrals can be reduced indefinitely at the cost of using more elements and Gauss points, the induced errors are easier to control based on our previous

experiments. As discussed earlier, the truncation of the Fourier series and irregularities in the interpolation $\mathcal{I}(\theta, \phi)$, as exhibited in Figure 2.9(b), can cause spurious oscillations on the surface of the SH representations. These errors are harder to keep in check since they depend on irregularities, or lack of convexity, of the particles, combined with the Gibbs phenomenon. To reduce the ringing for all particles of a sample, the authors in [16] have applied the so-called Lanczos sigma factor to the SH coefficients

$$c_\ell^{'m} = c_\ell^m \text{sinc}\left(\frac{(\ell - \ell_0)\pi}{(\ell_{\max} - \ell_0)}\right), \quad \ell_0 \leq \ell \leq \ell_{\max} \tag{2.24}$$

where $\text{sinc}(x) = \frac{\sin(x)}{x}$ is the cardinal sine and ℓ_0 is the frequency where the filter starts being applied. This has the effect of reducing ripples at the cost of having a larger L^2 error. However, empirical evidence suggest that applying a high-frequency filter to the SH coefficients can oversmooth the function, resulting in a loss of finer details [43].

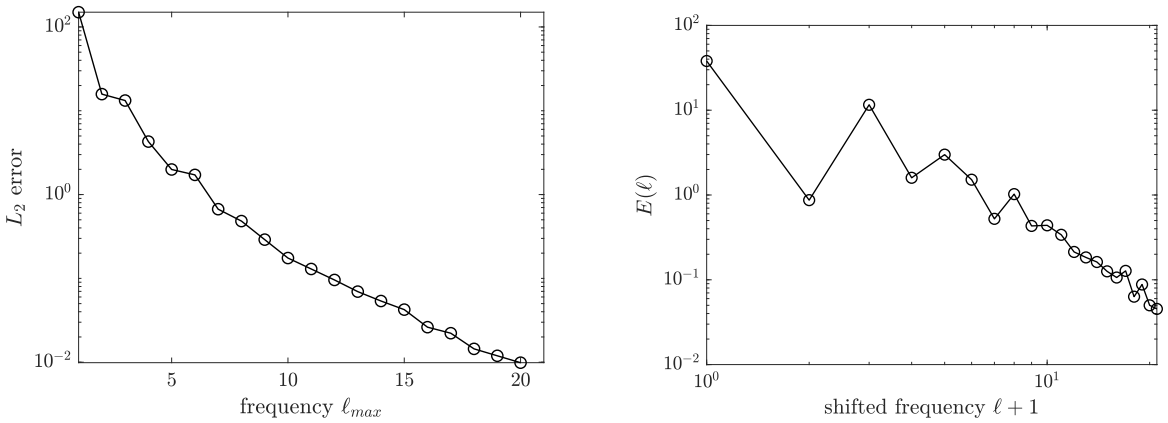
An alternative solution to the Gibbs phenomenon is to apply a Gegenbauer polynomial decomposition of the approximation $\sum_{\ell=1}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} c_\ell^m Y_\ell^m$. This technique requires the use of an edge detection algorithm that identifies and labels very steep variations in the function $r(\theta, \phi)$ as discontinuities [44]. A Gegenbauer polynomial decomposition of the SH representation is then done in the regions between the edges [43, 45]. Although this method is able to correct the Gibbs phenomenon without oversmoothing the function, its implementation is considerably more complex than the Lanczos filter. For this reason, the Lanczos filter was chosen in this work.

Now that all error sources have been described, the convergence study of the SH representation of particles can be conducted. First thing is to check whether the L^2 error between the STM and SH representations does decrease as more modes are added (2.5). The L^2 error is computed as

$$\epsilon = \left\| \mathcal{I} - \sum_{\ell=1}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} c_\ell^m Y_\ell^m \right\|_{S^2}^2. \tag{2.25}$$

The energy as a function of ℓ (2.18) can also be calculated to observe the decay rates of the coefficients. Figure 2.14(a) shows convergence of the L^2 error with respect to $\ell_{\max}$ and Figure 2.14(b) illustrates energy for various values of ℓ. Note that the frequency ℓ is shifted by 1 to avoid a singularity in the logarithmic scale. We see that the L^2 error is strictly decreasing which is an indicator that the implementation is not incorrect. The energy seems to reach the asymptotic regime $E(\ell) \sim \ell^{-\beta}$ for $\beta \approx 2$ when ℓ reaches 10.

To confirm the relation between regularity and coefficient decay rate, the energies for three



(a) L_2 error between $\mathcal{I}$ and SH expansion

(b) Energy with respect to the cutoff frequency

Figure 2.14 Convergence of the L^2 error and energy for an asphalt particle.

different particles are computed, see Figure 2.15. It appears that the SH coefficients of the regular particles have faster convergence rate than the SH coefficients of the very irregular particle (grain3). This suggests that $\ell_{\max}$ should be selected by studying the most complex particles in a sample.

2.2.4 Geometrical Interpretation

In this section, we introduce an intuitive geometrical interpretation that can be associated to the SH coefficients. This interpretation will help the reader to appreciate why the SH are particularly effective for simple shapes, but much more difficult to obtain for realistic grains. We begin by observing that the mean value of a function f on the sphere can be computed as

$$\langle f \rangle := \frac{1}{4\pi} \iint_{S^2} f(\theta, \phi) dS. \tag{2.26}$$

According to (2.15), $Y_0^0(\theta, \phi) = \frac{1}{\sqrt{4\pi}}$ so one obtains

$$c_0^0 = \langle r, Y_0^0 \rangle_{S^2} = \frac{1}{\sqrt{4\pi}} \iint_{S^2} r(\theta, \phi) dS = \sqrt{4\pi} \langle r \rangle. \tag{2.27}$$

In other words, the first SH mode $c_0^0 Y_0^0$ describes a sphere of radius $\langle r \rangle$. Subsequent SH modes act as perturbations of this initial approximation. Considering the coefficients c_ℓ^m, where $m, \ell \neq 0$, one can denote the perturbation they induce on the sphere by $\Delta_\ell^m := c_\ell^m Y_\ell^m$. Figure 2.16 illustrates how this perspective makes the spherical harmonics easier to interpret. The radial perturbations Δ_ℓ^m have the nice property that their mean value is zero, which is

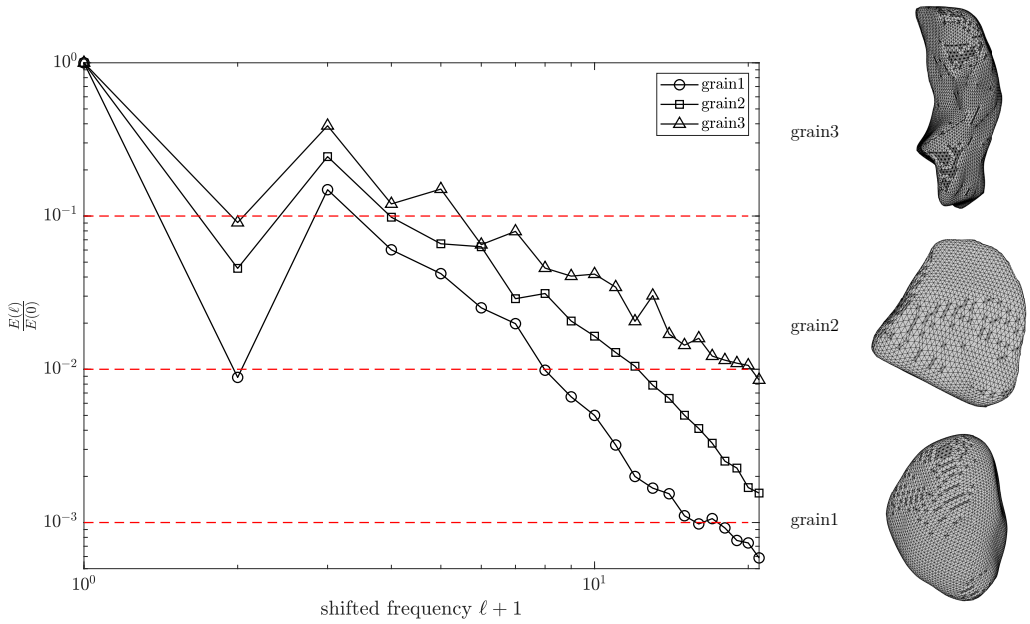


Figure 2.15 Decay of the energy with respect to ℓ for three particles.

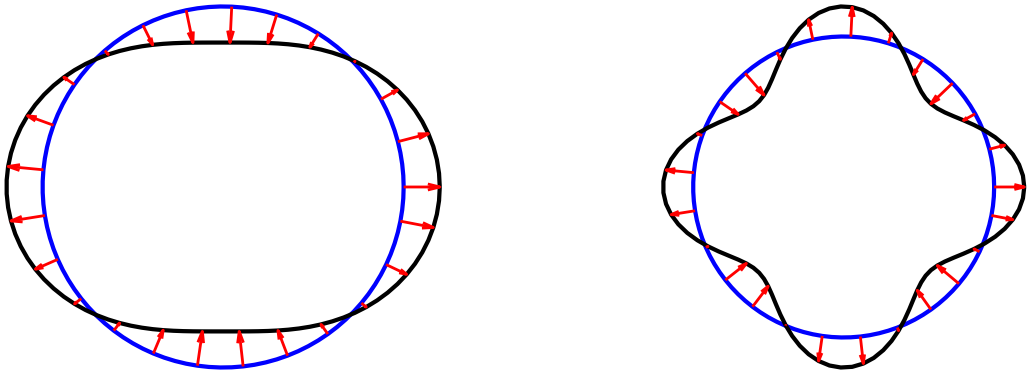
derived from the orthogonality property (2.4)

$$\begin{aligned}\langle \Delta_\ell^m \rangle &= \frac{1}{4\pi} \iint_{S^2} c_\ell^m Y_\ell^m(\theta,\phi) dS \\ &= \frac{c_\ell^m}{\sqrt{4\pi}} \iint_{S^2} \frac{1}{\sqrt{4\pi}} Y_\ell^m(\theta,\phi) dS \\ &= \frac{c_\ell^m}{\sqrt{4\pi}} \langle Y_0^0, Y_\ell^m \rangle_{S^2} \\ &= 0.\end{aligned} \tag{2.28}$$

By analogy, one can view the initial sphere as a ball of clay, that one can perturb in a sequence of steps. One cannot strictly compress or stretch the clay so both must be done equally.

> **Clay modeling analogy:** *Finding the SH representation of a particle is akin to modeling clay. One starts with a initial sphere of clay $c_0^0 Y_0^0$ with radius $\langle r \rangle$. One then applies radial perturbations Δ_ℓ^m on the clay to model the particle. As more modes are added, the amount of possible manipulations increase so one has more finesse over the final result.*

A more formal interpretation is that the SH correspond to directions of variation of zero net change in volume within the tangent space of $L^2(S^2)$ of the grain. So the lack of change in volume is only correct in an infinitesimal sense. For this reason, the analogy is not perfect



(a) (b)

Figure 2.16 The term $c_0^0 Y_0^0$ represents the initial sphere (blue) and Δ_ℓ^m represents the perturbation (red arrows), while $c_0^0 Y_0^0 + \Delta_\ell^m$ is the new perturbed shape (black). This is a simplified 2D slice of the sphere.

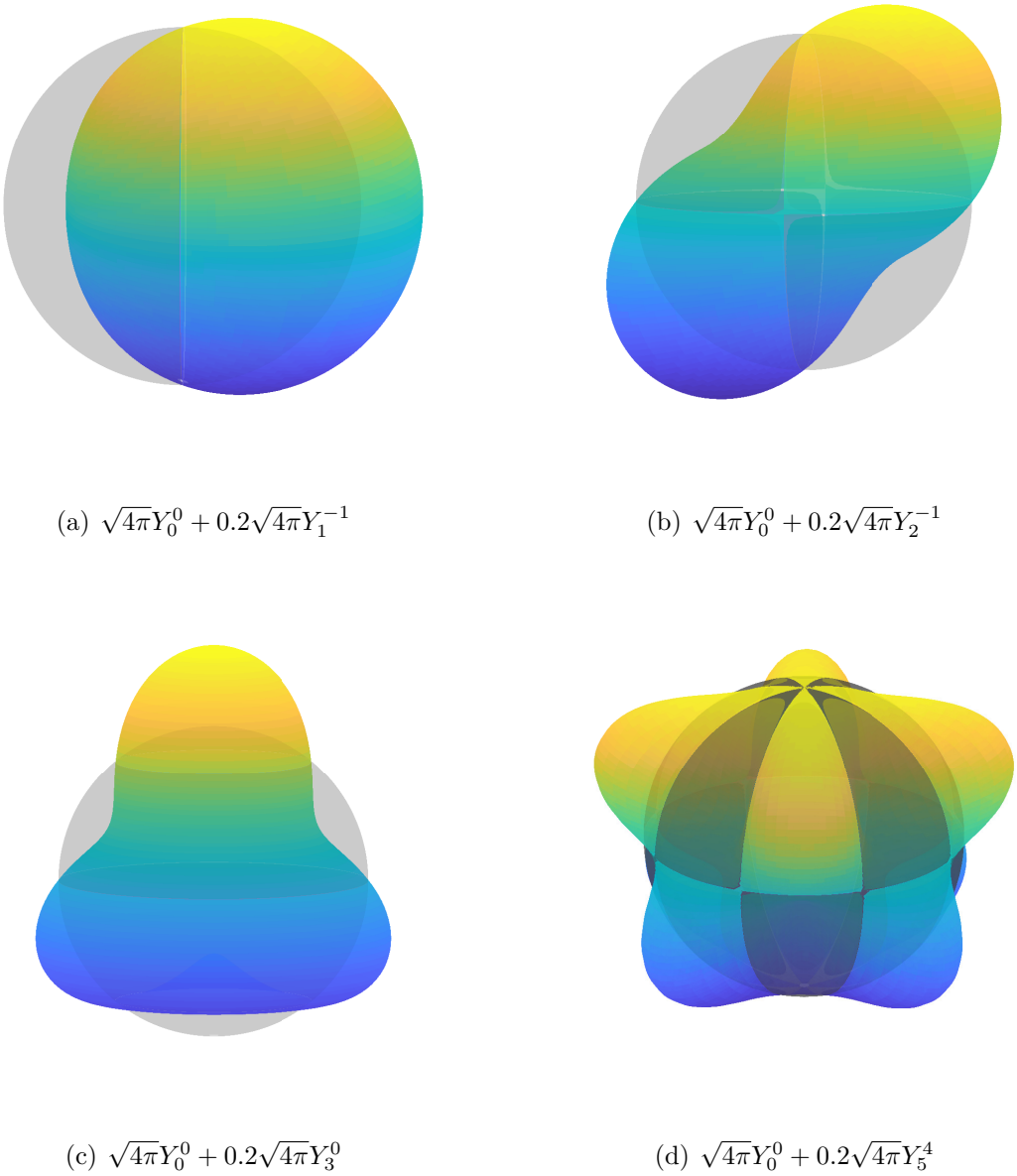
but it will nonetheless be used throughout the thesis. Now that the modes are understood as perturbations of some sphere, one must get a grasp on what the values of the coefficients c_ℓ^m actually represent. Using Parseval's equality, we get

$$\left\langle (\Delta_\ell^m)^2 \right\rangle = \frac{1}{4\pi} \iint_{S^2} (\Delta_\ell^m)^2 dS = \frac{1}{4\pi} |c_\ell^m|^2. \tag{2.29}$$

The quantity $|c_\ell^m|^2$ is therefore proportional to the mean-squared perturbation that the corresponding mode induces on the initial sphere. Taking the square root gives $\sqrt{\left\langle (\Delta_\ell^m)^2 \right\rangle} \propto |c_\ell^m|$. However the square root of the mean-squared perturbation is hard to interpret. A more interpretable value would be the mean absolute perturbation $\left\langle |\Delta_\ell^m| \right\rangle$. Jensen's Inequality [46, page 66] states that $\sqrt{\left\langle (\Delta_\ell^m)^2 \right\rangle} \geq \left\langle |\Delta_\ell^m| \right\rangle$, so the following holds true

$$|c_\ell^m| = \sqrt{4\pi} \sqrt{\left\langle (\Delta_\ell^m)^2 \right\rangle} \geq \sqrt{4\pi} \left\langle |\Delta_\ell^m| \right\rangle. \tag{2.30}$$

This relation is key, as it shows that the coefficient c_ℓ^m is the upper bound of a term with geometrical meaning. Figure 2.17 shows some perturbations on the unit sphere Δ_ℓ^m where c_ℓ^m is fixed to $0.2\sqrt{4\pi}$, ensuring that the average absolute perturbation is not higher than 0.2. The unit sphere $\sqrt{4\pi} Y_0^0$ is represented in grey and the perturbations must be interpreted as the differences between the grey and colored surfaces. Visualizing SH this way is a lot more intuitive than the representations shown in Figure 2.7. Most notably, negative and



(a) $\sqrt{4\pi}Y_0^0 + 0.2\sqrt{4\pi}Y_1^{-1}$

(b) $\sqrt{4\pi}Y_0^0 + 0.2\sqrt{4\pi}Y_2^{-1}$

(c) $\sqrt{4\pi}Y_0^0 + 0.2\sqrt{4\pi}Y_3^0$

(d) $\sqrt{4\pi}Y_0^0 + 0.2\sqrt{4\pi}Y_5^4$

Figure 2.17 Visualization of $c_\ell^m Y_\ell^m$ with $m, \ell \neq 0$ as perturbations of the unit sphere $\sqrt{4\pi}Y_0^0$. The colormap has no relation to the perturbation and is only included to increase the contrast with the sphere.

positive values clearly indicate wether the perturbation compresses or stretches the unit sphere, respectively.

Instead of considering the perturbation induced by a single mode, we can also study the

perturbation induced by a set of modes

$$\Delta_{1:\ell_{\max}} := \sum_{\ell=1}^{\ell_{max}} \sum_{m=-\ell}^{\ell} c_\ell^m Y_l^m(\theta, \phi). \tag{2.31}$$

By Parseval's equality we get

$$\left\langle \left(\Delta_{1:\ell_{\max}}\right)^2 \right\rangle = \frac{1}{4\pi} \iint_{S^2} \left(\Delta_{1:\ell_{\max}}\right)^2 dS = \frac{1}{4\pi} \sum_{\ell=1}^{\ell_{\max}} \sum_{m=\ell}^{\ell} |c_\ell^m|^2 = \frac{1}{4\pi} \sum_{\ell=1}^{\ell_{\max}} E(\ell)^2. \tag{2.32}$$

Equation (2.32) explains that the energies are related to the mean-squared perturbations $\left\langle \left(\Delta_{1:\ell_{\max}}\right)^2 \right\rangle$, which are geometrically meaningful.

2.3 Preliminary Results

Left for clarification are the choices of $\ell_{\max}$, ℓ_0, N_e, and N_g that were used when computing the SH representations of particles. In the literature, the parameter $\ell_{\max}$ is usually set heuristically between 12 and 15 [17–19, 24–26]. Another approach is to adapt the value $\ell_{\max}$ for each particle using a quality criterion based on the integral of the Gauss curvature of the SH representations [13, 20]. Though more rigorous, an adaptive $\ell_{\max}$ does not yield a consistent representation of particles unless one pads-out the missing coefficients with zeros. For this reason, we decide to fix the value $\ell_{\max}$ a priori for all particles. As stated previously, the choice of $\ell_{\max}$ should be driven by studying the most extreme particles of a dataset, since these are the ones where the SH coefficients decrease at the slowest rate. Early experimentation suggests that $\ell_{\max} = 20$ is a reasonable choice. Analyzing the curve in Figure 2.15 corresponding to the particle with the least convexity and smoothness (grain3), we observe that $\ell_{\max} = 20$ is the point where the perturbations induced by the spherical harmonics become two orders of magnitude smaller than the average radius of the particle. At this value of ℓ, one can assume that the spherical harmonics simply adjust the texture of the particle. Even though setting $\ell_{\max} = 20$ is maybe an overkill to faithfully represent the average particles from a dataset, we argue that it is safer to use more coefficients than necessary rather than fewer. Our argument is that the dimensionality of the data will be eventually reduced through the use of the Principal Component Analysis, see Section 4.2.

The choice of N_e and N_g essentially depends on $\ell_{\max}$ since high frequency functions are most susceptible to quadrature errors. The Gauss quadratures seen in the literature typically use from 14,000 to 60,000 Gauss points sampled in the $\theta - \phi$ plane. However, our quadrature

scheme is slightly different so the choice of N_e and N_g should be based on our own results. Looking at the curves corresponding to the uniform mesh in Figures 2.11, 2.12, and 2.13, we observe that setting $N_e = 3,000$ and $N_g = 9$ yields absolute errors of 10^{-4} and 10^{-15} on all three spherical harmonics integrals. For extra security measure, the number of elements is set to an even larger value $N_e = 5,000$. This choice leads to a total of $N_e \times N_g = 45,000$ Gauss points sampled on the unit sphere, which is not as extreme as some of the values seen in the literature.

Finally, the factor ℓ_0 at which the Lanczos filter starts being applied is set to 10. Our justification is based on one of the results in [16], where the authors demonstrate that the filtered SH representation of a prolate ellipsoid degrades rapidly in accuracy when considering $\ell_0 < 10$. They also explain that applying the filter at $\ell_0 = 10$ can drastically reduce the ripples seen on the faces of a smooth cube.

This chapter concludes by showing the STM and SH representations of four arbitrary particles. The values of $\ell_{\max}$, ℓ_0, N_e, and N_g discussed above are used. Looking at Figures 2.18 and 2.19, it appears that the SH coefficients are able to estimate reasonably well the shapes of all four particles, even non-convex ones. The major difference between the STM and unfiltered SH representations is the presence of ripples on the surface of the SH representations. These are artefacts of both truncation and Gibbs phenomenon. Particles which are highly non-convex encounter the risk of being non-star shaped. As seen in Figure 2.9(b), trying to represent non star-shaped particles in terms of a radial function $r(\theta, \phi)$ can induce some irregularities in their linear interpolation. These abrupt variations can potentially amplify the Gibbs phenomenon. Another major difference between the STM and SH representations is that the latter tends to have smoother corners than the former. This is due to the truncation of the SH expansion which prevents the spherical harmonics from representing such fine details.

Differences between the SH representations with and without filter are also visible. The filter clearly has the beneficial effect of reducing the ripples but it has the side effect of smoothing the corners even more. Because of its positive effect on surface ripples, the Lanczos sigma factor shall be consistently applied, unless specified otherwise.

2.4 Summary

In this chapter, we have explained how to compute the SH representations of real particles. The STM representation was obtained from micro-computed tomography, while the SH representation was calculated from the STM one using linear interpolation and numerical

integration. Our specific contributions to the subject include the followings

1. An in-depth investigation of multiple spherical quadratures for computing the SH coefficients;
2. An intuitive geometrical analogy of the SH based on clay modeling;
3. An explanation of the values of the SH coefficients based on mean-squared perturbations they induce on the sphere.

We describe in the next chapter the calculations of classical particle descriptors from their STM and SH representations.

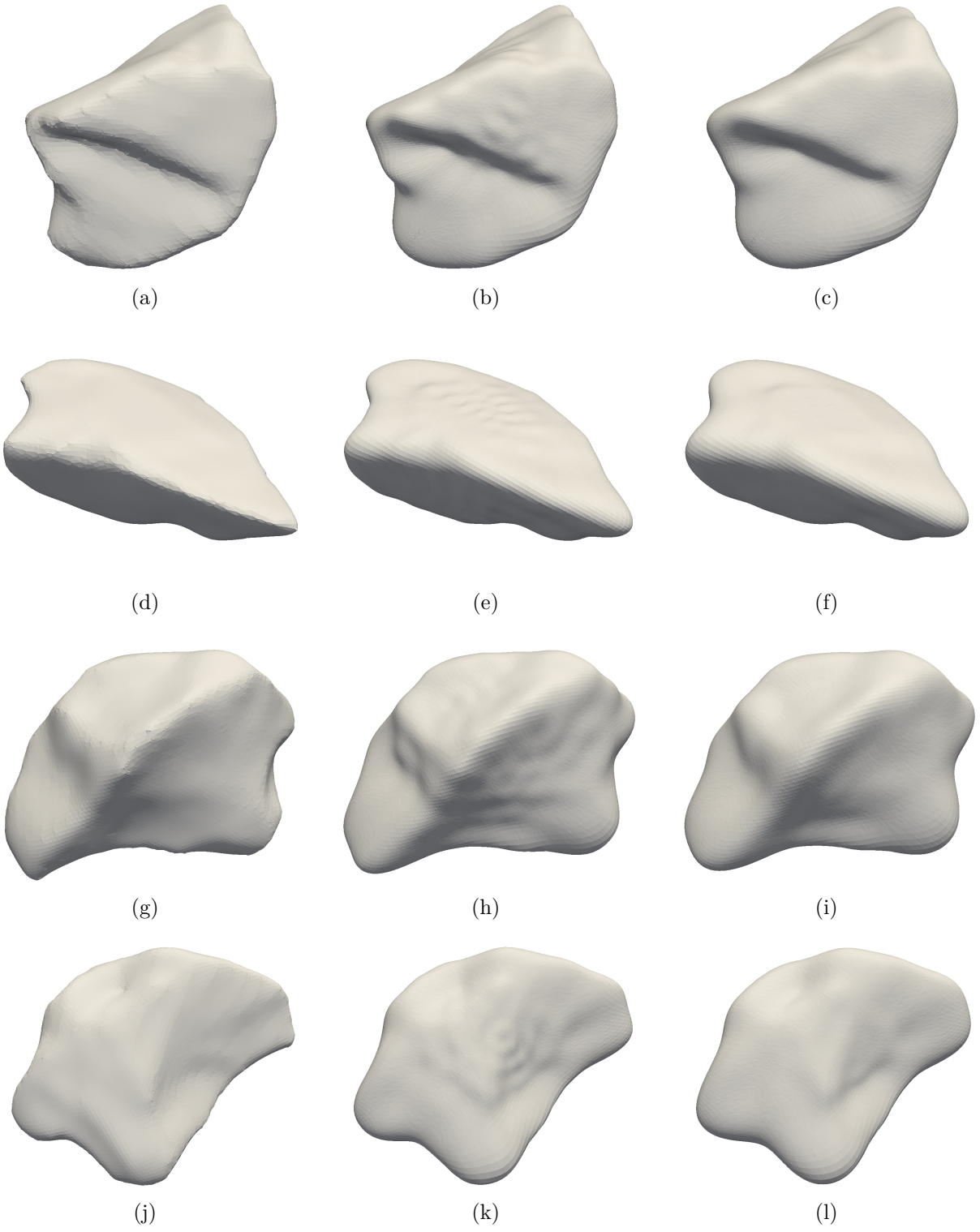
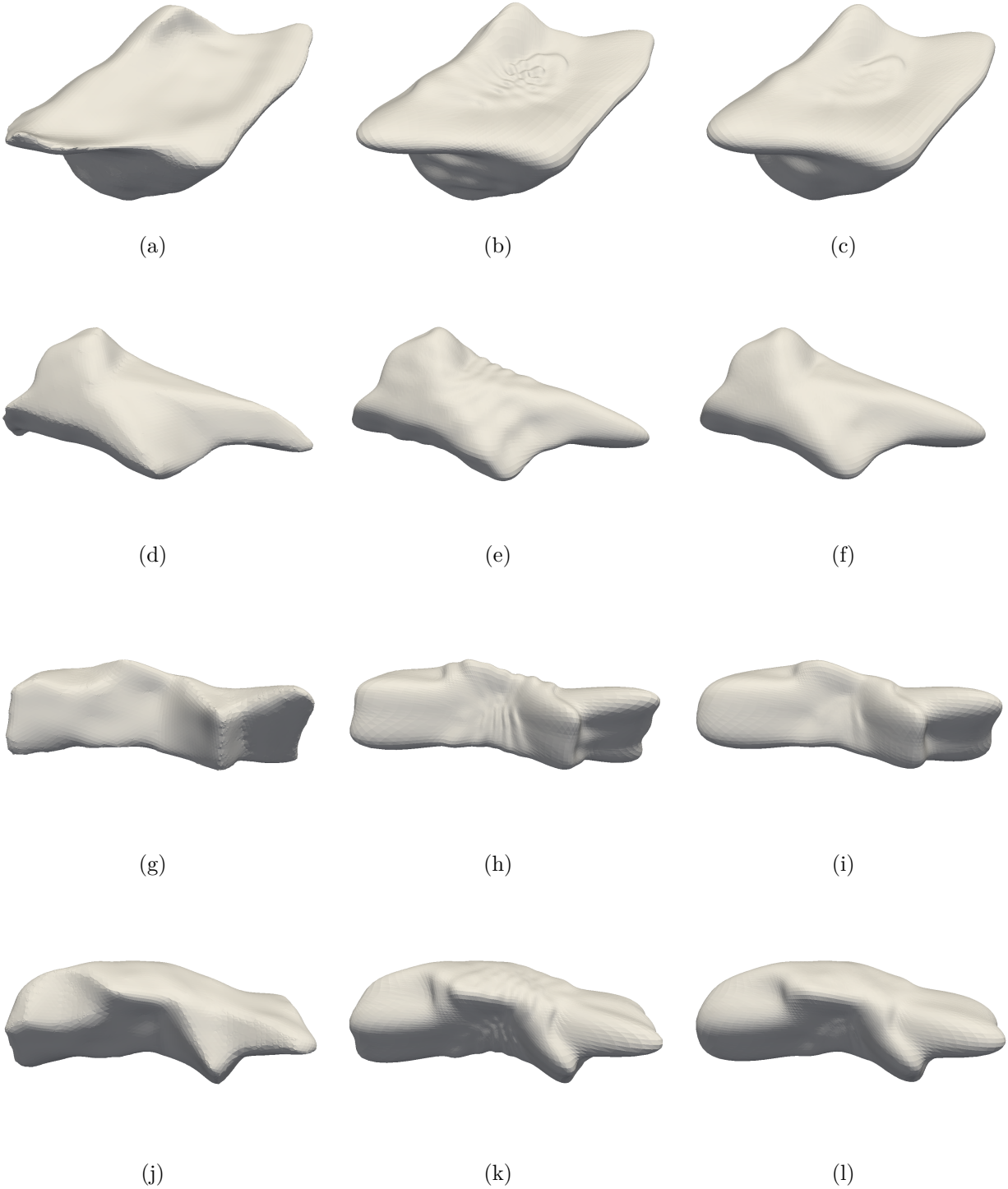


Figure 2.18 STM representation (left column), SH representation (middle column), and filtered SH representation (right column) of a set of particles.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 2.19 STM representation (left column), SH representation (middle column), and filtered SH representation (right column) of additional particles.

CHAPTER 3 GEOMETRICAL CHARACTERIZATION

The main objective of this research is to develop a general process that takes real particles and generate virtual ones that are geometrically similar, see Figure 1.2. Before describing the steps of the generative process which follow the discretization of particles, it is primordial to take the time to define rigorous notions of geometrical ressemblance. This chapter focuses on defining the classical shape descriptors found in the geology literature, which can be used to make meaningful comparisons between particles. These quantities of interest include volume, aspect ratios, sphericity, convexity, and roundness. Elementary geometry results are first discussed. The calculation of the shape descriptors in both STM and SH representations is then described in detail. The chapter concludes by showing convergence studies on various manufactured star-shaped particles and one asphalt particle.

3.1 Preliminary Results of Differential Geometry

This section summarizes those elementary results from differential geometry necessary to define the shape descriptors. Readers who are already familiar with the subject can safely skip to the next section.

3.1.1 Surface and Volume

Let $G \subset \mathbb{R}^3$ be a particle. The surface area of G is defined as

$$S = \oiint_{\partial G} dS, \tag{3.1}$$

where ∂G is the boundary of G, i.e. the surface of the particle. When a particle is described in the STM representation, the integral becomes

$$S_{\text{STM}} = \sum_{i=1}^{N_f} \iint dS_i = \sum_{i=1}^{N_f} A_i, \tag{3.2}$$

where dS_i represent the surface element of the ith simplex so that A_i represents the area of the ith simplex. In the SH representation, the integral, expressed in spherical coordinates, becomes [13]

$$S_{\text{SH}} = \int_0^{2\pi} \int_0^{\pi} r\sqrt{r_\theta^2 + (r^2 + r_\phi^2)\sin^2\phi}\; d\phi d\theta. \tag{3.3}$$

The derivatives r_ϕ and r_θ are computed from a linear combination of the derivatives of the spherical harmonics. As a reminder, the derivatives of the SH are listed in Appendix B. The volume of G is defined as

$$V = \int_G dV. \tag{3.4}$$

Since the STM representation provides only a description of the surface of the particles, the volume integral must be transformed into a surface integral via the divergence theorem

$$V_{\mathrm{STM}} = \int_G dV = \int_G \nabla \cdot (x\boldsymbol{i}) dV = \oiint_{\partial G} x\boldsymbol{i} \cdot \boldsymbol{n} dS = \sum_{i=1}^{N_f} \iint_{S_i} x\mathbf{N}(i,1) dS_i. \tag{3.5}$$

Using the SH representation, the integral over the volume has the following convenient simplification

$$V_{\mathrm{SH}} = \int_0^{2\pi} \int_0^{\pi} \int_0^{r(\theta,\phi)} r^2 \sin\phi \, dr d\phi d\theta = \frac{1}{3} \int_0^{2\pi} \int_0^{\pi} r(\theta,\phi)^3 \sin\phi \, d\phi d\theta. \tag{3.6}$$

3.1.2 Inertia and Semi-Axes

The inertia tensor $\mathbf{I}$ is an invariant which measures the resistance of a particle to rotate around certain axes. It is possible to show that in each frame of reference, the inertia tensor $\mathbf{I}$ can be represented by a symmetric positive definite matrix

$$\mathbf{I} = \begin{bmatrix} \int_G \Big((y-\bar{y})^2 + (z-\bar{z})^2\Big) dV & -\int_G (x-\bar{x})(y-\bar{y}) dV & -\int_G (x-\bar{x})(z-\bar{z}) dV \\ * & \int_G \Big((x-\bar{x})^2 + (z-\bar{z})^2\Big) dV & -\int_G (y-\bar{y})(z-\bar{z}) dV \\ * & * & \int_G \Big((x-\bar{x})^2 + (y-\bar{y})^2\Big) dV \end{bmatrix} \tag{3.7}$$

where the first moments

$$\bar{x} = \frac{1}{V} \int_G x \, dV, \qquad \bar{y} = \frac{1}{V} \int_G y \, dV, \qquad \bar{z} = \frac{1}{V} \int_G z \, dV, \tag{3.8}$$

define the center of mass. The orthonormal eigenvectors of the inertia tensor are referred to as the *principal axes* and the eigenvalues are called the inertias

$$\mathbf{I}\, \boldsymbol{p}_i = \mathrm{I}_i \, \boldsymbol{p}_i, \quad i = 1,2,3, \tag{3.9}$$

which are typically ordered as $\mathrm{I}_1 > \mathrm{I}_2 > \mathrm{I}_3$. The principal axes are stored as the columns of

the matrix

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \boldsymbol{p}_3 \end{bmatrix}. \tag{3.10}$$

The transpose of $\mathbf{P}$ is an orthogonal transformation, i.e. an isometry, which takes vectors in the original frame of reference and expresses them in the frame of the principal axes. In case $\mathbf{P}$ has a negative determinant, one can change the sign of one of the columns to ensure that $\boldsymbol{p}_1 \times \boldsymbol{p}_2 = \boldsymbol{p}_3$. Using the matrix $\mathbf{P}$ and the center of mass, the following change of basis is introduced

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{P}^T \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \\ z - \bar{z} \end{bmatrix}, \tag{3.11}$$

which puts the origin at the center of mass and orients the axes with the principal axes. This new frame of reference is called the principal frame and is illustrated in Figure 3.1.

Referring back to Section 2.2.2, the spherical harmonics are computed in the principal frame since angles θ and ϕ are consistently calculated with respect to the principal axes.

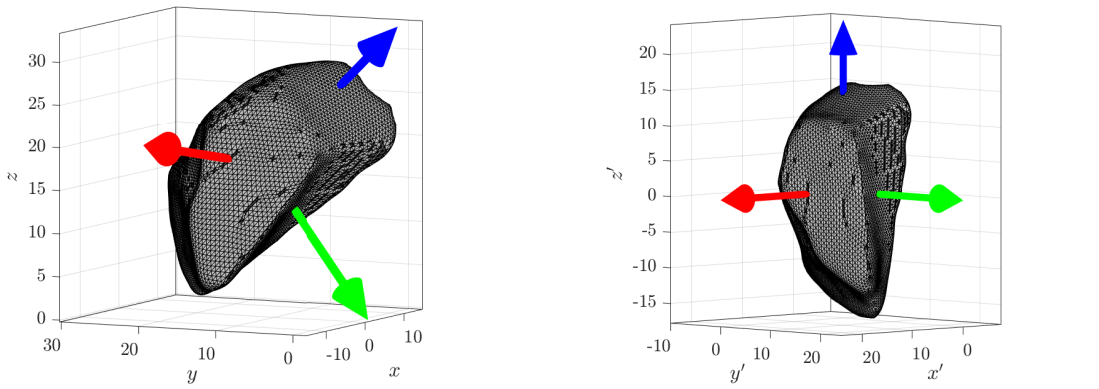
The principal axes and their associated eigenvalues allow us to associate to each grain a unique *equivalent ellipsoid* with the same center of mass and volume as the particle and whose inertias are proportional to I_1, I_2, and I_3. An ellipsoid satisfies the implicit relation

$$\left(\frac{x'}{a}\right)^2 + \left(\frac{y'}{b}\right)^2 + \left(\frac{z'}{c}\right)^2 = 1, \tag{3.12}$$

where $a < b < c$. The scalars a, b, and c are called the semi-axes of the ellipsoid. Note that this is not the usual convention $a > b > c$. The reason why the former has been chosen is that visualization of a set of particles is made easier by aligning the longest axis of a particle with the z axis. The unknown a, b, and c can be found by the solving a non-linear system of equations that requires the calculation of the inertia tensor and the volume, which can be done in the STM or SH representations.

3.1.3 Curvature

One of the most widely used shape descriptors is based on the surface curvature of the grains, and so we take the time to define this notion carefully. This section begins by studying the curvature of a single parametrized curve, before discussing curvature of surfaces in general. The presentation given here is novel but the interested reader can find additional information in [47].



(a) Particle in the original basis

(b) Particle in its principal frame $I_1 > I_2 > I_3$

Figure 3.1 Change of basis into the particle principle frame.

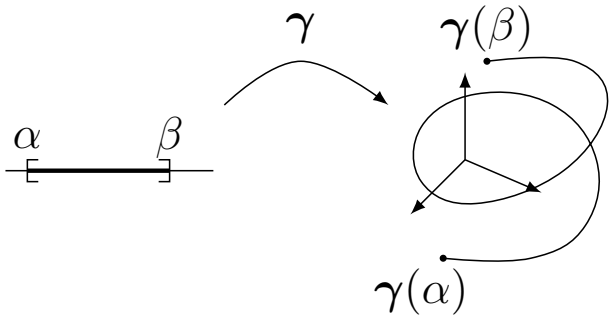


Figure 3.2 Example of parametric curve.

The simplest definition of curvature requires the use of parametrized curves, which are defined as maps $\boldsymbol{\gamma} : \mathbb{R} \supset [\alpha, \beta] \to \mathbb{R}^3$. An example of a parametric curve is shown in Figure 3.2. To define the curvature at a given point $\boldsymbol{\gamma}(s)$, one must calculate the tangent circle which is the unique circle with the same second order Taylor expansion around $\boldsymbol{\gamma}(s)$ as $\boldsymbol{\gamma}$. The curvature κ at $\boldsymbol{\gamma}(s)$ is then defined as the inverse radius of the tangent circle, see Figure 3.3.

Curvature of surfaces is an extension of the concept of curvature of parametric curves. Let $\mathcal{S}$ be a smooth surface and let $\boldsymbol{p}$ be a point on the surface. One can define the tangent plane $T_{\boldsymbol{p}}$ as the first order taylor expansion of the surface around $\boldsymbol{p}$ and $\boldsymbol{n}$ as a unit normal to the plane, see Figure 3.4. By selecting an arbitrary vector $\boldsymbol{t} \in T_{\boldsymbol{p}}$, one can define a normal plane $N_{\boldsymbol{p},\boldsymbol{t}}$ as the unique plane that contains $\boldsymbol{p}$ and is spanned by the vectors $\boldsymbol{t}$ and $\boldsymbol{n}$. The intersection between this normal plane and the surface $N_{\boldsymbol{p},\boldsymbol{t}} \cap \mathcal{S}$ generates a curve on the surface, as seen in Figure 3.4. The curvature of such curves were previously defined and so we define the *surface curvature* at $\boldsymbol{p}$ to be the function $\kappa_{\boldsymbol{p}} : T_{\boldsymbol{p}} \to \mathbb{R}$ that associate to each

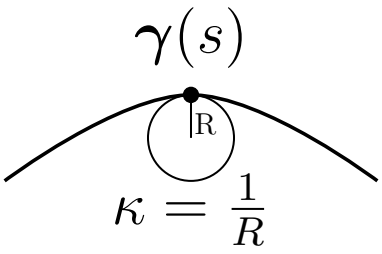


Figure 3.3 The unique circle which approximates the curve to the second order around $\boldsymbol{\gamma}(s)$ defines the curvature κ at $\boldsymbol{\gamma}(s)$.

tangent direction $\boldsymbol{t}$, the curvature $\kappa_{\boldsymbol{p}}(\boldsymbol{t})$ of the curve $N_{\boldsymbol{p},\boldsymbol{t}} \cap S$.

A fundamental result of differential geometry is that, at $\boldsymbol{p}$, these exist two unit vectors $\boldsymbol{w}_1, \boldsymbol{w}_2 \in T_{\boldsymbol{p}}$ and two scalars $\kappa_1 < \kappa_2$ such that

$$\kappa_{\boldsymbol{p}}(\boldsymbol{w}_i) = \kappa_i, \tag{3.13}$$

and

$$\boldsymbol{t} = \cos\theta \boldsymbol{w}_1 + \sin\theta \boldsymbol{w}_2 \in T_{\boldsymbol{p}} \implies \kappa_{\boldsymbol{p}}(\boldsymbol{t}) = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta. \tag{3.14}$$

This allows the computation of curvature in any direction. The two curvatures κ_1 and κ_2 are called the *principal curvatures* while the two directions $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are called the principal directions. One can define two geometric invariants from the principal curvatures

$$H = \frac{\kappa_1 + \kappa_2}{2}, \qquad K = \kappa_1 \kappa_2, \tag{3.15}$$

which are called the Mean and Gaussian curvatures, respectively.

We can compute the quantities κ_1, κ_2, H, K on all particles in all our samples using both STM and SH representations. Various techniques exist to estimate the curvature of triangulated surfaces and their technical details shall not be discussed in this thesis so we refer the interested reader to the following ressources. [48–50]. With the SH representation, one can compute the curvature analytically without requiring any approximation, but the formulas are quite complicated. In the appendix of [13], Garboczi collected all the formulas to compute the curvatures of any a smooth function $r(\theta, \phi)$ in spherical coordinates. Those formulas are reproduced in Appendix B.

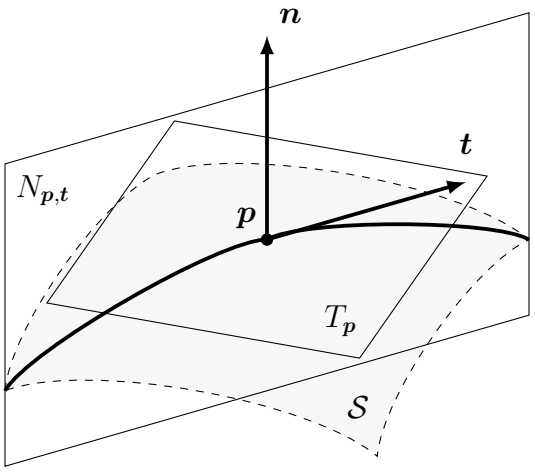


Figure 3.4 The vectors $\boldsymbol{t} \in T_{\boldsymbol{p}}$ and $\boldsymbol{n}$ define a normal plane $N_{\boldsymbol{p},\boldsymbol{t}}$ whose intersection with $\mathcal{S}$ defines a curve.

3.2 Classical Shape Descriptors

In this section, the geometrical quantities described in Section 3.1 are used to define the normalized scale-invariant classical shape descriptors from the geology literature [12, 16]. By normalized, it is meant that their value ranges from 0 to 1 and by scale-invariant, that they do not change when scaling a particle by some amount. These two properties are essential to make meaningful comparisons between particles, independently of their sizes.

3.2.1 Elongation and Flatness Indexes

The semi-axes a, b, c of the equivalent ellipsoid can be made both scale-invariant and normalized by taking their ratio, which yields the elongation and flatness indexes

$$\mathrm{EI} = \frac{b}{c}, \qquad \mathrm{FI} = \frac{a}{b}. \tag{3.16}$$

Recall that $a < b < c$, so these quantities are both smaller than one. Those indices give an overall idea of the shape of the particle. The authors in [12] use these two values to define an informal classification into four shapes: spheroids, prolates, oblates, and blades. Figure 3.5 illustrates this widely used classification.

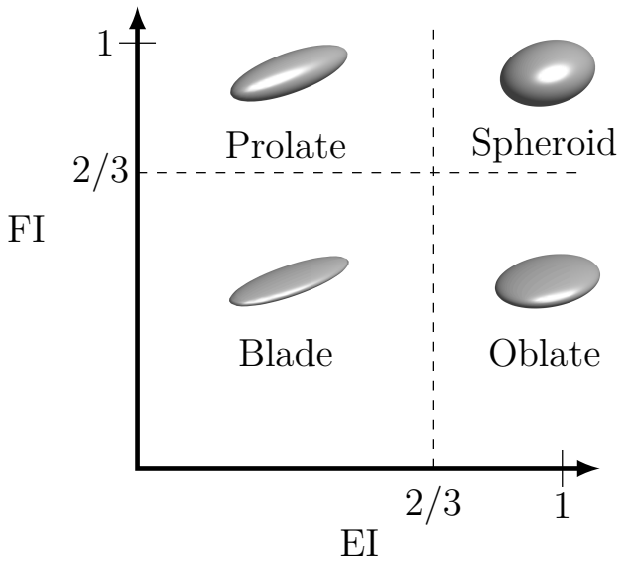


Figure 3.5 Four classes of ellipsoids based on their elongation (EI) and flatness (FI).

3.2.2 Sphericity

Sphericity naturally appears when one attempts to take the surface information and make it scale-invariant. To do so, notice that all shapes in 3D space respect the power law

$$S = \alpha \sqrt[3]{V^2}, \tag{3.17}$$

where the scaling factor α only depends on the shape of the particle. This factor α contains the intrinsic surface information. The isoperimetric inequality [47] states that the smallest possible value of the scaling factor is $\alpha_{\min} = \sqrt[3]{36\pi}$ which is only achieved in the case of spheres. This suggests defining a quantity called the *sphericity* [16]

$$\zeta = \frac{\alpha_{\min}}{\alpha} = \frac{\sqrt[3]{36\pi V^2}}{S}, \tag{3.18}$$

which is a measure of the equiaxiality of a particle. Equiaxiality refers to the property that lengths of particles do not vary widely with respect to directions. We refer the reader to refer to Table 1 in [16] for some insightful examples.

3.2.3 Roundness

The curvature of a sphere is the inverse of its radius, hence doubling a particles size halves its curvature. In order to obtain a scale-invariant measure, we consider the ratio with respect to

a reference curvature which is typically chosen to be the inverse radius of the largest inscribed sphere κ_{is} [12, 16]. The justification for using the largest inscribed sphere is that it allows to define corners of a particle as

$$\triangle := \{\boldsymbol{p} \in \partial G \,|\, |\kappa_2(\boldsymbol{p})| \geq \kappa_{\text{is}}\}. \tag{3.19}$$

First introduced by Wadell while studying 2D slices of particles [9], the *roundness* was later extended to 3D particles. Its most recent definition is [16]

$$R = \frac{\kappa_{is} \iint_{\triangle} |\kappa_2(\boldsymbol{p})|^{-1} dS}{\iint_{\triangle} dS}, \tag{3.20}$$

To compute roundness, one basically takes the average of $\kappa_{is}/|\kappa_2|$ over all corners of the particle. Sharper corners result in higher $|\kappa_2|$ which yields a smaller roundness factor. This quantity is normalized, scale-invariant, and reaches the value of one for spheres. Note that sphericity and roundness are different characterizations, see Figure 3.6. From the crooked look of the graph, it is implied that the sphericity and roundness are slightly correlated although they measure different morphological characteristics. This is because smoothing corners tends to make particles more equiaxed.

3.2.4 Convexity

The final classical shape descriptor is convexity which is defined as

$$C = \frac{V_{\text{CH}}}{V}, \tag{3.21}$$

where V_{CH} is the volume of the convex hull of the particle. Convexity is straightforward to interpret and reaches the value of one when G is a convex set. It can be computed for both STM and SH representations using the Matlab native function `convhull` [51].

3.3 Manufactured Star-Shaped Particles

In order to verify the implementation of the previous formulas, one needs to consider manufactured star-shaped particles for which the exact values of some of the geometrical quantities of interest are known. The Lanczos filter shall not be applied to the SH coefficients of said particles since its effect on particle geometry is not yet fully understood.

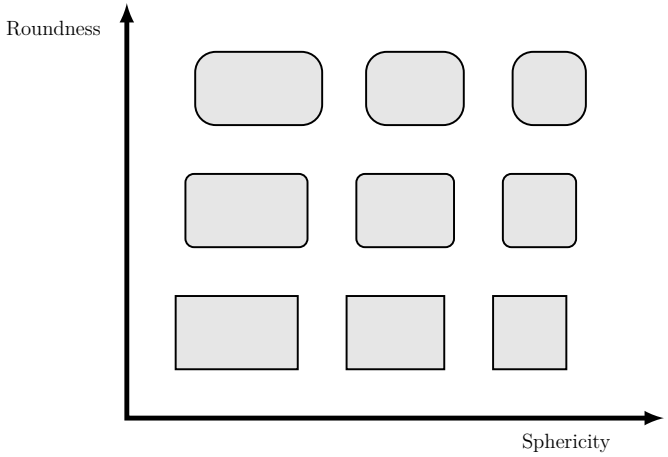


Figure 3.6 Intuitive difference between Sphericity and Roundness. Sphericity measures equiaxiality while roundness measures the sharpness of corners.

3.3.1 Revolution Ellipsoid

A revolution ellipsoid is a special case of the ellipsoid where $a = b$ and $a < c$. Its implicit equation becomes

$$\frac{x^2 + y^2}{a^2} + \frac{z^2}{c^2} = 1. \tag{3.22}$$

The parametrization in spherical coordinates is given by

$$r(\theta, \phi) = \frac{a^2 c}{\sqrt{a^2 c^2 \sin^2 \phi + a^2 \cos^2 \phi}}, \tag{3.23}$$

which is used to compute the SH coefficients without requiring interpolation. This shape is of interest because it is more complicated than a sphere but its geometrical features can be

computed analytically [52, 53]

$$
\begin{aligned}
S &= 2\pi a^2 \left(1 + \frac{c}{a}\frac{\arcsin\sqrt{1-(\frac{a}{c})^2}}{\sqrt{1-(\frac{a}{c})^2}}\right), \\
V &= \frac{4\pi}{3}a^2 c, \\
H &= \frac{r^2 - 2a^2 - c^2}{2a^4c^2 \left(\frac{x^2}{a^4} + \frac{y^2}{a^4} + \frac{z^2}{c^4}\right)^{\frac{3}{2}}}, \\
K &= \frac{1}{a^4c^2 \left(\frac{x^2}{a^4} + \frac{y^2}{a^4} + \frac{z^2}{c^4}\right)^{2}}.
\end{aligned}
\tag{3.24}
$$

To obtain the curvatures H and K as a function of θ and ϕ, one must replace x, y, z by their expression in spherical coordinates (2.13). In the following convergence study, we compute the geometrical descriptors in the SH representation on a revolution ellipsoid with $a = 1$ and $c = 2$. A mesh of 5,000 elements with $9 = 3 \times 3$ Gauss points per element is used to evaluate the integrals. Let Θ be the true value of a parameter and Θ_{SH} be its approximation using the SH representation, the absolute and relative errors are defined as

$$
\begin{aligned}
\Delta\Theta &= |\Theta - \Theta_{SH}|, \\
\Delta_*\Theta &= \frac{\Delta\Theta}{\Theta}.
\end{aligned}
\tag{3.25}
$$

Figure 3.7 illustrates the convergence of the relative errors of the geometrical descriptors. Note that the mean operator $\langle \cdot \rangle$ from (2.26) is employed here. This was done to yield a single scalar value since curvatures are functions over the surface of the particle. It is observed that the geometrical properties calculated with the SH representation converge to the true values of the revolution ellipsoid. However, it appears the curvatures converge more slowly than other shape descriptors.

3.3.2 Smooth Cubes

Another informative class of star-shaped particles are the smooth cubes defined by the implicit relation

$$
|x|^{2/\epsilon} + |y|^{2/\epsilon} + |z|^{2/\epsilon} = 1, \tag{3.26}
$$

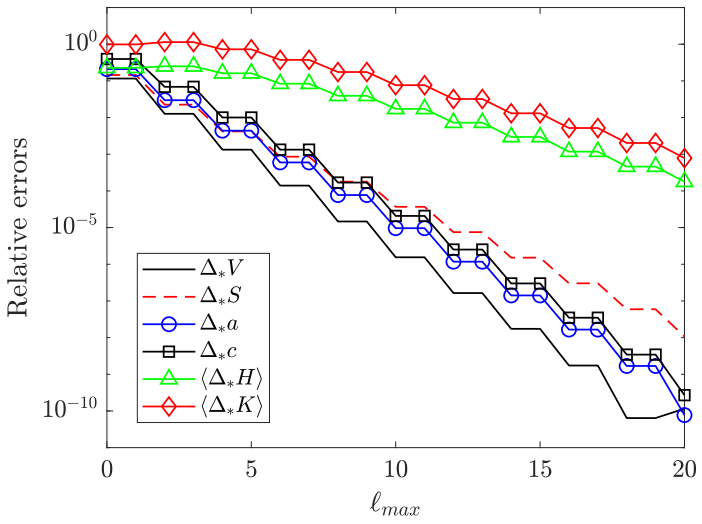


Figure 3.7 Relative errors with respect to $\ell_{\max}$ in geometrical quantities associated with a revolution ellipsoid ($a = 1, c = 2$).

where $\epsilon \in]0, 1]$. Such surfaces have the following expression in spherical coordinates

$$r(\theta, \phi) = \left(|\cos\theta \sin\phi|^{2/\epsilon} + |\sin\theta \sin\phi|^{2/\epsilon} + |\cos\phi|^{2/\epsilon} \right)^{-\epsilon/2}. \tag{3.27}$$

Figure 3.8 offers some insight on the effect of ϵ on the particle shape. The analytical values of some geometrical quantities of smooth cubes are known [54]

$$\begin{aligned} V &= 2\epsilon^2 \beta\left(\frac{\epsilon}{2}, \epsilon + 1\right) \beta\left(\frac{\epsilon}{2}, \frac{\epsilon}{2} + 1\right), \\ \mathrm{I}_{xx} = \mathrm{I}_{yy} = \mathrm{I}_{zz} &= \epsilon^2 \beta\left(\frac{3\epsilon}{2}, \frac{\epsilon}{2}\right) \beta\left(\frac{\epsilon}{2}, 2\epsilon + 1\right), \end{aligned} \tag{3.28}$$

where

$$\beta(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x + y)}, \tag{3.29}$$

and $\Gamma(x)$ is the well-known Gamma function introduced by Daniel Bernoulli. Moreover, the exact curvatures K and H are found by computing the first and second-order derivatives of (3.27) using the Sympy python package for symbolic mathematics. The discovered formulas will not be shown because they are too heavy to be put in the document. Figure 3.9 shows the convergence for a smooth cube with $\epsilon = 0.25$ using the same amount of mesh elements and Gauss points as for the revolution ellipsoid.

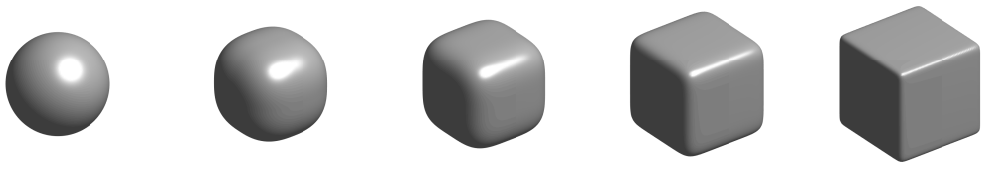


Figure 3.8 Effect of the ϵ parameter on the shape of smooth cubes.

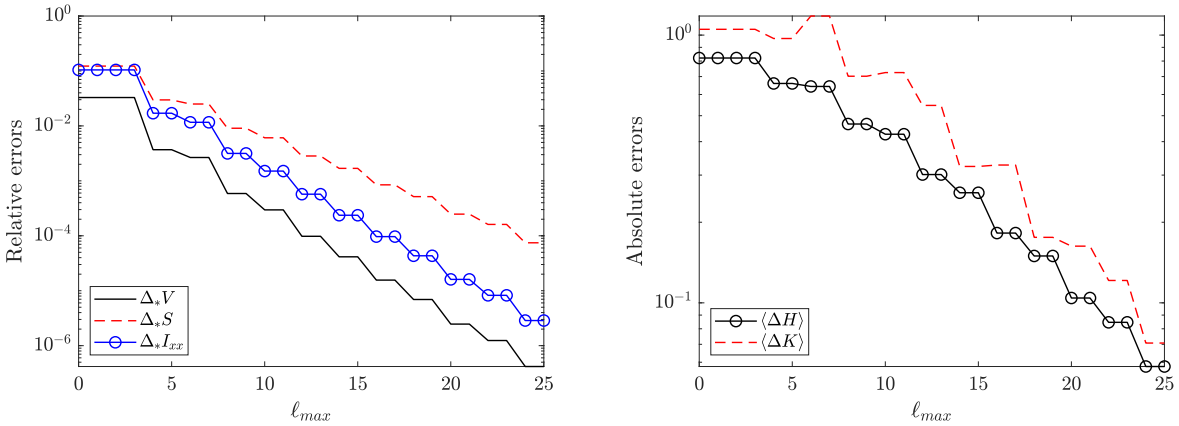
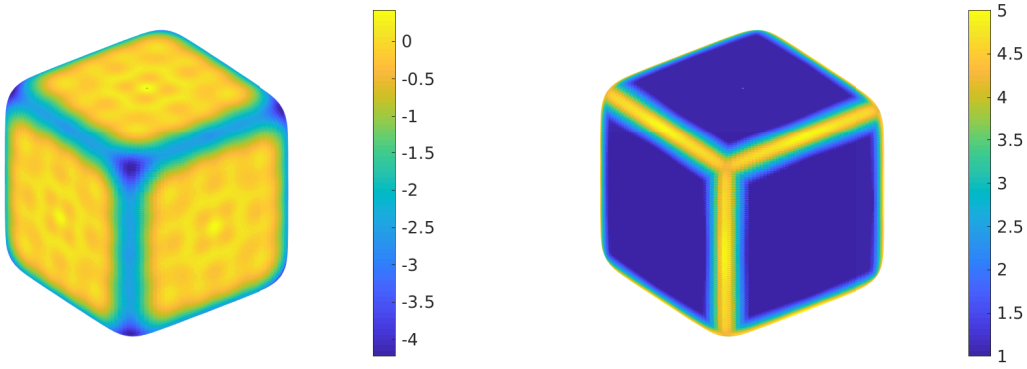


Figure 3.9 Absolute and relative errors with respect to $\ell_{\max}$ in geometrical quantities associated with a smooth cube ($\epsilon = 0.25$).

The inertias I_{yy} and I_{zz} are not plotted since they are extremely close to I_{xx}. Moreover, the absolute errors of the mean and Gauss curvatures are computed because the exact curvatures vanish on the faces of the cube, making the use of the relative error impossible. Once again, the convergence of the geometrical features in the SH representation is observed. Nevertheless, the surface area and curvatures converge more slowly than other descriptors.

The last computation that requires verification is that of roundness. The first test is to verify subjectively if the criteria $|\kappa_2| \geq \kappa_{\text{is}}$ does indeed identify corners. Figure 3.10 exhibits the results for $\ell_{\max} = 25$. Looking at Figure 3.10(a), the presence of ripples on the surface which is due to the truncation of the SH coefficients is seen. Those ripples could potentially be identified as corners if they become too large. The Figure 3.10(b) illustrates that the criterion is able to identify corners of the particles without being affected by the surface ripples.

The second verification is to approximate the roundness for different values of ϵ and verify if it decreases with ϵ. Since the curvature is known at every surface point, the roundness can be approximated with a Gauss quadrature of the integral defined by (3.19) and (3.20).



(a) Mean curvature H_{SH} (b) Colormap of $\max\{1, {}^{|\kappa_2|}/_{\kappa_{\mathrm{is}}}\}$.

Figure 3.10 Curvatures computed on the SH representation of a smooth cube with $\epsilon = 0.25$.

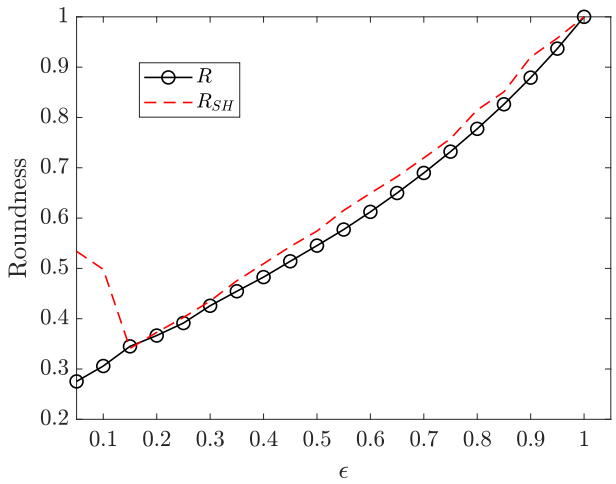


Figure 3.11 Effect of ϵ on the approximated roundness R (smooth cubes) and R_{SH} (SH representation of smooth cubes).

This process can be applied to both the analytical curvature and its estimation using the spherical harmonics. Figure 3.11 demonstrates the different approximations of roundness as a function of ϵ with $\ell_{\max} = 25$. It appears that both approximations start at unity when $\epsilon = 1$, as expected, and then shrink to zero as ϵ is reduced. When ϵ reaches 0.15, the roundness approximated with the SH representation starts increasing. This behavior is not yet fully understood and requires more investigation. Our main hypothesis is that as ϵ decreases, the cube becomes more irregular since the derivatives on its edges and corners begin to vary abruptly. This growth in irregularity could increase the surface ripples already seen in Figure 3.10(a), therefore increasing the estimate of roundness. A possible solution would be

to increase $\ell_{\max}$ or to use the Lanczos factor to remove ripples.

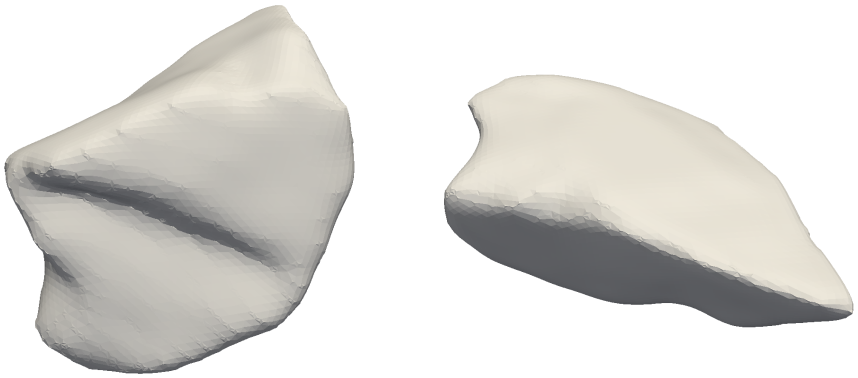
3.4 Asphalt Particle

We conclude this chapter with a brief convergence study of the geometric quantities computed with the SH representation of a real asphalt particle, see Figure 3.12. When working with real particles, the true geometrical properties are not accessible. However, the geometrical features computed in the STM representation can be treated as the ground truth. This specific grain is chosen because of its non-convexity which would induce large surface ripples on the SH representation. In order to observe the ripples, the Lanczos filter was not applied.

Figure 3.13 depicts the convergence of geometric quantities of interests. Note that the curvature error was is not treated since the technique currently used to evaluate the curvatures on STM representations has not yet been fully verified. It was found that the geometric quantities computed with the SH coefficients converge to the same quantities calculated with the STM representation. Figure 3.14 compares the largest inscribed spheres for both representations. Though the spheres are not centered at the same location inside the particle, they have a similar radius which is the most important quantity to compute the roundness. With the knowledge of the largest inscribed sphere, the corners of the particle and the roundness factor can be identified, see Figure 3.15. We observe that the ripples on the surface of the SH representation are also identified as corners. This error could potentially bias the SH roundness downward. Moreover, it appears that the actual edges in the SH representation are more round since the relative curvature $|\kappa_2|/\kappa_{\text{is}}$ reaches a maximal value of 20 for the STM representation and only 14 for the SH representation. This phenomenon could bias the SH roundness upward. Because of these two opposite effects, one should always be cautious when analyzing the roundness computed on SH representations of particles.

3.5 Summary

In this chapter, we have defined classical shape descriptors and have shown how these are commonly computed using both the STM and SH representations. Moreover, we have conducted a convergence analysis of the SH representation using a revolution ellipsoid, a smooth cube, and an asphalt particle. Classical shape descriptors will be of importance later when comparing the real particles to the virtual grains generated by the statistical model, see Figure 1.2. In the next chapter, we describe the remaining modeling steps that are needed to develop a generative process used to create virtual particles from real ones.



(a) (b)

Figure 3.12 Views of the asphalt particle under study.

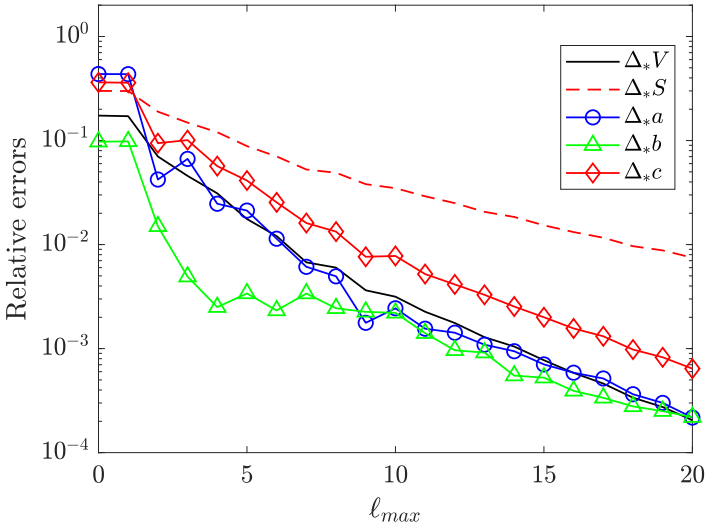
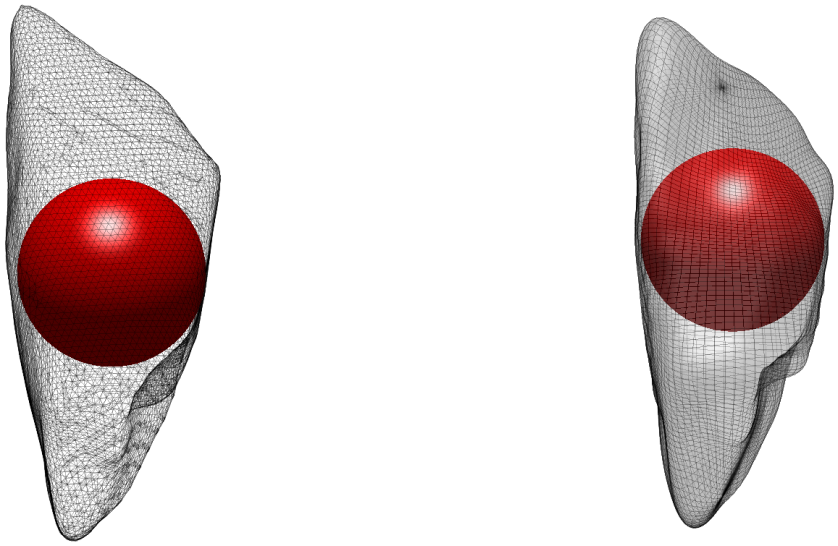
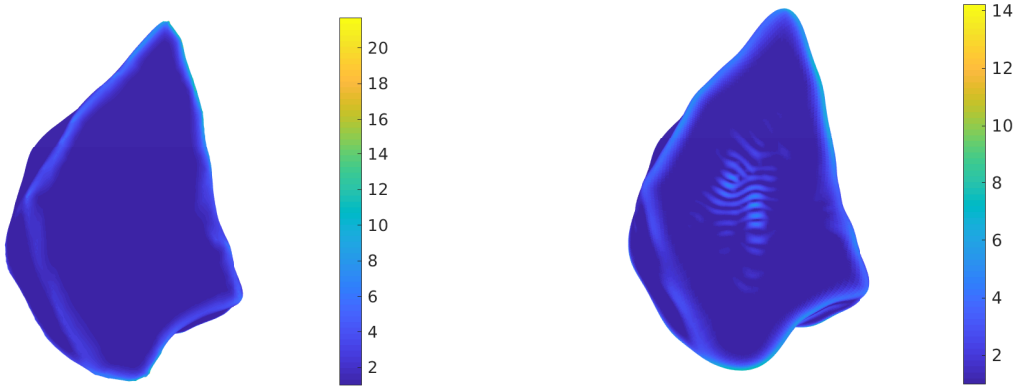


Figure 3.13 Convergence of the geometrical properties computed with the SH representation of the asphalt particle shown in Figure 3.12.



(a) STM inscribed radius of 6.52 (b) SH inscribed radius of 6.46

Figure 3.14 Largest inscribed sphere for STM and SH representations.



(a) $R_{\mathrm{STM}} = 0.49$ (b) $R_{\mathrm{SH}} = 0.52$

Figure 3.15 Colormap of $\max\{1, {}^{|\kappa_2|}/_{\kappa_{\mathrm{is}}}\}$ for the STM representation (a) and the SH representaiton (b).

CHAPTER 4 PROBABILISTIC MODELING

Left for clarification are the mandatory steps to generate virtual particles when given the SH representations of real particles, see Figure 4.1. These remaining steps are the focus of this chapter. We begin by detailing the required preprocessing to simplify the statistical modeling of the data. Preprocessing includes normalization, dimensionality reduction, and clustering. Normalization is important because it makes the SH coefficients independent on particle size. The dimensionality reduction is primordial to reduce the *curse of dimensionality* and is done using the Principal Component Analysis (PCA) algorithm. Finally, clustering algorithms allow one to group particles in different subpopulations with specific geometrical traits. Partitioning the data this way holds the promise of requiring less complex statistical models to learn the geometrical patterns within each subpopulation. Once the data has been processed in this manner, it can be used to calibrate a statistical model from which we can sample virtual particles. Before discussing the preprocessing and statistical models, we first proceed by defining some populations of manufactured particles used for verifying the algorithms.

4.1 Data Samples and Storage

Several populations of particles are considered, some are made of manufactured particles generated from analytical radial functions in spherical coordinates while other arise from STM representations of real particles. The manufactured particles are created from normalized shapes and then scaled by some random scaling factor A in order to obtain arbitrary size, independently of the geometry. Size is measured in terms of the quantity $L = \sqrt[3]{V}$ which is sampled by a uniform distribution, i.e. $L \sim U(1, 10)$, for all manufactured particles. The uniform distribution is chosen by simplicity but note that sampling L will results in a distribution of volumes that is concentrated towards 1.

Ellipsoids are closed surfaces that can be generated in spherical coordinates by the radial function

$$r(\theta, \phi) = \frac{A}{\sqrt{(\cos\theta \sin\phi)^2 + (r_{\mathrm{F}} \sin\theta \sin\phi)^2 + (r_{\mathrm{F}} r_{\mathrm{E}} \cos\phi)^2}}, \tag{4.1}$$

where r_{E} and r_{F} are the elongation and flatness indices. The scaling factor A is computed

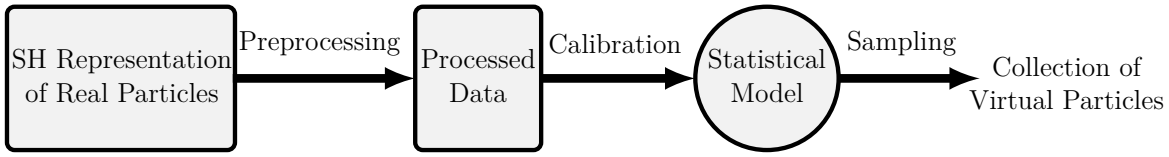


Figure 4.1 Steps to generate virtual particles when given the SH coefficients of a collection of real particles.

from the volume $V = L^3$, the elongation index r_{E}, and flatness index r_{F} via

$$A = \sqrt[3]{\frac{3\, r_{\mathrm{E}} r_{\mathrm{F}}^2}{4\pi V}} = \frac{1}{L}\sqrt[3]{\frac{3\, r_{\mathrm{E}} r_{\mathrm{F}}^2}{4\pi}}. \tag{4.2}$$

Three families of ellipsoids are generated, namely spheroids, prolates, and oblates using specific distributions of the parameters V, r_{E}, and r_{F} as shown in Table 4.1. The ranges of the uniform distributions are chosen heuristically to sample particles with distinct shapes.

Superquadrics form a rich set of shapes that are described by the following function in spherical coordinates

$$r(\theta, \phi) = \frac{A}{\left(\left(|\cos\theta \sin\phi|^{2/\epsilon_2} + |r_{\mathrm{F}} \sin\theta \sin\phi|^{2/\epsilon_2} \right)^{\epsilon_2/\epsilon_1} + |r_{\mathrm{F}} r_{\mathrm{E}} \cos\phi|^{2/\epsilon_1} \right)^{\epsilon_1/2}}, \tag{4.3}$$

where r_{F} and r_{E} are related to elongation and flatness indices and the degrees of freedom ϵ_1 and ϵ_2 affect the shapes in a continuous way. Note that superquadrics include ellipsoids by taking $\epsilon_1 = \epsilon_2 = 1$. Knowing $V, r_{\mathrm{F}}, r_{\mathrm{E}}, \epsilon_1$, and ϵ_2, one can compute the scaling parameter A using [54]

$$V = 2\frac{A^3}{r_{\mathrm{F}}^2\, r_{\mathrm{E}}} \epsilon_1 \epsilon_2\, \beta\Big(\frac{\epsilon_1}{2} + \epsilon_2, \epsilon_1\Big) \beta\Big(\frac{\epsilon_2}{2}, \frac{\epsilon_2}{2}\Big). \tag{4.4}$$

Four different families of superquadrics are created: cubes, cylinders, diamonds and boxes and the sampling distributions associated with each of these families are shown in Table 4.2. Like for the ellipsoids, the ranges of the uniform distributions is chosen heuristically to make the four groups of particles very distinct from one another.

Figures 4.2 and 4.3 show 36 samples from each of these manufactured populations. The spheroids, prolates, oblates, cubes and cylinders can be viewed as simple populations since only three degrees of freedom need to be sampled. Diamonds and boxes form populations with richer geometries in the sense that five degrees of freedom are sampled from uniform distributions. It is important to note that considering every degree of freedom impacts the

Table 4.1 – Distribution of the degrees of freedom for three populations of ellipsoids.

Degree of freedom	Spheroids	Prolates	Oblates
L	$U(1, 10)$	$U(1, 10)$	$U(1, 10)$
r_{E}	$U(0.8, 1)$	$U(0.4, 0.6)$	$U(0.75, 1)$
r_{F}	$U(0.8, 1)$	$U(0.75, 1)$	$U(0.4, 0.6)$

Table 4.2 – Distribution and values of the degrees of freedom on four populations of superquadrics.

Degree of freedom	Cubes	Cylinder	Diamonds	Boxes
L	$U(1, 10)$	$U(1, 10)$	$U(1, 10)$	$U(1, 10)$
r_{E}	$U(0.8, 1)$	$U(0.8, 1)$	$U(0.6, 1)$	$U(0.6, 1)$
r_{F}	1	1	$U(0.6, 1)$	$U(0.6, 1)$
ϵ_1	$U(0.2, 0.75)$	$U(0.1, 0.5)$	$U(1, 2)$	$U(0.2, 1)$
ϵ_2	ϵ_1	1	$U(1, 2)$	$U(0.2, 1)$

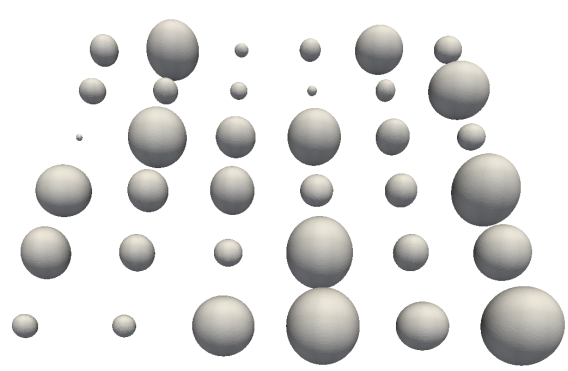
shape in a continuous way, and since the SH coefficients are a continuous function of $r(\theta, \phi)$, each population will lie on a d-manifold in the SH coefficient space where d is the number of sampled degrees of freedom.

The last population consists of particles collected from a river bed. This population exhibits more complex geometries and, unlike the manufactured ones, the true nature of their statistical distribution is unknown.

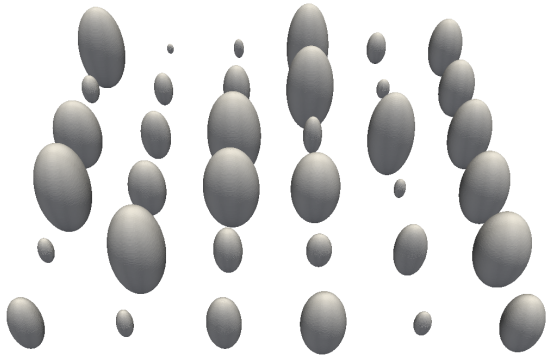
As explained in Chapter 3, the STM representation of particles is useful for 3D visualization, but not for statistical analysis. The SH coefficients of particles, however, lend themselves to statistical analysis since they allow one to represent particles as points in the vector space $\mathbb{R}^{(\ell_{\max}+1)^2}$. For notational simplicity, we shall denote $d := (\ell_{\max} + 1)^2$. For the rest of this chapter, the cutoff frequency $\ell_{\max}$ is set to 20 meaning that we are working in a 441 dimensional space.

Figure 4.4 compares the STM and unfiltered SH representations of diamond and box particles. We observe that, while both representations of boxes are nearly undistinguishable, the SH representation of diamonds is subject to major defects. Indeed, large surface ripples can be seen as well as an excess of edge smoothing. These issues can be attributed to the non-convexity and sharp edges of the diamonds, making the SH representation converge slowly.

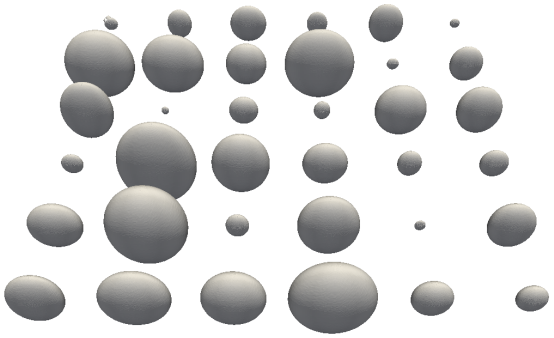
Figure 4.5 displays 36 samples of the SH representation of river particles. Most apparent is the presence of small surface ripples that could be easily reduced by applying the lanczos sigma factor. Note that despite having a very rich geometry, these particles are relatively



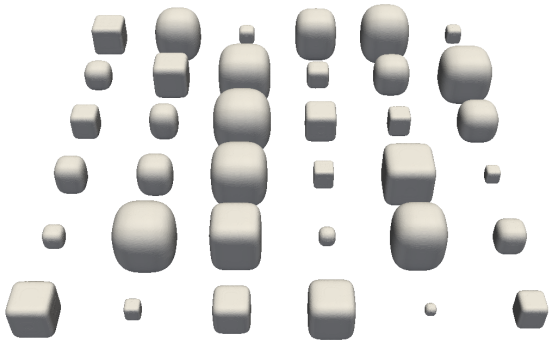
(a) Spheroids



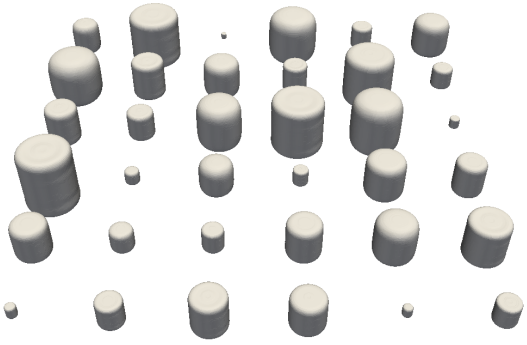
(b) Prolates



(c) Oblates

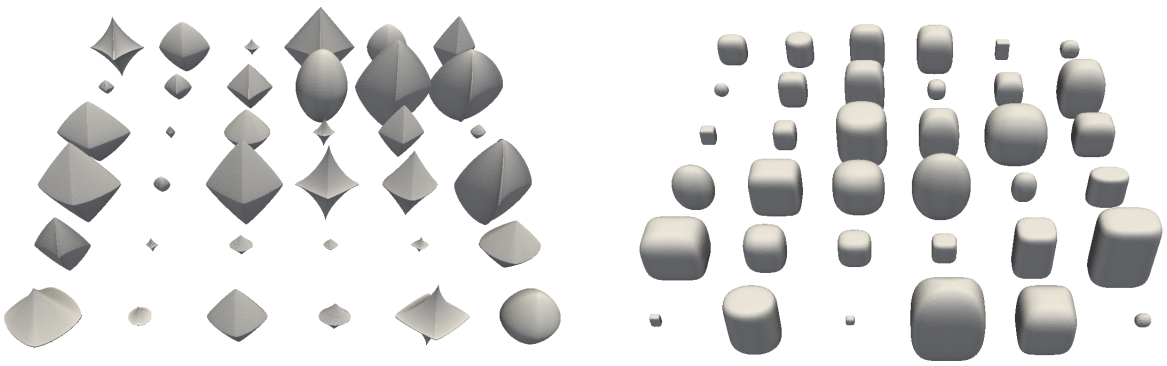


(d) Cubes



(e) Cylinders

Figure 4.2 STM representations of manufactured populations.



(a) Diamonds (b) Boxes

Figure 4.3 Additional STM representations of manufactured populations.

smooth implying that their SH coefficients should converge quickly.

Before analyzing the data, we need to simplify the notation of the SH coefficients. In order to do so, note that the coefficients are ordered as $c_0^0, c_1^{-1}, c_1^0, c_1^1, c_2^{-2}, c_2^{-1}, c_2^0, c_2^1, c_2^2, \ldots$ which suggests using the new index i such that

$$i = \ell^2 + \ell + m, \quad \text{for } i = 0, 1, 2, 3, \ldots \ d - 1. \tag{4.5}$$

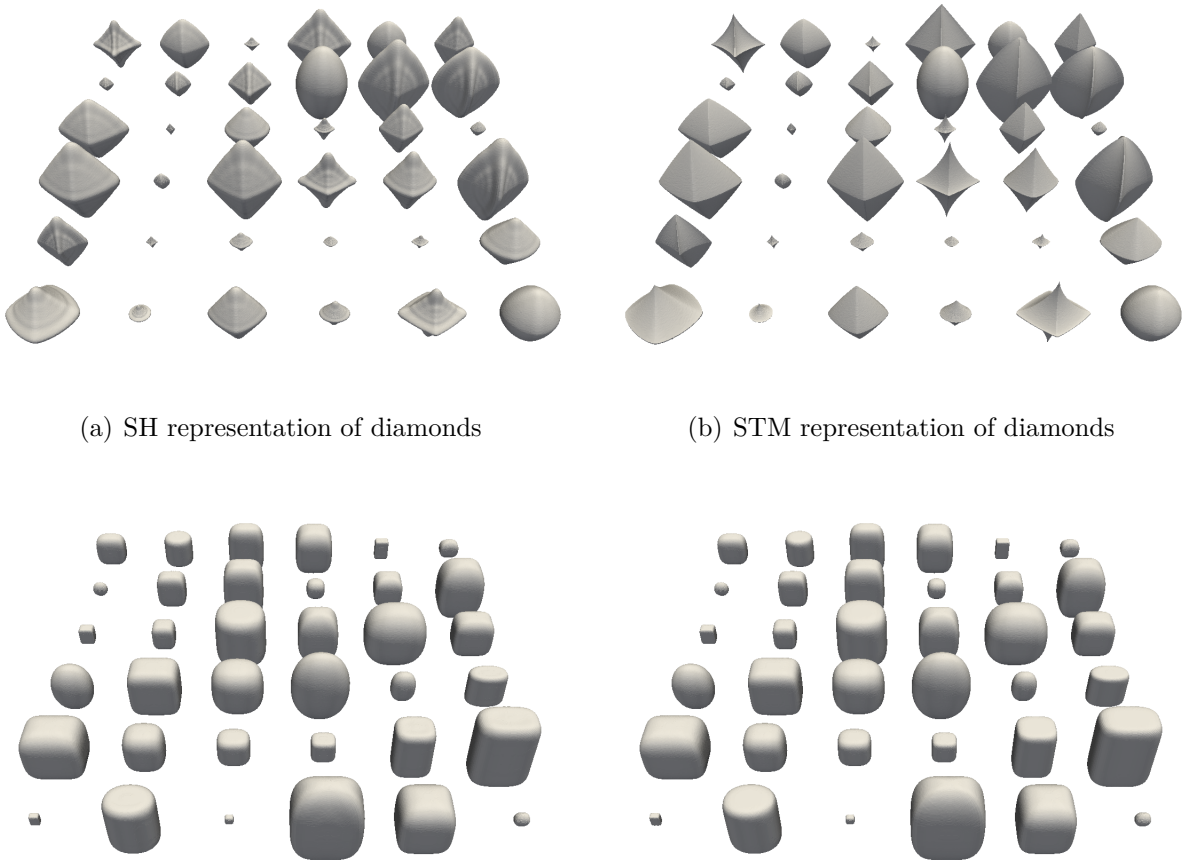
Given i, the original indices ℓ and m are recovered with

$$\begin{aligned} \ell &= \lfloor \sqrt{i} \rfloor, \\ m &= i - \ell^2 - \ell, \end{aligned} \tag{4.6}$$

where the correspondence between the coefficients is provided as follows:

$$\begin{array}{cccccccccc} c_0^0 & c_1^{-1} & c_1^0 & c_1^1 & c_2^{-2} & c_2^{-1} & c_2^0 & c_2^1 & c_2^2 & \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \cdots \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & \end{array}$$

The first step in the preprocessing stage is to distinguish, for each particle described by its SH representation, the information on its size from the information associated with other geometrical properties. Two size characteristic sizes in the literature are the mean radius



(a) SH representation of diamonds

(b) STM representation of diamonds

(c) SH representation of boxes

(d) STM representation of boxes

Figure 4.4 SH representation of manufactured populations.

$\langle r \rangle$ [14] and the characteristic length $L = \sqrt[3]{V}$ [17, 19, 24, 25]

$$\langle r \rangle = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} r(\theta, \phi) \sin \phi d\phi d\theta, \qquad L = \left(\frac{1}{3} \int_0^{2\pi} \int_0^{\pi} r(\theta, \phi)^3 \sin \phi \, d\phi d\theta \right)^{\frac{1}{3}}. \tag{4.7}$$

Despite being less popular, the mean radius is a more natural measure of the particle size than the characteristic length L when treating SH coefficients of particles. This is because the mean radius only depends on one of the SH coefficients, i.e. $c_0 = \sqrt{4\pi}\langle r \rangle$, see Equation (2.27). To the contrary, L depends on all SH coefficients through a nonlinear relation. This suggests that the coefficient c_0 is the best candidate for size characterization. Left for characterization is the *intrinsic* geometry, which refers to all geometric properties that are independent of size, i.e. aspect ratio, sphericity, roundness, convexity, etc. Encoding the intrinsic geometry using the SH coefficients is subtle since a scale-invariant quantity is required. It was previously

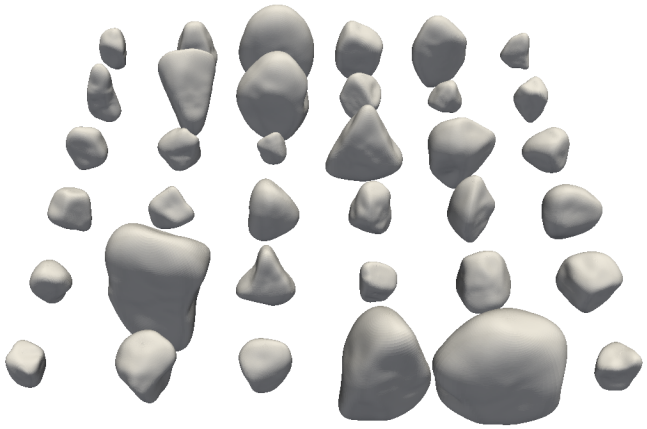


Figure 4.5 SH representations of river particles.

discussed that the modes $c_i Y_i$, $i \neq 0$, represent radial perturbations of the sphere of radius $\langle r \rangle$. Those perturbations hold the information about the intrinsic geometry. The fact that $|c_i| = \sqrt{4\pi}\sqrt{\left\langle (\Delta_i)^2 \right\rangle}$ implies that the coefficients are not scale-invariant. However, we observe that the ratio

$$\frac{|c_i|}{c_0} = \frac{\sqrt{4\pi}\sqrt{\left\langle (\Delta_i)^2 \right\rangle}}{\sqrt{4\pi}\langle r \rangle} = \frac{\sqrt{\left\langle (\Delta_i)^2 \right\rangle}}{\langle r \rangle} \geq \frac{\left\langle |\Delta_i| \right\rangle}{\langle r \rangle}, \tag{4.8}$$

is both scale-invariant and geometrically meaningful. It indeed provides an upper bound on the average absolute perturbation of the ith mode relative to the particle size $\langle r \rangle$. For this reason, we shall refer to this ratio as the relative perturbation. We therefore propose to consider the normalized coefficients

$$\widehat{c}_i = \begin{cases} c_0 & i = 0 \qquad \text{(Size)}, \\ \dfrac{c_i}{c_0} & i \neq 0 \qquad \text{(Intrinsic geometry)}, \end{cases} \tag{4.9}$$

when performing the statistical analysis of the particles in each population. The same normalization of the coefficients was previously used in [14] while other sources usually normalized the volume to unity, which is identical to dividing the SH coefficients by L. As explained earlier, dividing the SH coefficients by c_0 is a more natural normalization as it allows one to encode all information related to particle size in the coefficient $\widehat{c}_0$, and all the intrinsic geometry information into the relative perturbations $\widehat{c}_i = \dfrac{c_i}{c_0}$, $i = 1, 2, \ldots, d-1$.

Let $N_s \in \mathbb{N}$ be the number of particles in a given population. Each particle in the population

will be indicated by the index $j = 1, 2, \ldots, N_s$ and quantities associated with a particle j will be denoted with a superscript $^{(j)}$. For example, $\widehat{c}_i^{(j)}$ represents the ith coefficient of the jth particle. All the relative perturbations can be stored in a matrix $\widehat{\mathbf{C}} \in \mathbb{R}^{(d-1)\times N_s}$ such that $\widehat{\mathbf{C}}(i,j) = \widehat{c}_i^{(j)}$. We note that the jth column of $\widehat{\mathbf{C}}$ thus provides the relative perturbations $\widehat{\boldsymbol{c}}^{(j)}$ of the particle j. The reason $\widehat{c}_0^{(j)}$ is not included will become clear soon.

4.2 Principal Component Analysis

Dimensionality reduction refers to all techniques that allow one to represent data using fewer features while retaining the meaningful properties of the original data. We shall rely on dimensionality reduction here by reason of working in a space of dimension $d = 441$. The first coefficient $\widehat{c}_0$ contains all information on particle size so it must be kept intact. In light of this reasoning, dimensionality reduction is only applied to the relative perturbations. The simplest and most used dimensionality reduction algorithm is the Principal Component Analysis (PCA), which consists of projecting the data onto the affine subspace that minimizes the reconstruction loss [28].

4.2.1 Affine Changes of Basis

We first study affine changes of basis in the space of relative perturbations. Let $\boldsymbol{\mu} \in \mathbb{R}^{d-1}$ be an arbitrary vector. One can apply the translation $\widehat{c}_i^{(j)} - \mu_i$, $i = 1, 2, \ldots, (d-1)$, which relocates the origin at $\boldsymbol{\mu}$, see Figure 4.6. Let $\mathbf{1} \in \mathbb{R}^{d-1}$ be a vector of ones, the translation can be written in matrix notation as

$$\widehat{\mathbf{C}} - \boldsymbol{\mu}\mathbf{1}^T. \tag{4.10}$$

Let $\boldsymbol{p}_i \in \mathbb{R}^{d-1}$, $i = 1, 2, \ldots, d-1$, be orthonormal vectors forming a basis of $\mathbb{R}^{d-1}$. They can be stored as the square matrix

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \boldsymbol{p}_3 & \cdots & \boldsymbol{p}_{d-1} \end{bmatrix}. \tag{4.11}$$

Let $\mathbf{X} \in \mathbb{R}^{(d-1)\times N_s}$ be a matrix, $\mathbf{X}(i,j) = x_i^{(j)}$, such that

$$\mathbf{X} = \mathbf{P}^T(\widehat{\mathbf{C}} - \boldsymbol{\mu}\mathbf{1}^T), \tag{4.12}$$

$$\widehat{\mathbf{C}} = \mathbf{P}\mathbf{X} + \boldsymbol{\mu}\mathbf{1}^T. \tag{4.13}$$

The above transformation corresponds to a change of basis as shown in Figure 4.6. We now

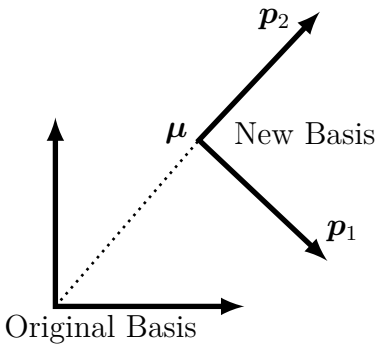


Figure 4.6 Orthonormal change of basis in $\mathbb{R}^2$.

provide a geometric interpretation of the coefficients $x_i^{(j)}$ by computing the radius of the jth particle:

$$\begin{aligned}
r^{(j)}(\theta,\phi) &= c_0^{(j)} Y_0(\theta,\phi) + \sum_{i=1}^{d-1} c_i^{(j)}\, Y_i(\theta,\phi) \\
&= c_0^{(j)} Y_0(\theta,\phi) + c_0^{(j)} \sum_{i=1}^{d-1} \widehat{c}_i^{(j)} Y_i(\theta,\phi) \\
&= c_0^{(j)} Y_0(\theta,\phi) + c_0^{(j)} \sum_{i=1}^{d-1} \left(\sum_{k=1}^{d-1} x_k^{(j)} \mathbf{P}(i,k) + \mu_i \right) Y_i(\theta,\phi) \\
&= c_0^{(j)} Y_0(\theta,\phi) + c_0^{(j)} \sum_{k=1}^{d-1} x_k^{(j)} \sum_{i=1}^{d-1} \mathbf{P}(i,k) Y_i(\theta,\phi) + c_0^{(j)} \sum_{i=1}^{d-1} \mu_i Y_i(\theta,\phi) \\
&= \underbrace{c_0^{(j)} \left(Y_0(\theta,\phi) + \sum_{i=1}^{d-1} \mu_i\, Y_i(\theta,\phi) \right)}_{\text{Initial shape}} + \underbrace{c_0^{(j)} \sum_{k=1}^{d-1} x_k^{(j)}\, X_k(\theta,\phi)}_{\text{Perturbations with new modes}} ,
\end{aligned} \tag{4.14}$$

where we have defined the functions $X_k(\theta,\phi) = \sum_{i=1}^{d-1} \mathbf{P}(i,k) Y_i(\theta,\phi)$. Since the matrix $\mathbf{P}$ that transforms the $Y_i(\theta,\phi)$ into $X_k(\theta,\phi)$ is orthonormal, the $X_k(\theta,\phi)$ functions form another orthonormal system of $L^2(S^2)$

$$\iint_{S^2} X_i(\theta,\phi) X_j(\theta,\phi) dS = \delta_{ij}. \tag{4.15}$$

Let us discuss the term *Initial shape*. It was previously discussed that $c_0^{(j)} Y_0$ represents the sphere that best approximates the jth particle. Adding $c_0^{(j)} \sum_{i=1}^{d-1} \mu_i\, Y_i$ perturbs the initial

sphere but this perturbation depends only on j through $c_0^{(j)}$, which is a scale parameter that does not affect the geometry. This means that the perturbation of the sphere is applied to **all** particles relative to their size. Using the clay modeling analogy, one begins to model all particles from chunks of clay with specific shapes instead of spheres.

We now interpret the term *Perturbation with new modes*. By considering the new basis in the space of relative perturbations, different types of radial perturbations are applied on the clay. These new perturbations are denoted $\Delta_k^{(j)} = c_0^{(j)} x_k^{(j)} X_k(\theta, \phi)$. The functions $X_k(\theta, \phi)$ being orthonormal, we obtain $4\pi\left\langle (\Delta_k^{(j)})^2 \right\rangle = \left(c_0^{(j)} x_k^{(j)}\right)^2$ so that

$$|x_k^{(j)}| = \frac{\sqrt{\left\langle (\Delta_k^{(j)})^2 \right\rangle}}{\langle r^{(j)} \rangle}, \tag{4.16}$$

which is the same as (4.8). To conclude, the new coefficients $x_k^{(j)}$ represent relative perturbations, but induced by the new functions $X_k(\theta, \phi)$ instead of $Y_i(\theta, \phi)$. Just like the SH coefficients, the clay modeling analogy is still valid when understanding the coefficients $x_k^{(j)}$.

4.2.2 Reconstruction Loss

PCA consists in finding the affine subset (hyperplane) of the relative perturbations space which best approximates the data. The vector $\boldsymbol{\mu}$ is now interpreted as a point that lives on a hyperplane. Let $m < d-1$ be the dimension of the hyperplane and define m orthonormal vectors $\boldsymbol{q}_i \in \mathbb{R}^{d-1}$ such that $\boldsymbol{\mu} + \boldsymbol{q}_i$ span the hyperplane. The vectors $\boldsymbol{q}_i$ are stored in a matrix $\mathbf{Q} \in \mathbb{R}^{(d-1)\times m}$.

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_m \end{bmatrix}. \tag{4.17}$$

We introduce $\mathbf{X} \in \mathbb{R}^{m \times N_s}$ such that

$$\mathbf{X} = \mathbf{Q}^T(\widehat{\mathbf{C}} - \boldsymbol{\mu}\mathbf{1}^T). \tag{4.18}$$

However, unlike in (4.13),

$$\begin{aligned} \widehat{\mathbf{W}} &= \mathbf{Q}\mathbf{X} + \boldsymbol{\mu}\mathbf{1}^T \\ &= \mathbf{Q}\mathbf{Q}^T(\widehat{\mathbf{C}} - \boldsymbol{\mu}\mathbf{1}^T) + \boldsymbol{\mu}\mathbf{1}^T \\ &= \mathbf{Q}\mathbf{Q}^T\widehat{\mathbf{C}} + (\mathbf{I} - \mathbf{Q}\mathbf{Q}^T)\boldsymbol{\mu}\mathbf{1}^T \\ &\neq \widehat{\mathbf{C}}, \end{aligned} \tag{4.19}$$

since $\mathbf{Q}\mathbf{Q}^T \neq \mathbf{I}$ ($\mathbf{Q}$ is not full-rank). The coefficients $\widehat{\mathbf{W}}(i,j) = \widehat{w}_i^{(j)}$ of the matrix $\widehat{\mathbf{W}} \in$

$\mathbb{R}^{(d-1)\times N_s}$ are called the reconstructions of $\widehat{c}_i^{(j)}$. The fact that relative perturbations are not equal to their reconstruction means that there was a loss of information, which is encapsulated in the reconstruction loss

$$R(\mathbf{Q}, \boldsymbol{\mu}) = \frac{1}{N_s}||\widehat{\mathbf{C}} - \widehat{\mathbf{W}}||_F^2, \tag{4.20}$$

where $||\mathbf{A}||_F^2 = \sum_{i=1}\sum_{j=1}(\mathbf{A}(i,j))^2$ is the Frobenius norm. Basically, the reconstruction loss is the averaged squared distances between the coefficients and their reconstruction, over all particles in the dataset. PCA consists in minimizing this function with respect to $\mathbf{Q}$ and $\boldsymbol{\mu}$ under the constraint that the columns of $\mathbf{Q}$ be orthonormal. By first minimizing with respect to $\boldsymbol{\mu}$, one obtains

$$\boldsymbol{\mu}^* = \frac{1}{N_s}\sum_{j=1}^{N_s}\widehat{\boldsymbol{c}}^{(j)}. \tag{4.21}$$

Minimizing with respect to $\mathbf{Q}$ is a lot more complex and a good outline of the proof is presented in the second chapter of [28]. The solution is to compute the sample covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{N_s}(\widehat{\mathbf{C}} - \boldsymbol{\mu}^*\mathbf{1}^T)(\widehat{\mathbf{C}} - \boldsymbol{\mu}^*\mathbf{1}^T)^T. \tag{4.22}$$

This $(d-1)\times(d-1)$ matrix is symmetric and positive definite. Therefore its eigenvectors can be chosen orthonormal. The positive eigenvalues are denoted by σ_i^2 and indexed in descending order: $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \cdots \geq \sigma_{d-1}^2$. For m fixed, the optimal matrix $\mathbf{Q}$ is the matrix whose columns are the first m eigenvectors of $\boldsymbol{\Sigma}$ with the highest eigenvalues. These vectors are conventionally referred to as the principal axes. This terminology extends to the projected coefficients $x_i^{(j)}$, which are now referred to as the principal components. A visual example of PCA is shown in Figure 4.7. Applying PCA with $m = 1$ on this example would project the data on the line spanned by $\boldsymbol{q}_1$, the principal axis with the largest variance. For more numerical details on PCA, we suggest the following tutorial [55].

4.2.3 Geometrical Intuition

This section shall conclude with the geometrical interpretation of PCA. The reconstructed particle radius is defined as

$$r_w^{(j)}(\theta,\phi) := c_0^{(j)}Y_0(\theta,\phi) + c_0^{(j)}\sum_{i=1}^{d-1}\widehat{w}_i^{(j)}Y_i(\theta,\phi). \tag{4.23}$$

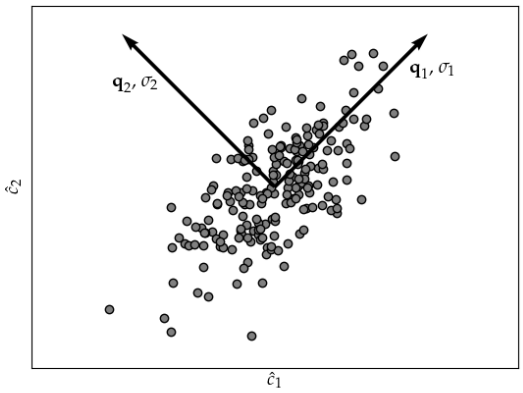


Figure 4.7 Two principal axes and associated standard deviations for a bivariate Gaussian distribution.

Using this definition, the reconstruction loss (4.20) can be expressed in geometrical space

$$R(\mathbf{Q}, \boldsymbol{\mu}) = \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{\left\langle \left(r^{(j)} - r_w^{(j)} \right)^2 \right\rangle}{\langle r^{(j)} \rangle^2}. \tag{4.24}$$

The proof of (4.24) is provided in Appendix A. Note that the dependencies on $\boldsymbol{\mu}$ and $\mathbf{Q}$ are hidden in the reconstructed particle coefficients $\widehat{w}_i^{(j)}$. This equation demonstrates why it is justifiable to apply PCA on relative perturbations instead of all the SH coefficients. Doing so enables one to obtain an objective function that minimizes the relative error of the radius functions. Minimizing the reconstruction loss can be explained in terms of the clay-modeling analogy. The following holds true

$$r^{(j)}(\theta,\phi) \approx r_w^{(j)}(\theta,\phi) = \underbrace{c_0^{(j)} \quad Y_0(\theta,\phi) + \sum_{i=1} \mu_i \, Y_i(\theta,\phi) \Big)}_{\text{Initial shape}} + \underbrace{c_0^{(j)} \sum_{k=1}^{m} x_k^{(j)} \, X_k(\theta,\phi)}_{\text{Perturbations with new modes}} , \tag{4.25}$$

the only difference with (4.14) being that the m functions $X_k(\theta,\phi)$ are now derived from the principal components $\mathbf{Q}$ instead of the arbitrary vectors $\mathbf{P}$. Minimizing (4.24) with respect to $\boldsymbol{\mu}$ and $\mathbf{Q}$, is equivalent to searching for the initial shapes and new m perturbations $x_k^{(j)} X_k(\theta,\phi)$, which best approximate the radial functions $r^{(j)}(\theta,\phi)$ over all particles in the data. The division by $\langle r^{(j)} \rangle^2$ in (4.24) is a way to balance the contributions of the small and large particles.

The final aspect of PCA that must be discussed is the selection of m. The most established practice is to compute the cumulative variance $\mathrm{CV}(m)$ of the m first principal components

$$\mathrm{CV}(m) = \frac{\sum_{i=1}^{m} \sigma_i^2}{\sum_{i=1}^{d-1} \sigma_i^2}. \tag{4.26}$$

The heuristic choice of m is to set a threshold between 95% and 99% and fix m as the first value where $\mathrm{CV}(m)$ becomes larger than the selected threshold. This heuristic can be justified by demonstrating there is an affine relationship between the reconstruction loss and the cumulative variance, see Appendix A.

4.3 Clustering

The final step of the data preprocessing is the identification subpopulations of particles. A subpopulation can be described as a set of particles which look similar independently of their sizes. Good examples include spheroids, prolates, oblates, cubes, or cylinders. To make this statement more precise, one needs to mathematically define the notion of similarity between particles independently of their size. Independence with respect to size requires the use of relative perturbations while ressemblance can be measured in terms of distance within the vector space of the SH coefficients. In other words, computing the distance between the relative perturbations of the jth and kth particle should be an indicator of resemblance that ignores size. Since $\mathbb{R}^{d-1}$ is a normed vector space, the distance $d(j,k)$ between the particles j and k can be inferred from a norm $\left\|\widehat{\boldsymbol{c}}^{(j)} - \widehat{\boldsymbol{c}}^{(k)}\right\|$ and the definition of a subpopulation can thus be stated as

> *A subpopulation is a set of particles which are close to each other with respect to some norm $\|\,.\,\|$ in the space of the relative perturbations $\widehat{c}_i$, $i \neq 0$.*

Several choices for a norm are possible. We shall restrict ourselves to norms of the form

$$\left\|\widehat{\boldsymbol{c}}^{(j)} - \widehat{\boldsymbol{c}}^{(k)}\right\|_{\mathbf{A}}^2 = \left(\widehat{\boldsymbol{c}}^{(j)} - \widehat{\boldsymbol{c}}^{(k)}\right)^T \mathbf{A}^{-1} \left(\widehat{\boldsymbol{c}}^{(j)} - \widehat{\boldsymbol{c}}^{(k)}\right), \tag{4.27}$$

where $\mathbf{A}$ is a symmetric positive definite matrix. Taking $\mathbf{A} = \mathbf{I}$, one gets the classical Euclidean norm

$$\left\|\widehat{\boldsymbol{c}}^{(j)} - \widehat{\boldsymbol{c}}^{(k)}\right\|_2^2 := \sum_{i=1}^{d-1} \left(\widehat{c}_i^{(j)} - \widehat{c}_i^{(k)}\right)^2. \tag{4.28}$$

Note that the index starts at $i = 1$ since the vectors $\widehat{\boldsymbol{c}}$ only include the relative perturbations. The Euclidean norm has simple expression in terms of the radial function of the particles j

and k, i.e. $r^{(j)}(\theta,\phi)$ and $r^{(k)}(\theta,\phi)$

$$\left\|\widehat{\boldsymbol{c}}^{(j)}-\widehat{\boldsymbol{c}}^{(k)}\right\|_2^2=\left\langle\left(\frac{r^{(j)}}{\langle r^{(j)}\rangle}-\frac{r^{(k)}}{\langle r^{(k)}\rangle}\right)^2\right\rangle. \tag{4.29}$$

A second norm of interest is the Mahalanobis norm which is obtained by setting $\mathbf{A}=\boldsymbol{\Sigma}$, the covariance matrix of the data,

$$\left\|\widehat{\boldsymbol{c}}^{(j)}-\widehat{\boldsymbol{c}}^{(k)}\right\|_{\boldsymbol{\Sigma}}^2=\left(\widehat{\boldsymbol{c}}^{(j)}-\widehat{\boldsymbol{c}}^{(k)}\right)^T\boldsymbol{\Sigma}^{-1}\left(\widehat{\boldsymbol{c}}^{(j)}-\widehat{\boldsymbol{c}}^{(k)}\right), \tag{4.30}$$

Open unit balls in that case are ellipsoids that are elongated in the directions of large variance in the data. Note that in the literature, one usually introduces the Mahalanobis distance, but considering the Mahalanobis norm is more general. This norm is of interest because it naturally appears when working with multivariate Gaussians.

To get preleminairy evidence of the relation between geometrical ressemblance and distance in the vector space relative perturbation $\mathbb{R}^{d-1}$, the following experiment is conducted: three arbitrary river particles are chosen and their three closest neighbors with respect to the Euclidean and Mahalanobis norms are identified, see Figure 4.8. It appears that Euclidean proximity in the space of relative perturbations is indeed linked to our intuition of ressemblance, which is a consequence of (4.29). However, the Mahalanobis norm seems to have failed at identifying neighbor particles with meaningful geometric similarities. Relations between norms, proximity and data dimension are studied further in Appendix D. The theoretical and empirical results discussed in Appendix D, as well as Figure 4.8, suggest that Euclidean nearest neighbors are well-defined despite high dimensionality, while Mahalanobis nearest neighbors become ill-defined in high dimensions.

The problem of identifying subpopulations becomes that of finding aggregates of points which are close to each other in the relative perturbations space. This branch of statistical analysis is known as clustering so each subpopulation shall now be referred to as a cluster. Clustering algorithms are susceptible to the curse of dimensionality which is why it is better, in practice, to apply them on the principal components rather than the relative perturbations $\widehat{c}_i^{(j)}$. We shall then search for clusters in the space $\mathbb{R}^m$ of the principal components. Note the norms (4.28) and (4.30) can also be applied to the coefficients $x_i^{(j)}$ while still retaining the same interpretation. Let N_c be the number of clusters and $\mathcal{T}=\{\boldsymbol{x}^{(j)}\}_{1\leq j\leq N_s}$ be the unordered dataset of principal components. Clusters, which are denoted K_i, $i=1,2,\ldots,N_c$, form a

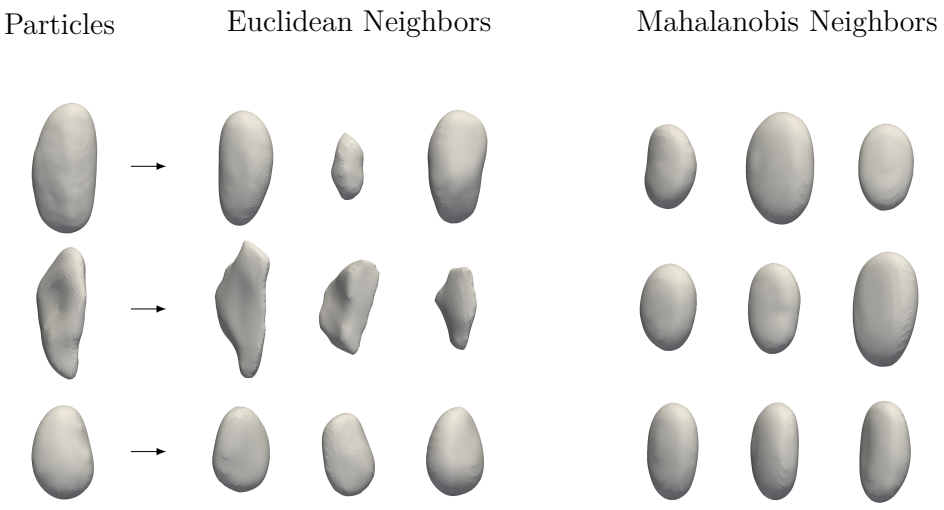


Figure 4.8 3 Nearest neighbors of three river particles with respect to the Euclidean and Mahalanobis norms.

partition of the dataset

$$K_i \subset \mathcal{T} \quad \text{such that} \quad \bigcup_{i=1}^{N_c} K_i = \mathcal{T} \quad \text{and} \quad K_k \bigcap K_i = \varnothing. \tag{4.31}$$

Partitioning data this way has two main applications. Firstly, since each cluster is constituted of particles with similar geometry, one can hope that fitting a distinct statistical model associated with each cluster would result in simpler models. These models could be easier to train and to interpret than a single complex model fitting the whole dataset. For example, if one attempts to fit a single model on a dataset consisting of triangles and circles, the model would need to learn about the concepts of equiaxiality, smoothness, flat faces, and sharp edges. On the other hand, fitting distinct models on circles and triangles would allow each model to learn only a subset of those concepts. The second application of partitioning the data is to gain geological insight about the data distribution. Indeed, different clusters could correspond to particles with different geological backgrounds.

Two clustering algorithm, namely the K-Means and Gaussian Mixture (GM) will be discussed. The K-Means algorithm is the simplest and most commonly used. It assumes that the clusters are approximately spherical and have similar sizes. It is based on computing the Euclidean distance between the data points and points $\boldsymbol{\mu}_i$, $i = 1, 2, \ldots, N_c$, called the centroids of the

clusters. The decision rule to assign SH representations of particles to clusters is

$$K_i = \left\{ \boldsymbol{x}^{(j)} \in \mathcal{T} \mid i = \underset{1 \leq k \leq N_c}{\operatorname{argmin}} \left\| \boldsymbol{x}^{(j)} - \boldsymbol{\mu}_k \right\|_2^2 \right\}. \tag{4.32}$$

Basically, each cluster consists of the data points that are closer to its centroid than to any other centroid. The full procedure to obtain K-Means clusters can be found in [29]. This algorithm is appealing because of its simplicity and its use of the Euclidean norm, which has a nice geometrical interpretation as seen in (4.29).

Unlike K-Means, Gaussian Mixture (GM) allows one to describe non-spherical clusters. The main assumption is that each cluster can be approximated by a multivariate Gaussian. The process of finding clusters is replaced by a fit of N_c Gaussians on the data using the Expectation Maximization algorithm [46]. After fitting the Gaussians, the algorithm computes the probabilities $\mathbb{P}[\boldsymbol{x}^{(j)} \in K_i]$ and the decision rule for cluster assignment is

$$K_i = \left\{ \boldsymbol{x}^{(j)} \in \mathcal{T} \mid i = \underset{1 \leq k \leq N_c}{\operatorname{argmax}} \mathbb{P}[\boldsymbol{x}^{(j)} \in K_k] \right\} \tag{4.33}$$

For mathematical details on GM and Expectation Maximization, we suggest reading [29, 46].

4.3.1 Silhouette

One of the challenges of K-Means and GM clustering is to determine the number of clusters. This is actually a fuzzy notion since there is no precise mathematical definition of a cluster. Intuitively, a cluster should contain points that are more similar to each other than to points from other clusters. The silhouette is a quality measure of clusters that is based on this intuition [56]. The silhouette can be computed for every point $\boldsymbol{x}^{(j)}$ in a sample. If we let $\boldsymbol{x}^{(j)} \in K_i$, we first define the following quantities

$$a(j) = \frac{1}{\operatorname{card}(K_i) - 1} \sum_{\boldsymbol{x}^{(k)} \in K_i \,|\, k \neq j} d(j,k), \tag{4.34}$$

$$b(j) = \min_{k \neq i} \left(\frac{1}{\operatorname{card}(K_k)} \sum_{\boldsymbol{x}^{(m)} \in K_k} d(j,m) \right), \tag{4.35}$$

where $d(j,k)$ is the distance between the relative perturbations of the jth and kth particles. The term $a(j)$ represents the average similarity between the particle j and the other particles from the same cluster while $b(j)$ represents the average similarity between the jth particles

and particles from the most similar cluster. The silhouette is defined as

$$s(j) = \frac{a(j) - b(j)}{\max\{a(j), b(j)\}}, \tag{4.36}$$

and its value lies between -1 and 1. A value close to 1 indicates that the jth particle is assigned to the right cluster, while a value of -1 suggests that it is put in the wrong cluster. Finally, a value close to 0 signifies ambiguity as to which cluster the particle should belong to. To select the number of clusters, one should try different values of N_c and select the one that yields the highest average silhouette over all particles.

The Euclidean and Mahalanobis norms were previously proposed as measures of distance between particles. When computing the silhouette of clusters obtained with K-Means, one should use $d(j,k) = \left\|\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}\right\|_2$ since it is the *build-in* metric of the algorithm. Choosing the right norm to compute the silhouette of GM clusters is more subtle because clusters can be very elongated in certain directions. The Euclidean norm would automatically assign low silhouettes to points at the extreme ends of the clusters, even when such elongations are intrinsic to the structure of the clusters. Using the Mahalanobis norm with $\mathbf{A} = \boldsymbol{\Sigma}$ does not solve this issue since the total covariance cannot describe the shape of a specific cluster. Our solution is to set $\mathbf{A}$ to $\mathbf{A}_i^* := \det(\boldsymbol{\Sigma}_i)^{-\frac{1}{d-1}}\boldsymbol{\Sigma}_i$ where $\boldsymbol{\Sigma}_i$ is the covariance of the ith cluster, which is the cluster the jth particle is assigned to. One justification for this choice is that it captures the shape of the local cluster while removing any distortion of volumes, see Figure 4.9. We therefore use $d(j,k) = \|\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}\|^2_{\mathbf{A}_i^*}$ when computing the silhouette of GM clusters[1]. Clusters are likely to have different covariances, so there will be a loss of symmetry $d(j,k) \neq d(k,j)$.

4.3.2 Cluster Compactness

Another common approach to determine N_c is to define a notion of cluster compactness. This measure should diminish as N_c increases since clusters become more localized, until they become singletons. As N_c grows, one should reach a regime where the gain in compactness becomes lesser. The largest values of N_c before that regime is attained should be selected. For K-Means, compactness is measured in terms of the average inertia of the clusters

$$\mathrm{I}(N_c) = \frac{1}{N_c}\sum_{i=1}^{N_c}\left(\frac{1}{\mathrm{card}(K_i)}\sum_{\boldsymbol{x}^{(j)}\in K_i}\left\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}_i\right\|_2^2\right). \tag{4.37}$$

[1]Computing the silhouette with this norm is not common in the literature compared to the Euclidean norm. Therefore, one should approach this suggestion with skepticism.

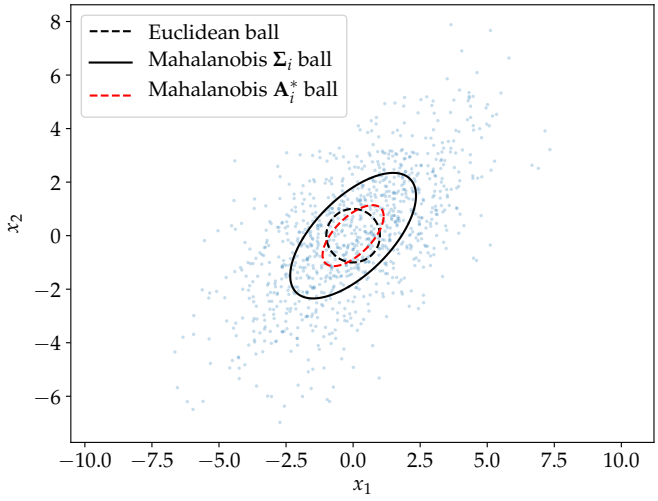


Figure 4.9 Comparisons of the unit ball with various metrics on the ith cluster K_i.

To heuristically select N_c, one must plot the average inertia versus the number of clusters and search for a point where the slope changes abruptly. This point is often referred to as the *elbow* of the graph. The value of N_c corresponding to the *elbow* is selected as the number of clusters. When working with GM, one can get a similar measure with the Bayesian Information Criterion (BIC) which was first introduced by Schwarz in the context of model selection [57]. It is defined as

$$\mathrm{BIC}(N_c) = \log(N_s)p - 2\sum_{j=1}^{N_s} \log\left(f(\boldsymbol{x}^{(j)};\Theta)\right), \tag{4.38}$$

were p is the number of parameters in the model and $f(\boldsymbol{x}^{(j)};\Theta)$ is the probability density given by the GM algorithm with N_c Gaussians. The symbol Θ accentuates the fact that the density is parametric. Common practice is to choose N_c that minimizes the BIC. The log-likelihood term in (4.38) measures cluster compactness. Indeed, as the number of Gaussian clusters increases, a higher probability density is assigned to each individual data point. This results in a decrease of the negative log-likelihood as N_c increases. However, at some point, the decrease in negative log-likelihood becomes lesser compared to the growth of the left term $\log(N_s)p$, which expands with N_c. This will result in a minimum of the BIC, that plays the same role as the *elbow* for the heuristic average inertia.

4.4 Statistical Model

In order to build the probabilistic model from Figure 4.1, we are going to consider machine learning algorithms from a branch called *density estimation*. This rich branch of machine learning includes models of probability density that are calibrated using the data points. The calibration process is often referred to as the training step so we shall use both terminologies interchangeably. Once the training is done, it is possible to randomly sample from such models which allows one to generate virtual particles with, hopefully, similar geometries as particles from the training set. Since the sampling is random, one can sample an arbitrary large number of virtual particles making large scale DEM simulations possibles or any other application that requires large samples of realistic particles. We first suppose that the size $\widehat{c}_0^{(j)}$ and each principal components $x_i^{(j)}$ are one realization of the random variables $\widehat{c}_0$ and $\widehat{x}_i$, and that the set of coefficients follows the joint probability density distribution ρ

$$\left(\widehat{c}_0, x_1, x_2, \ldots, x_m\right) \sim \rho. \tag{4.39}$$

Note that the parenthesis superscript $^{(j)}$ now represents a specific realization of the random variables. The following steps are required to generate a virtual particle:

1. Approximate the unknown density ρ with a model ϕ;
2. Sample new principal components $\left(\widehat{c}_0^{(j)}, x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)}\right)$ from ϕ;
3. Reconstruct the relative perturbations $\widehat{w}_i^{(j)}$, $i = 1, 2, \ldots, d-1$ using (4.19),
4. Scale the reconstructed relative perturbations as

$$w_i^{(j)} = \begin{cases} \widehat{c}_0^{(j)} & i = 0, \\ \widehat{c}_0^{(j)} \widehat{w}_i^{(j)} & i \neq 0, \end{cases} \tag{4.40}$$

5. Compute the particle radius $r^{(j)}(\theta, \phi)$ from the reconstructed coefficients $w_i^{(j)}$.

4.4.1 Simplification Hypotheses

It is helpful to make hypotheses about the density ρ in order to simplify its modeling. The primary hypothesis (H_0) is to suppose that the size and intrinsic geometry components of

the density are independent,

$$\phi(\widehat{c}_0, x_1, x_2, \ldots, x_m) \overset{H_0}{=} \underbrace{\phi_0(\widehat{c}_0)}_{\text{Size}} \underbrace{\phi_{1:m}(x_1, x_2, \ldots, x_m)}_{\text{Intrinsic geometry}}. \tag{4.41}$$

This hypothesis was first proposed by Grigoriu et al. [14]. It implies that one can sample the size and geometry independently, which will generate samples of particles where small and large grains look alike. By construction, it is expected to hold for the populations of manufactured particles. However, the hypothesis may not be valid for populations of real particles considering small grains could be sharper and thinner than larger ones. One way to resolve this issue could be clustering in order to identify aggregates of particles with similar geometries independently of their size. Therefore, we could expect the independence hypothesis to hold better on each separate cluster rather than on the whole population. This reasoning is illustrated in Figure 4.10. To interpret the figure, note that H_0 holds if and only if small and large particles all look alike. The hypothesis clearly is not valid on the whole dataset (left) since large particles tend to be more ellipsoidal and small particles tend to be both ellipsoidal and rectangular. Training a single model to generate this dataset could be very complicated since it would not only require to generate rectangles and ellipses, but it would also need to learn that rectangles tend to be smaller in average than ellipses. Examining the right part of Figure 4.10, we observe that the ideal clustering algorithm separates the ellipse and rectangle subpopulations. Moreover, the H_0 hypothesis holds on each cluster separately since all particles look alike independently of their size. Training a separate model on each cluster could prove to be simpler since the models would generate particles with a specific geometry and with arbitrary sizes within some range.

Due to the finite number of samples, determining whether the independence hypothesis holds is not a trivial task. A common measure to infer H_0 is given by the Pearson correlation coefficient between the two random variables X and Y

$$\mathrm{Cor}[X, Y] = \frac{\mathbb{E}\Big[(X - \mathbb{E}[X])\,(Y - \mathbb{E}[Y])\Big]}{\mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big]\,\mathbb{E}\Big[(Y - \mathbb{E}[Y])^2\Big]\Big)^{1/2}}, \tag{4.42}$$

where $\mathbb{E}[X]$ denotes the expected value of X. The Pearson correlation coefficient is related to the notion of independence, but is not equivalent

$$X \text{ and } Y \text{ are independent } \implies \mathrm{Cor}[X, Y] = 0. \tag{4.43}$$

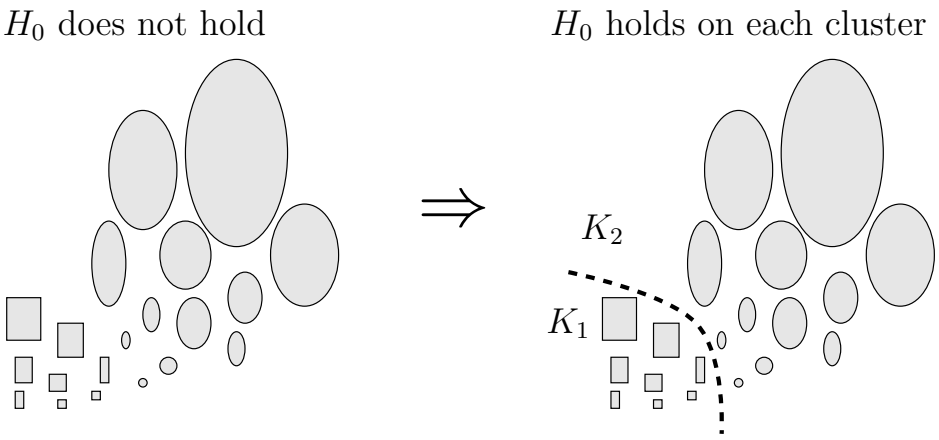


Figure 4.10 Idealized example of how clustering could help uncovering subpopulations where the independence hypothesis (H_0) holds.

The converse can be shown to be true only for very specific distributions like multivariate Gaussians. A more powerful measure of independence between two random variables X and Y is the Mutual Information (MI)

$$\mathrm{MI}[X,Y] = D_{\mathrm{KL}}\Big(f_{XY}(x,y)||f_X(x)f_Y(x)\Big) = \int_{\mathcal{X}}\int_{\mathcal{Y}} f_{XY}(x,y)\log\left(\frac{f_{XY}(x,y)}{f_X(x)f_Y(x)}\right)dydx, \tag{4.44}$$

where $f_{XY}(x,y)$ is the joint distribution for X and Y and $f_X(x)$ and $f_Y(x)$ are the marginal distributions of X and Y, respectively. $D_{\mathrm{KL}}(f||g)$ is the Kullback–Leibler divergence, which is a positive measure of the distance between distributions. The following holds

$$X \text{ and } Y \text{ are independent } \iff \mathrm{MI}[X,Y] = 0, \tag{4.45}$$

which is a stronger statement than (4.43). We compute the mutual information with the function `mutual_information_regression` from the Scikit-learn Python Library [58]. The Scikit-learn implementation is based on the K-Nearest-Neighbors algorithm discussed in [59]. Because the KL divergence is estimated using a finite number of samples, the mutual information may not vanish even if the variables are independent. To solve this issue, one must build a statistical test. The authors in [60] describe how to build a test of independence based on MI. The first step is to generate fake samples under H_0 by using resampling methods on each variable independently. By computing MI on each fake sample, a sampling distribution of MI is generated. By evaluating MI on the actual data, one can introduce the p-value of the test as the ratio of elements of the sampling distribution that have a larger

MI than the one computed on the actual data. The significance of a test is noted α, with $0 < \alpha < 1$. and represents an upper bound on the probability to reject the null-hypothesis when it is in fact valid. Therefore, to build a Mutual Information independence test of significance α, one must reject the null-hypothesis when the p-value is smaller than α. This way, one gains confidence that large values of MI are not attributed to errors caused by finite sample estimation.

To generate the sampling distribution under H_0, the Bootstrap scheme [46, page 107] with $B = 100$ different bootstrap samples is used. Since B is so low, the p-value returned by the test has a very low resolution and may not always be trustworthy. Using larger values of B is not ideal since the test becomes extremely expensive, especially when it is applied on multiple variables. An alternative solution would be to use the MI test in conjunction with a second test. We thus pair the test with the Spearman correlation test implemented as `spearmanr` from the SciPy Python Library [61]. The null-hypothesis of this test is that the data is uncorrelated. To verify this hypothesis, the test computes the Spearman correlation which is defined as the Pearson correlation applied to the rank of the variables instead of their values. This implies that small p-values are strong indicators of monotonic relationships and not just linear ones, which is the case with the basic Pearson correlation. Both MI and Spearman tests are verified on bivariate distributions and results are shown in Appendix C.

4.4.2 Multivariate Gaussian

In the specific scenario where $\phi_{1:m}(x_1, x_2, ..., x_m)$ can be shown to follow a multivariate Gaussian distribution, the problem becomes staggeringly simple. This is because the principal components of a multivariate Gaussian follow independent univariate Gaussians with mean zero and variance σ_i^2. Since the variances σ_i^2 are directly given by the PCA procedure, no training is required. Moreover, in order to sample from such this model, each principal component $\boldsymbol{x}_i$ can be sampled independently from univariate Gaussians. This method is very common in the geology literature [17,19,24], however, hypothesis tests of normality are rarely discussed.

In this study, two normality tests are applied to the SH coefficients using the statistical module of the SciPy Python library [61]. Since the principal components and the SH coefficients are related by an affine relation, the tests can be applied to either of them. This holds because any multivariate Gaussian can be shown to be an affine transformation of another Gaussian. The first test is the D'agostino test implemented as the function `normaltest` [61]. This univariate test assumes that the data is normally distributed and computes the p-value based on the sample skewness and kurtosis. The second test is the so-called Shapiro-Wilk test im-

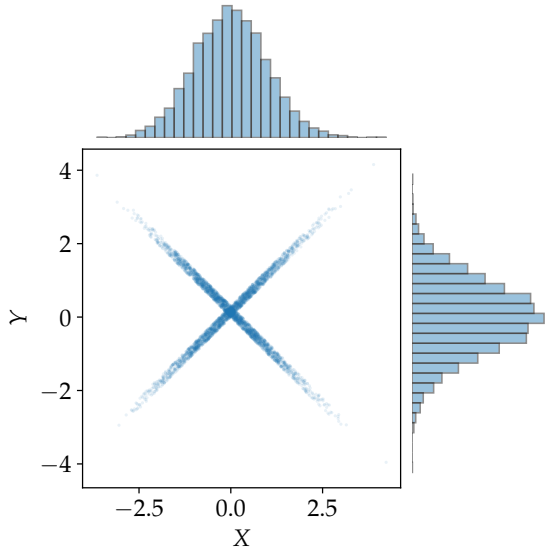


Figure 4.11 Classic example where X and Y follow Gaussian distributions but their joint distribution is not a multivariate Gaussian. Jitter is applied to the scatter plot to make points more visible.

plemented as the function `shapiro` [61]. This univariate test also assumes that the data is normally distributed and computes the p-value by using the ranked data points. Since each test is univariate, they must be applied to every single feature in the dataset.

Note that even in the ideal scenario where every single SH coefficient can be shown to follow a univariate Gaussian, this is still not enough evidence to prove the data follows a multivariate normal. This is because one can create distributions that have Gaussian marginals but are not multivariate Gaussians. The typical example is to consider $X \sim N(0,1)$ and $W \sim \text{Bernoulli}(0.5)$. One can show that $Y = (2W-1)X$ follows a Gaussian distribution. Therefore, we have constructed a random vector (X, Y) whose marginals are Gaussians but whose joint distribution is clearly not a multivariate Gaussian, as seen in Figure 4.11.

Other similar counter-examples exist but are usually very contrived so, in practice, the multivariate normal hypothesis can still hold reasonably well if one manages to prove that all SH coefficients follow normal distributions.

4.4.3 Kernel Density Estimation

When there is strong evidence that the data does not follow a multivariate Gaussian, other probabilistic models should be considered. One simple alternative is the Kernel Density

Estimation (KDE)

$$\phi_{1:m}(x_1, x_2, ..., x_m) = \frac{1}{N_s} \sum_{j=1}^{N_s} \prod_{i=1}^{m} \frac{1}{h\sigma_i} K\left(\frac{x_i - x_i^{(j)}}{h\sigma_i}\right), \tag{4.46}$$

where σ_i is the standard deviation of the ith principal component and the so-called kernel $K(.)$ is a univariate probability density distribution chosen symmetric around the origin. The parameter h is called the bandwidth and is used to control the spread of the kernel. The most popular kernel is the Gaussian

$$K(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \tag{4.47}$$

Figure 4.12 illustrates the motivation behind KDE. The model is calibrated by setting the bandwidth to its optimal value, which is critical as small values induce a high variance of the density estimation and large values oversmooth the distribution. There are two common approaches to select the value of the parameter. The first technique is to use a point estimate using the data. For example, Scott [27] showed that for m-dimensional multivariate Gaussian data with no correlations the optimal bandwidth with a Gaussian kernel is

$$h_{\text{Scott}} = N_s^{-1/(m+4)}. \tag{4.48}$$

Notice that the optimal bandwidth decreases as the amount of data N_s increases. Moreover, when N_s is fixed, the optimal bandwidth increases with dimension m. This is a manifestation of the curse of dimensionality, where the growing sparsity of data points forces the kernel to increase its spread in order to *fill out* the empty regions of space. The second approach consists in performing K-fold cross-validation using either the Mean Integrated Squared Error or the log-likelihood as performance measures [27]. We choose to work with the log-likelihood since it is standardly used in the Scikit-learn Python library. Like Scott's estimator, cross-validation is subject to the curse of dimensionality. More precisely, maximizing the cross-validated log-likelihood forces the kernels to increase their spread in order to assign a considerable probability density to each of their neighboring data points that are not part of the same fold, see Figure 4.13. In high dimensions, data points tend be located very far away from their nearest neighbors, which forces the bandwidth to become ridiculously large. Our method for selecting the bandwidth is a mix of Scott's estimator and cross-validation. More precisely, K-fold cross-validation is applied with Scott's estimator used as a starting value for the linesearch of h.

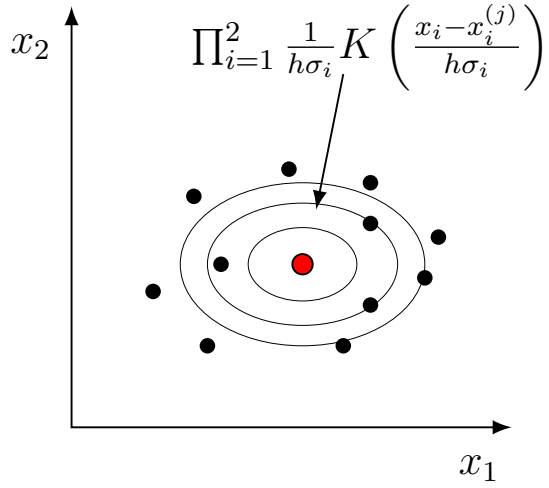
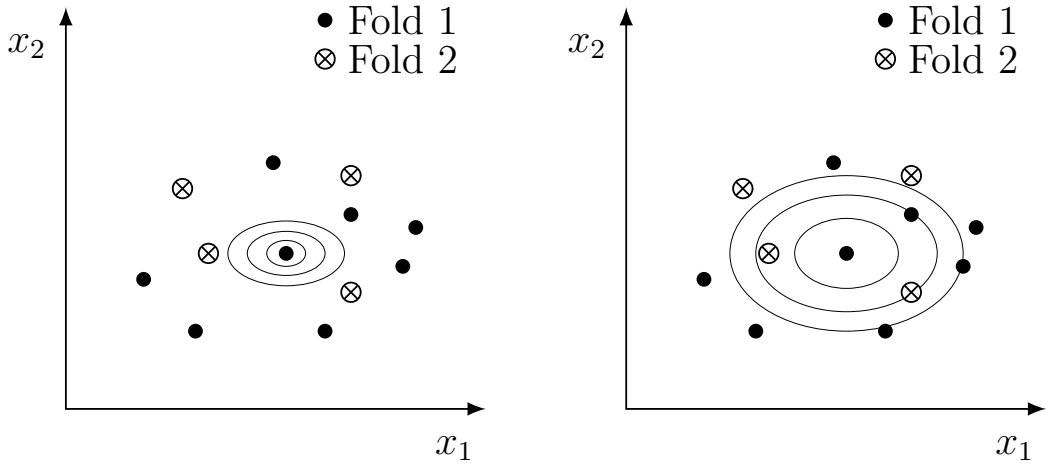


Figure 4.12 Illustration of KDE. The kernel centered at the point $x_i^{(j)}$ (marked in red on the figure) assigns a probability density to its neighborhood. Notice that the kernel is more spread out in the directions of higher data variance.



(a) Bandwidth too small

(b) Optimal Bandwidth

Figure 4.13 Example of how 2-fold cross-validation guides the choice of bandwidth. The kernel is forced to attribute a high probability density to its nearest neighbor data points from the second fold ⊗.

Once the calibration is finished, sampling is as simple as choosing a random data point $\boldsymbol{x}^{(j)}$ with uniform probability and sampling from the Kernel centered at that point. That way, KDE sampling can be seen as generating virtual particles which are perturbations of already existing particles from the training set.

KDE has many shortcomings. First, its performance degrades rapidly with dimension because

of the increasing sparsity of the data. As explained earlier, the growing sparsity of data points leads the Scott and cross-validation estimators to select extremely large bandwidths which result in over-smoothed models. One way to reduce data sparsity is to increase the number of data points N_s. The table in [46, page 319] illustrates that when modeling multivariate Gaussian data with KDE, having 4 samples in one dimension is similar to having 768 and 84,2000 samples in 5 and 10 dimensions, respectively. This exponential constraint on sample sizes makes KDE very impractical for applications with more than 5-6 dimensions. However, when working with SH coefficients, we suspect that this behavior could be avoided. In fact, we show in Appendix D that the SH coefficients of river particles are very close to their nearest neighbor when considering the Euclidean norm, even in a space of dimension 440. For this reason, we believe it is justifiable to experiment with KDE on high-dimensional populations of real particles.

The second shortcoming of KDE is that it does not consider local structures. Data is often concentrated around low-dimensional manifolds. Such situations are encountered so frequently that this is now referred to as *Manifold Hypothesis* [28]. A kernel centered at a data point spreads probability density in all directions of the SH coefficient space, even in those that are not tangent to the manifold. This results in probability density leaking outside the manifold. The leaking worsens with dimensionality since the number of bad directions increases. A solution introduced by Vincent and Bengio [34] is called Manifold Kernel Density Estimation ($\mathcal{M}$-KDE) which we now discuss.

Let $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a m-dimensional Gaussian density

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{m}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{4.49}$$

with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The $\mathcal{M}$-KDE model is

$$\phi_{1:m}(\boldsymbol{x}) = \frac{1}{N_s} \sum_{j=1}^{N_s} \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^{(j)}, \boldsymbol{\Sigma}^{(j)}), \tag{4.50}$$

where a Gaussian kernel is still centered at every data point, but now the covariances may vary between data points. Is this manner, one can design local covariances $\boldsymbol{\Sigma}^{(j)}$ to capture the low-dimensional structures of the manifolds. This is typically accomplished with a *pancake* covariance [28, 34]

$$\boldsymbol{\Sigma}^{(j)} = \boldsymbol{\Sigma}^{(j)}_{\text{local}} + h\mathbf{I}, \tag{4.51}$$

where the local covariance $\boldsymbol{\Sigma}^{(j)}_{\text{local}}$ is a singular rank-k ($k < m$) matrix whose k eigenvectors

are tangent to the local manifold at $\boldsymbol{x}^{(j)}$. The term $h\mathbf{I}$, where h is the bandwidth and $\mathbf{I}$ the identity matrix, provides small isotropic noise to ensure the Gaussian kernels are not singular. The covariance (4.51) is called *pancake* because the resulting Gaussians are spread out along all tangent directions of the manifold and very flat in the normal directions. Computing the local covariance $\boldsymbol{\Sigma}^{(j)}$ around each data point as well as adjusting the noise h constitutes the calibration step of the $\mathcal{M}$-KDE model. Sampling from this model is the same as sampling from the KDE, i.e. a data point $\boldsymbol{x}^{(j)}$ is chosen at random and virtual particles are sampled from its kernel $\mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^{(j)}, \boldsymbol{\Sigma}^{(j)})$. The main difference is that the new kernels will sample in directions that are nearly tangent of the manifolds.

To illustrate this, consider a 1-Manifold embedded in 2D space. Let $\boldsymbol{t}$ be the tangent vector to the manifold, the pancake covariance is given by

$$\boldsymbol{\Sigma} = \sigma_t^2 \, \boldsymbol{t}\boldsymbol{t}^T + h \, \mathbf{I}, \tag{4.52}$$

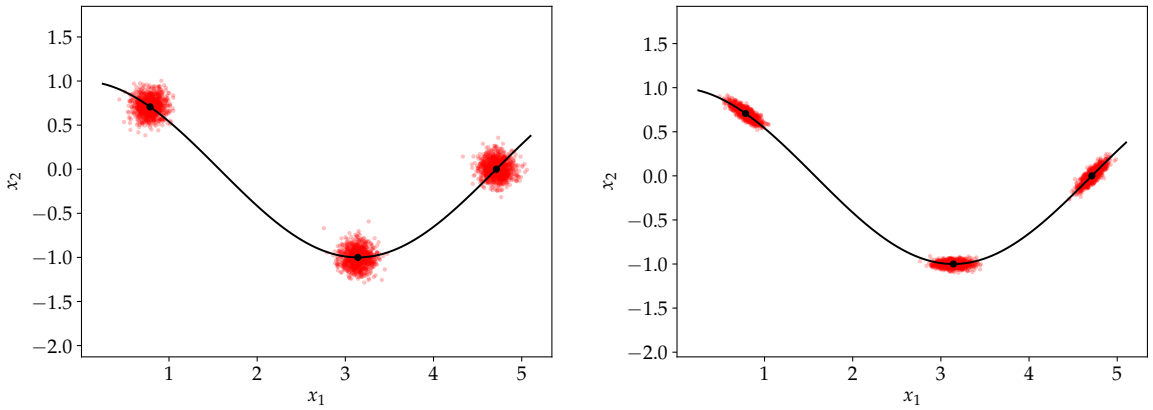
where σ_t^2 is a new parameter which controls the variance along the curve. Figure 4.14 shows how the choices of σ_t^2 and h affect the samples of virtual particles. We observe that the local covariance matrix allows one to sample along the tangent direction of the manifold, which is not guaranteed when one only considers the $h\mathbf{I}$ covariance.

In this contrived example, the analytical expression of the manifold is known so one can compute its tangent vector. With real data, one does not have this luxury and must resort to estimating the local covariance. Similar to what was done in [34], we shall select the k nearest Euclidean neighbors of every data point $\boldsymbol{x}^{(j)}$ and use them to compute a rank-k covariance matrix at every data point. The eigenvectors of this matrix are approximations of the manifold tangent vectors and the eigenvalues indicate variances in those directions. Let $n_k(j) \subset \{1, 2, \ldots, N_s\}$ be a list containing the index of the k Euclidean nearest neighbors of $\boldsymbol{x}^{(j)}$ and let $\mathbf{1} \in \mathbb{R}^k$ be a vector of ones, we define

$$\boldsymbol{\Sigma}^{(j)} = \frac{1}{k}\Big(\mathbf{X}\big(:, n_k(j)\big) - \boldsymbol{x}^{(j)}\mathbf{1}^T \Big)\Big(\mathbf{X}\big(:, n_k(j)\big) - \boldsymbol{x}^{(j)}\mathbf{1}^T \Big)^T + h\mathbf{I}. \tag{4.53}$$

Though $\mathcal{M}$-KDE is a lot more flexible than KDE, it also requires more memory storage. Indeed, KDE only needs to store the coordinates of each data point, while $\mathcal{M}$-KDE requires to store information about the local covariances at each data point in addition to their coordinates.

Using isotropic noise is standard for multiple machine learning tasks but one must be careful when working with SH coefficients, the reason being that SH coefficients of higher indices



(a) $\sigma_t^2 = 0$ and $h = 10^{-2}$

(b) $\sigma_t^2 = 10^{-2}$ and $h = 10^{-3}$

Figure 4.14 Sampling points using $\mathcal{M}$-KDE with various covariances.

must have small amplitudes and variances compared to the first coefficients. Therefore, the anisotropic noise should not be larger than the smallest standard deviation of the SH coefficients. An alternative approach would be to use anisotropic noise, but doing so, an important property of the matrix $\mathbf{\Sigma}^{(j)}$ is lost. It can be shown that the eigenvectors of $\mathbf{\Sigma}^{(j)}$ with isotropic noise only depend on $\mathbf{\Sigma}^{(j)}_{\text{local}}$ and not on h. Moreover, with isotropic noise, the eigenvalues of $\mathbf{\Sigma}^{(j)}$ are equal to the eigenvalues of the local covariance $\mathbf{\Sigma}^{(j)}_{\text{local}}$ plus the noise h. This property allows for fast computations of the log-likelihood for any given h, making the linesearch over all h values far less expensive. With anisotropic noise, each value of h leads to drastically different eigen-decompositions of $\mathbf{\Sigma}^{(j)}$.

4.5 Summary

In this chapter, we have described several populations of manufactured particles that follow simple distributions. These will be used for the validation of the modeling steps decribed in Figure 4.1 thanks to their simple geometries and their known exact statistical distributions. Moreover the preprocessing step, including normalization, PCA, and clustering was explained in-depth. Finally, the multivariate Gaussian, KDE, and $\mathcal{M}$-KDE statistical models were all introduced along with their assumptions and limitations. The contributions from this chapter are as follows:

1. Introduction of manufactured particle populations with known statistical distributions;
2. An in-depth and intuitive description of the PCA procedure applied to the SH coefficients. More precisely, the clay modeling analogy is adapted to the principal compo-

nents, see (4.14);

3. Implementation of an independence test between the size and intrinsic geometry of particles based on the Mutual Information and the Spearmann correlation.

CHAPTER 5 NUMERICAL RESULTS

In this chapter, we present some numerical results to illustrate the generative process in action. More precisely, the PCA, clustering, and statistical models from the last chapter are used on manufactured and river particles. Manufactured particles are used for the validation of the clustering and statistical models since their exact geometries and distributions are known. The Gaussian Mixture (GM) clustering algorithm applied to man-made populations manages to identify the subpopulations of prolates, oblates, spheroids, cubes and cylinders. The Kernel Density Estimation also shows promise for generating virtual diamond and box particles. River particles are studied afterwards as a way to illustrate how the statistical models perform in real-life applications. We discover that GM identifies two subpopulations with very distinct geometries. Moreover, by sampling virtual river particles from a single $\mathcal{M}$-KDE kernel centered around a river particle, evidence is provided that the SH coefficients concentrate near low dimensional manifolds. We finally suggest that understanding and estimating those manifolds will be key to develop the next generation of generative models.

5.1 Manufactured Particle Populations

The results on manufactured particles generated in Section 4.1 are discussed first. Even though these datasets are not representative of real-life grains, they allow for the validation of the clustering and data generation processes.

5.1.1 PCA

The principal component analysis is currently implemented using the `decomposition.PCA` class of the Scikit-Learn Python Library [58]. To select the optimal number of principal components, the cumulative variance (4.26) must be computed for multiple values of m, and the values achieving the 99% threshold of cumulative variance are selected, see Table 5.1. There seems to be a slight correlation between the selected values of m and the number of geometrical degrees of freedom for each population, which suggests that high-dimensional manifolds must be embedded in a larger space $\mathbb{R}^m$ to be represented faithfully. The PCA modes for some of these populations can be visualized. For simplicity, the term *Initial Shape* from (4.25) shall be noted

Table 5.1 – Cumulative variance with respect to m for the populations of manufactured particles. The selected values of m are indicated with *.

	Spheroid	Prolate	Oblate	Cubes	Cylinders	Boxes	Diamond
Geometrical Degrees of Freedom							
	3	3	3	3	3	5	5
Cumulative Variance							
q=1	0.775	0.760	0.815	0.517	0.800	0.633	0.625
q=2	0.998*	0.997*	0.997*	0.989	0.992*	0.834	0.816
q=3	0.999	0.999	0.999	0.994*	0.998	0.920	0.948
q=4	0.999	0.999	0.999	0.999	0.999	0.982	0.990*
q=5	0.999	0.999	0.999	0.999	0.999	0.988	0.994
q=6	1	0.999	0.999	0.999	0.999	0.992*	0.996
q=7	1	1	1	0.999	1	0.995	0.997

$$r_0^{(j)}(\theta,\phi) := c_0^{(j)} \left(Y_0(\theta,\phi) + \sum_{i=1} \mu_i \, Y_i(\theta,\phi) \right). \tag{5.1}$$

Figure 5.1 illustrates the initial shape $r_0^{(j)}(\theta,\phi)$ obtained with PCA on the prolate population. We observe that the initial shape of clay intuitively resembles an ellipsoid. The first two PCA modes are shown in Figure 5.2. The gray surface represents the initial shape while the colored surfaces represent $r_0^{(j)}(\theta,\phi) + 0.2\sqrt{4\pi}\, X_i(\theta,\phi),\ i = 1,2$. The perturbations induced by the PCA modes must be interpreted as the difference between the gray and colored surfaces. It appears that the first mode elongates the initial shape while making it flatter by compressing it in the x-direction. The second mode compresses the particle in the z-direction, making it less elongated, but also stretches it in the y-direction, making it flatter.

5.1.2 Clustering

As previously discussed, clustering on the SH coefficients has the potential to identify aggregates of particles which look alike independently of their size. Before applying clustering on real particles, the algorithms must be validated using man-made populations by checking if they are capable of separating the different families. The following experiment consists of running clustering on the first 4 principal components of a dataset containing 5,000 particles equally sampled from the prolates, oblates, spheroids, cubes, and cylinders subpopulations. To demonstrate that clustering is potentially viable for real particles, it is primordial that the algorithm uncovers those five underlying subpopulations.

Firstly, the K-Means algorithm is implemented with class `cluster.KMeans` from the Scikit-

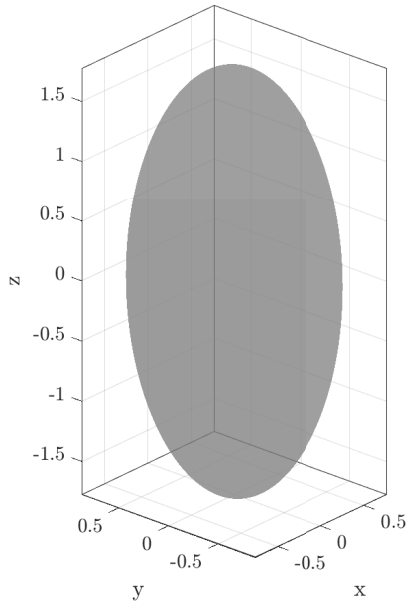
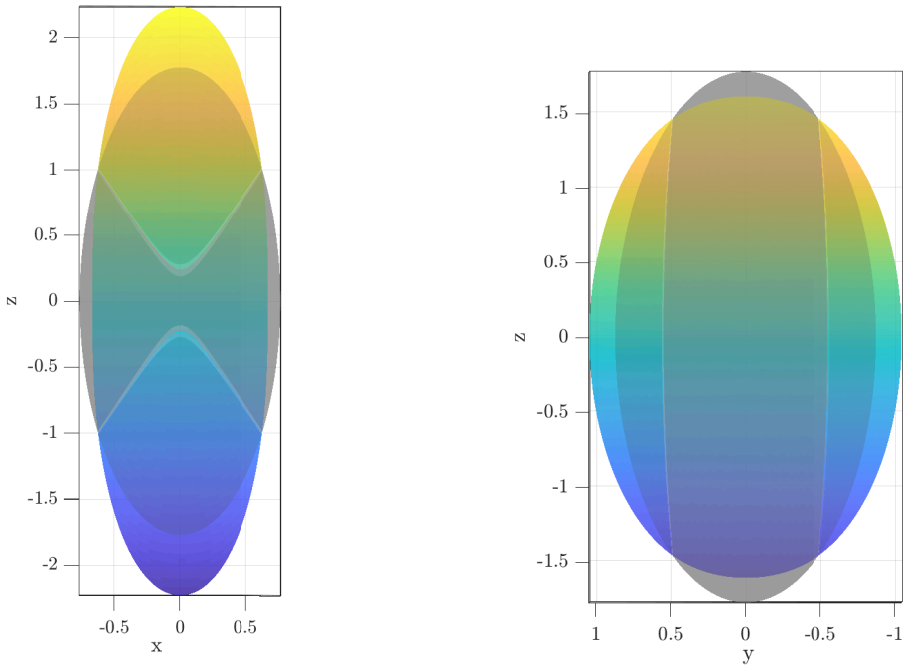
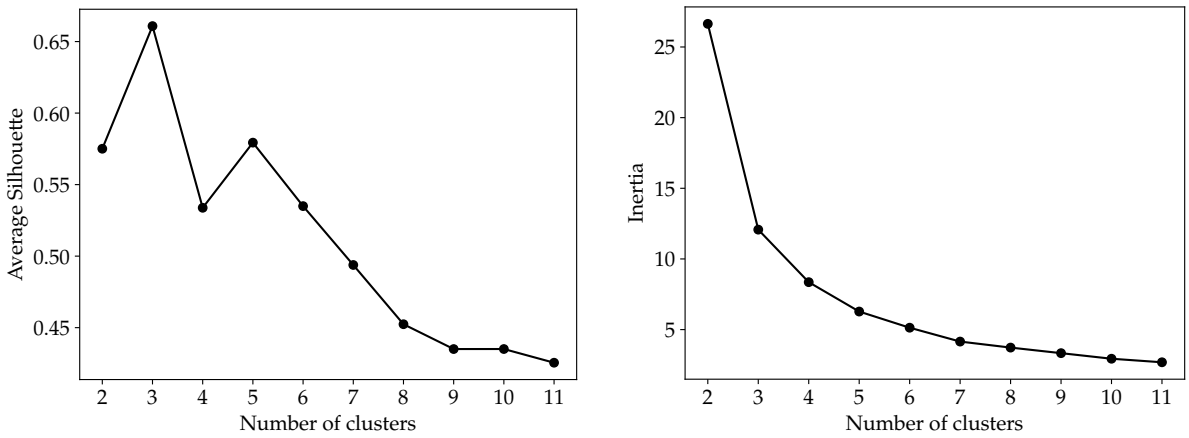


Figure 5.1 Initial shape $r_0^{(j)}(\theta, \phi)$ computed on the prolate population.



(a) $r_0^{(j)}(\theta, \phi) + 0.2\sqrt{4\pi}\, X_1(\theta, \phi)$ (b) $r_0^{(j)}(\theta, \phi) + 0.2\sqrt{4\pi}\, X_2(\theta, \phi)$

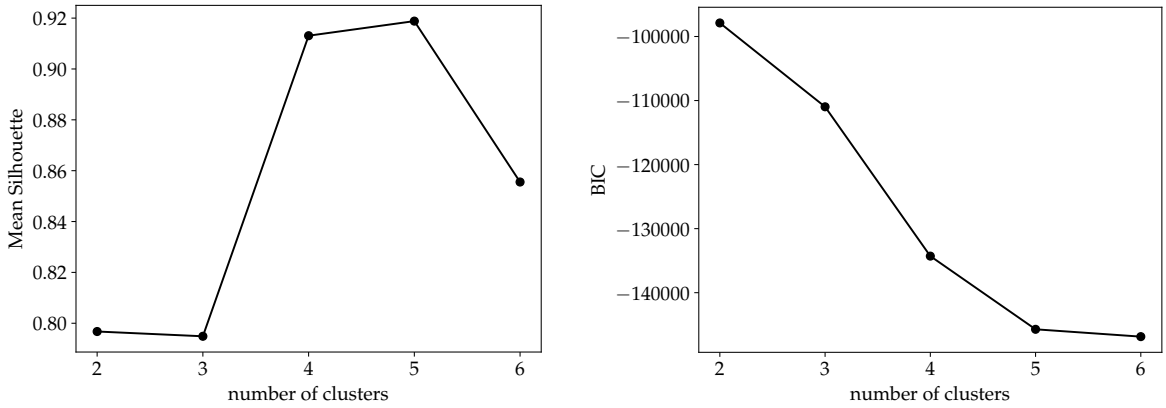
Figure 5.2 First two PCA modes on the prolate population.



(a) Silhouette with K-Means

(b) Inertia with K-means

Figure 5.3 Selection of N_c for K-Means.



(a) Silhouette with GM

(b) BIC with GM

Figure 5.4 Selection of N_c for the Gaussian Mixture algorithm.

Learn Library with the hyperparameter `n_init` set to 10. The hyperparameter `n_init` enables the K-Means algorithm to be run multiple times, so that the result with minimal average inertia can be selected. It is primordial to set it to a high value, considering that the initialization of the cluster centroids is random and that the predicted clusters are highly sensitive to the initialization. To select the optimal number of clusters, the silhouette (4.36) and average inertia (4.37) are then computed for several values of N_c, see Figure 5.3. We observe a maximum of the average silhouette at $N_c = 3$ and the *elbow* in the inertia graph at $N_c = 3$. Looking closely at the results, we discovered that K-Means combined the spheroids, cylinders and cubes families into a single cluster. A glance at Figure 5.5 demonstrates that the clusters are not spherical, which could explain the failure of K-Means.

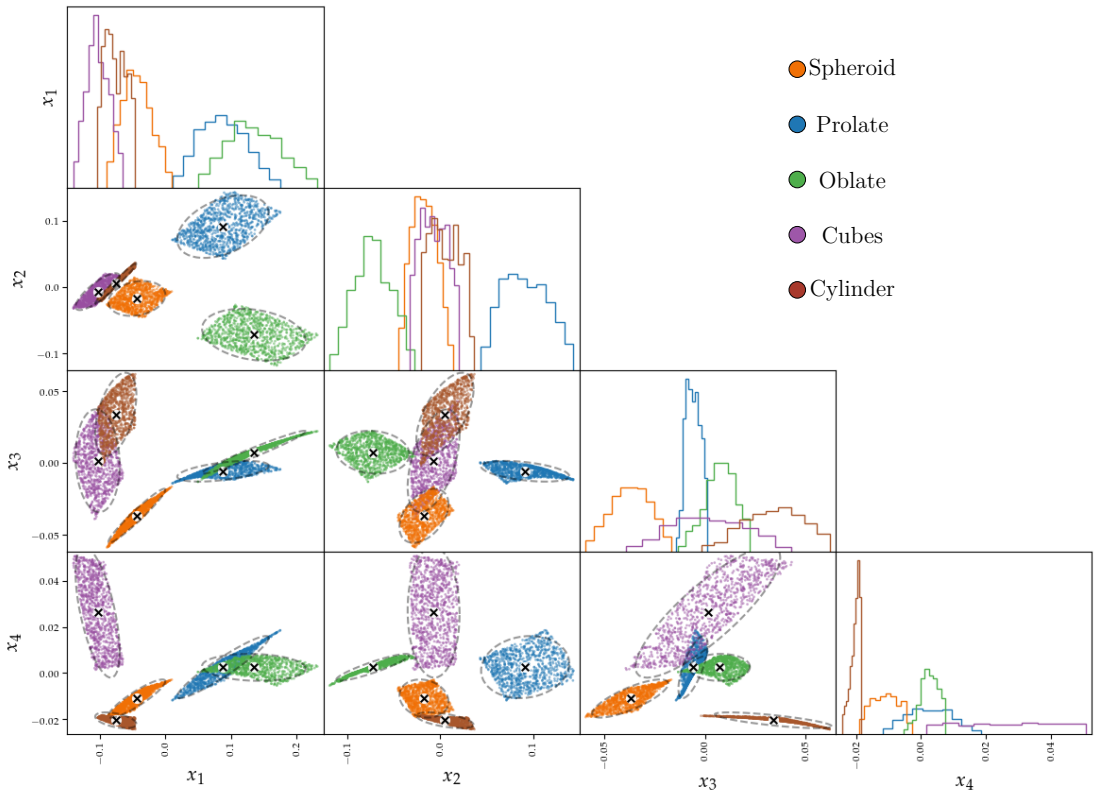
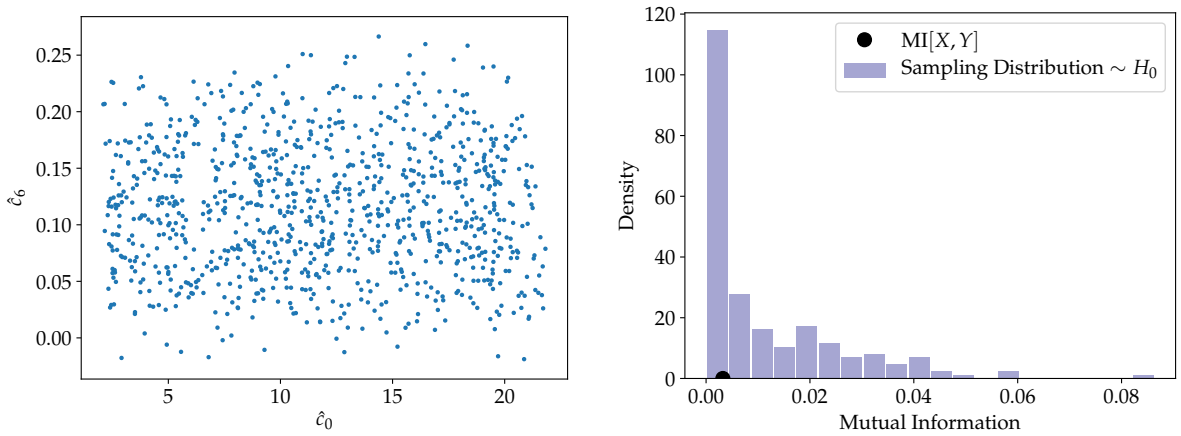


Figure 5.5 Clusters predicted by the GM algorithm. Dashed lines represent 95% confidence ellipses of the Gaussian for each cluster. Centroids are represented with a "x".

To remedy this fact, the GM clustering algorithm is applied using the Scikit-Learn class `mixture.GaussianMixture` with the hyperparameter `n_init` once again set to 10. Selection of N_c is now based on the silhouette (4.36) and the BIC (4.38), see Figure 5.4. It seems that $N_c = 6$ yields a minimum in the BIC and $N_c = 5$ results in the highest silhouette. In scenarios where BIC and silhouette do not agree on the optimal choice of N_c, one must visualize the data to obtain additional evidence as to which N_c is the most appropriate. Looking once again at the scatter plots in Figure 5.5, we clearly observe the presence of $N_c = 5$ clusters. By setting $N_c = 5$, the GM is able to identify all five subpopulations with no errors. This promising result illustrates how clustering in the space of relative perturbations can uncover subsets of particles with similar geometries.

5.1.3 Statistical Model

The diamond and box families are datasets of particles with rich geometries that still only require a small number of principal components, i.e. 4 and 6 respectively. In light of this, these



(a) Joint distribution

(b) Sampling distribution for MI

Figure 5.6 Independence test between $\widehat{c}_0$ and $\widehat{c}_6$ for the box population.

populations can be considered for the validation of the density estimation model. Even though the dimensions of the manufactured datasets are far smaller than real particle populations, studying them is justifiable considering that a model that fails on diamonds and boxes is unlikely to perform well on real particles.

Before generating virtual particles, the independence hypothesis (H_0) between size and intrinsic geometry must be validated. By construction, it is expected to hold on both diamonds and boxes populations. Note that this hypothesis does not necessarily need to be tested using the principal components. In fact, proving that $\widehat{c}_0$ is independent of $\left(\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_{d-1}\right)$ is sufficient to prove (4.41). Figure 5.6 exhibits a typical result of MI independence test. The estimated Mutual Information score is 0.0032 with a p-value of 55% while the Spearman correlation is -0.0025 with a p-value of 94%. These two measurement suggests that the coefficients are independent. Results for other coefficients of the boxes and diamonds populations are similar.

The following step is to fit the distributions of the size coefficient $\widehat{c}_0$ and the principal components $x_i^{(j)}$. The size distribution is fitted with a uniform distribution, while the joint probability density of the principal components is fitted with KDE. The Kernel Density Estimation is implemented using the Scikit-Learn class `neighbors.KernelDensity` with a Gaussian kernel and the optimal bandwidth is selected by K-fold cross-validation with $k = 4$, see Section 4.4.3. The results of the generative model on the box population are illustrated in Figure 5.7.

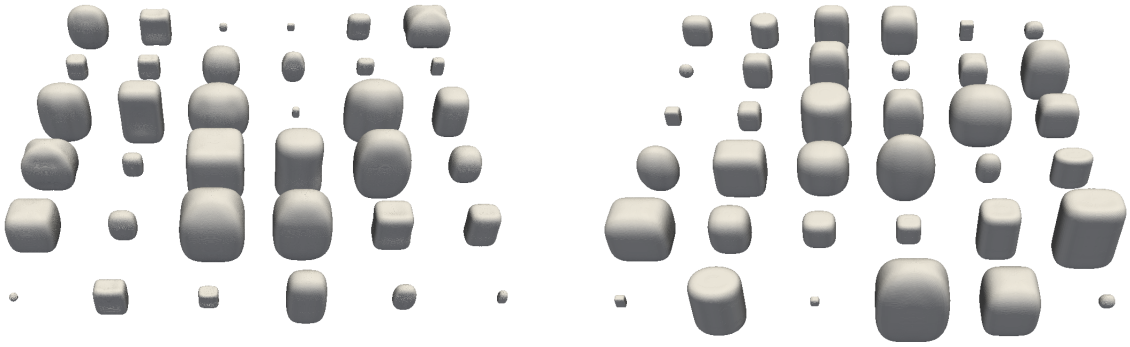
We observe in Figures 5.7(a) and 5.7(b) that the generative model manages to create realistic virtual boxes. However, when looking at the histograms of the shape descriptors, subtle

differences between particles become apparent. The particle size, elongation, and flatness follow similar distributions while roundness and convexity distributions exhibit some major dissimilarities. The roundness histogram suggests that the generative model is unable to create round particles or particles with sharp edges. On the other hand, according to the convexity histogram, virtual boxes tend to be less convex than real boxes. Two main error sources are suspected to cause these observations. Firstly, KDE is only an approximation of the true distribution. This error can however be arbitrarily reduced by increasing the size of the dataset, which is easily done considering the particles are manufactured. Secondly, the value of m is selected using the cumulative variance which is related to the reconstruction loss (4.24) by an affine relation. This reconstruction loss does not involve any derivatives while curvature and convexity are highly sensitive to the first and second order derivatives of the surface. To reduce errors in roundness and convexity, it could be necessary to build a selection criterion for m that includes derivatives. More investigation on the subject is required.

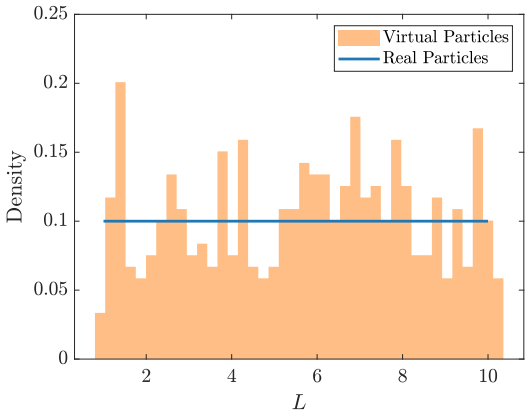
The same generative model is also applied to the diamond population, which is also of interest because, unlike boxes, they are highly non-convex and exhibit sharp corners. Figures 5.8 illustrates the results of the generative model on this manufactured population. Looking at Figures 5.8(a) and 5.8(b), it once again appears that the model captures the geometrical patterns of the original population. It is important to note that the model succeeds even though the SH representation is subject to defects such as large surface ripples. This is because the model simply learns to reproduce the patterns it is trained on. When the particles given to the model are riddled with defects, a well fitted model is able to generate new particles that share those same defects. Looking at the histograms of roundness and convexity, we observe similar behaviors as with the virtual boxes. Once again the model is not able to generate sharp corner as indicated by the lack of virtual diamonds with small roundness and the convexity of virtual diamonds is once again biased downward.

5.2 Real Particle Population

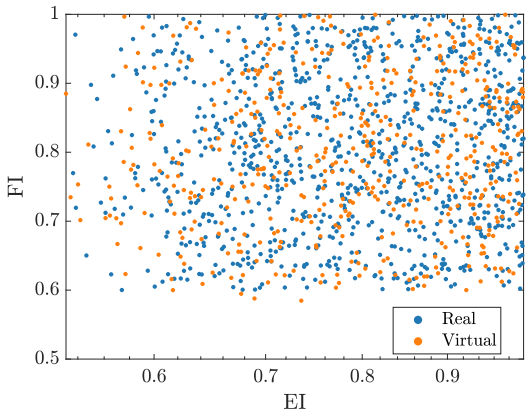
In this section, the clustering and generative algorithms which were previously validated, are now executed on river particles. The following outcomes are more representative of the viability of the methodology on real-life applications. These specific grains are chosen for the reason that they are rounder and more convex than asphalt, rouge, and margelle particles, which implies that their SH coefficients converge much faster, resulting in less principal components selected by PCA.



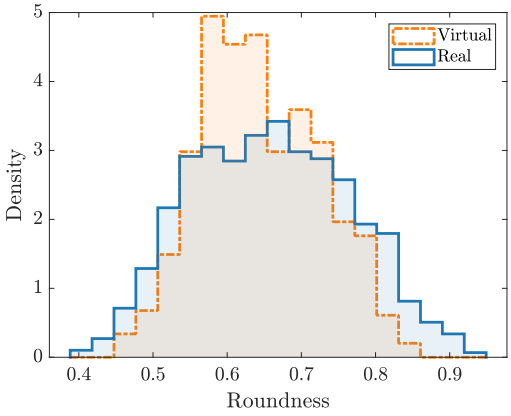
(a) Virtual (b) Real



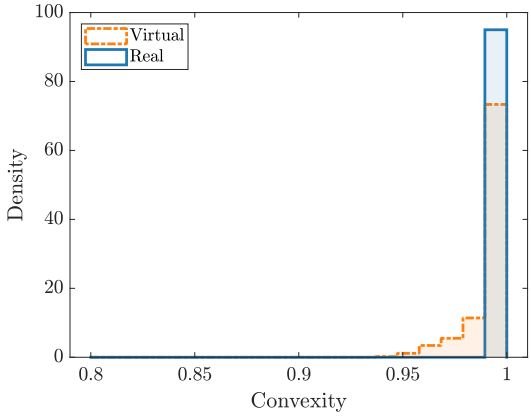
(c)



(d)

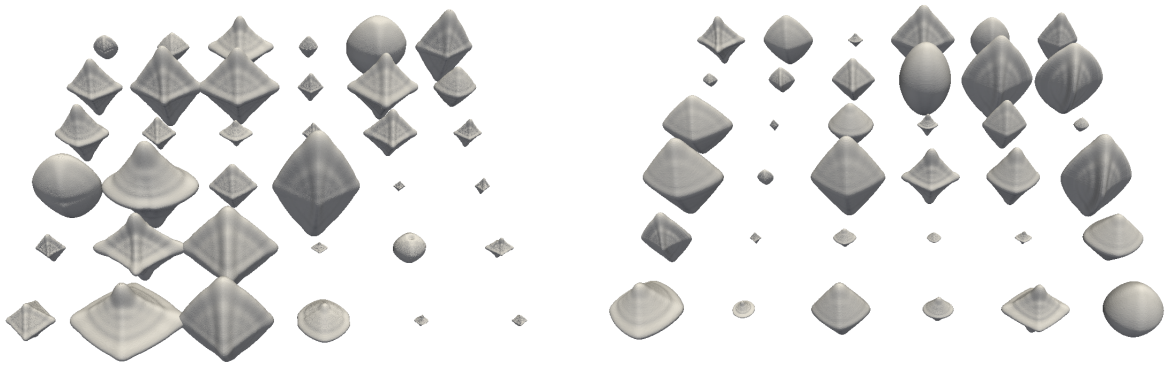


(e)

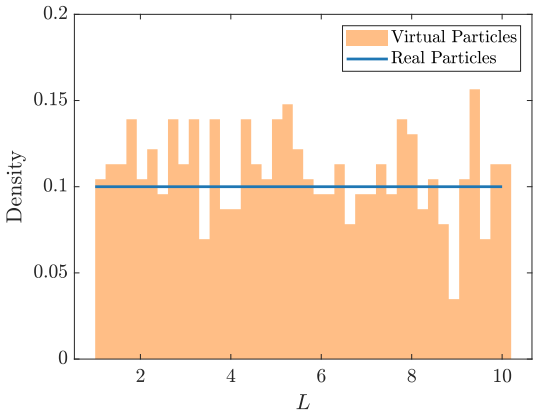


(f)

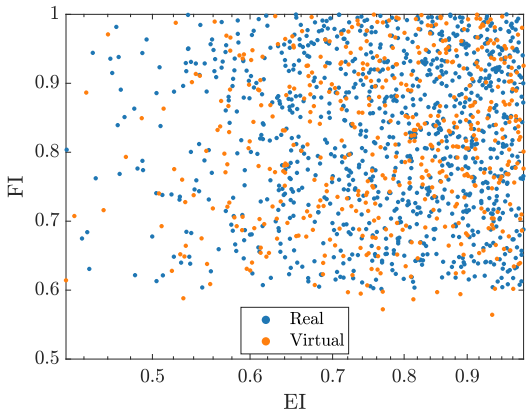
Figure 5.7 Real and virtual boxes.



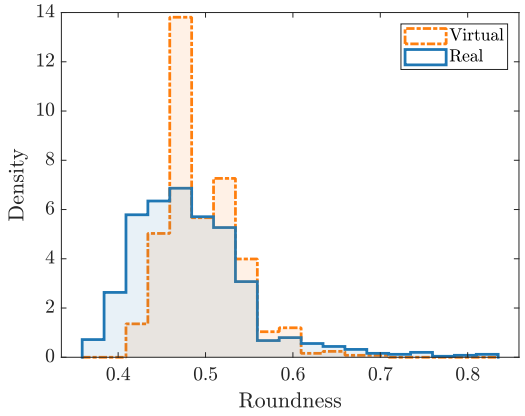
(a) Virtual (b) Real



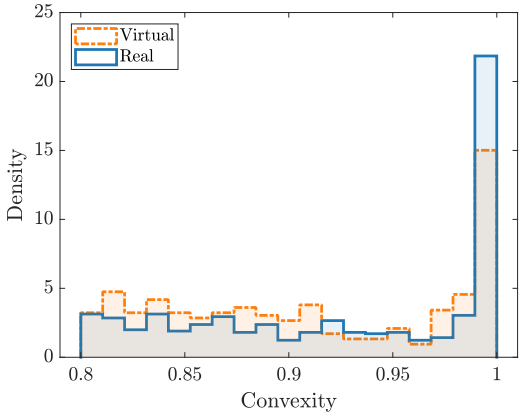
(c)



(d)



(e)



(f)

Figure 5.8 Real and virtual diamonds.

5.2.1 PCA

Table 5.2 shows the number of principal components m selected based on two thresholds on the cumulative variance. Applications that require a very faithful representation like

Table 5.2 – Value of m selected based on the the 95-99% thresholds of cumulative variance on river particles.

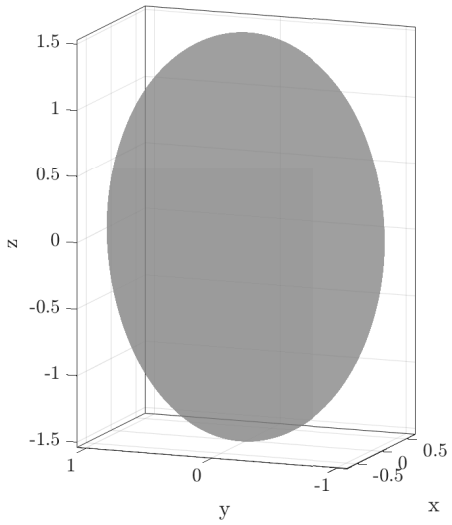
CV	Selected m
95%	28
99%	64

generative models should use the 99% threshold while applications that only need a coarse representation of shape, i.e. clustering, should use the 95% threshold. Three PCA modes of the river population are visualized in Figure 5.9. The first two modes are similar to the ones of the prolate populations from Figure 5.2, while the 10th mode is harder to understand. Considering that all PCA modes work in tandem to sculpt particles, a plot of only three modes is nevertheless not informative enough to understand the role of every single mode.

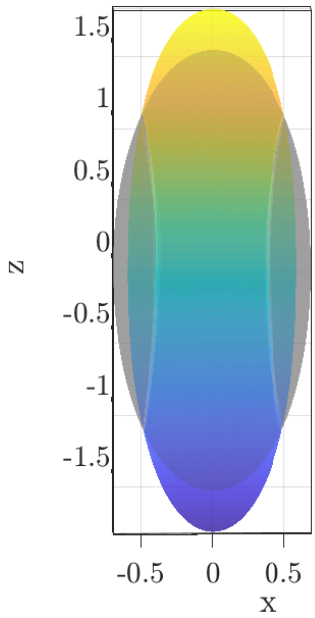
5.2.2 Clustering

When applying clustering algorithms on the river population, 28 principal components are used. The 95% threshold of cumulative variance is chosen because fine details may not be critical to determine the key geometrical features of subpopulations. A brief look at a scatter plot of the principal components yields evidence that the clusters are non-spherical, which motivates the use of the GM algorithm. See Figure 5.10 for the selection of N_c based on the silhouette (4.36) and the BIC (4.38). Both silhouette and BIC suggest that $N_c = 2$ is the optimal choice, which is confirmed by the results of Figure 5.11. These two clusters are especially apparent in the scatter plot of x_1 and x_{10}.

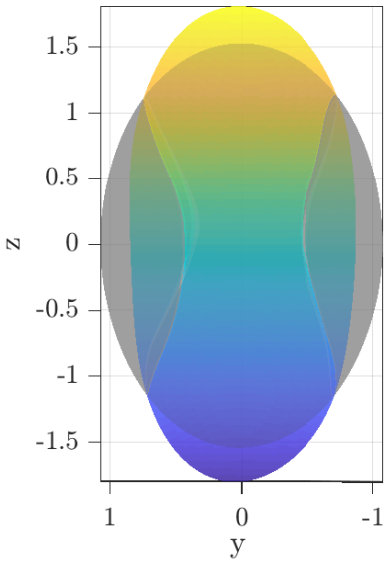
As suggested by looking at the first column of the scatter plots in Figure 5.11, Gaussian Mixture discovers a cluster of particles that tend to be smaller in average e.g. the cluster K_2. This is a pure coincidence since the clustering algorithm is oblivious to the concept of size. The second column of scatter plots demonstrates that the first principal component x_1 plays a big role in the separation of the two clusters. Figure 5.9 shows that the associated perturbation flattens and elongates the particles. Because of this, we expect K_2 to contain flatter and more elongated particles than K_1. When looking at the other principal components, it becomes harder to distinguish the clusters as they appear to be superimposed in the scatter plots. However, the variances of the second cluster are systematically higher than for the first one, which suggests that K_2 contains a large range of particles that are highly non-spherical. Those two observations are consistent with Figures 5.12 and Figure 5.13, which exhibit the STM representations and classical shape descriptors of particles from both clusters. It is seen that the particles from K_2 have a smaller flatness index, sphericity



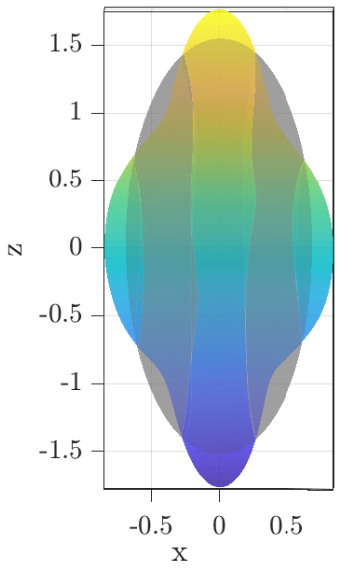
(a) Initial shape $r_0^{(j)}(\theta, \phi)$ of river population



(b) $r_0^{(j)}(\theta, \phi) + 0.2\sqrt{4\pi}\, X_1(\theta, \phi)$



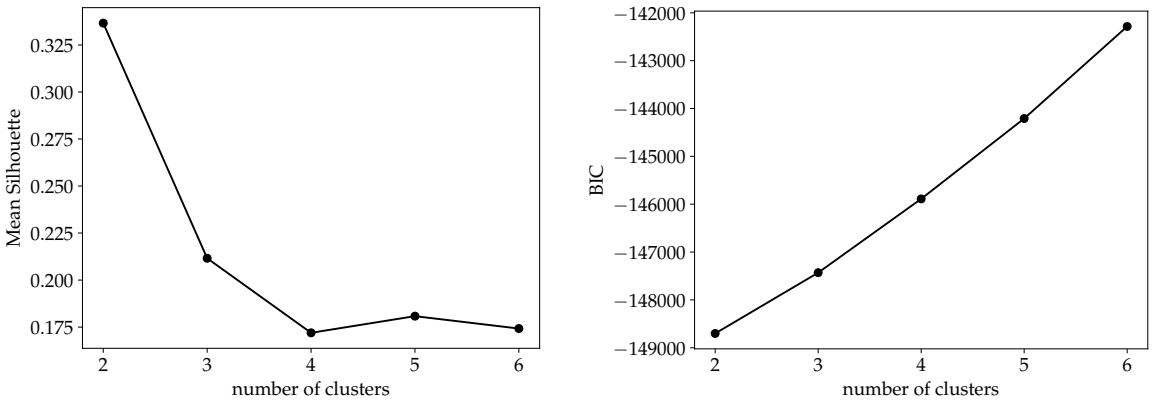
(c) $r_0^{(j)}(\theta, \phi) + 0.2\sqrt{4\pi}\, X_2(\theta, \phi)$



(d) $r_0^{(j)}(\theta, \phi) + 0.2\sqrt{4\pi}\, X_{10}(\theta, \phi)$

Figure 5.9 PCA mean and modes on the river population.

and roundness. This early result demonstrates the potential of clustering algorithms for the identification of subpopulations of particles within a dataset obtained with micro-computed tomography.



(a) Silhouette with GM

(b) BIC with GM

Figure 5.10 Selection of N_c for Gaussian Mixture on river particles.

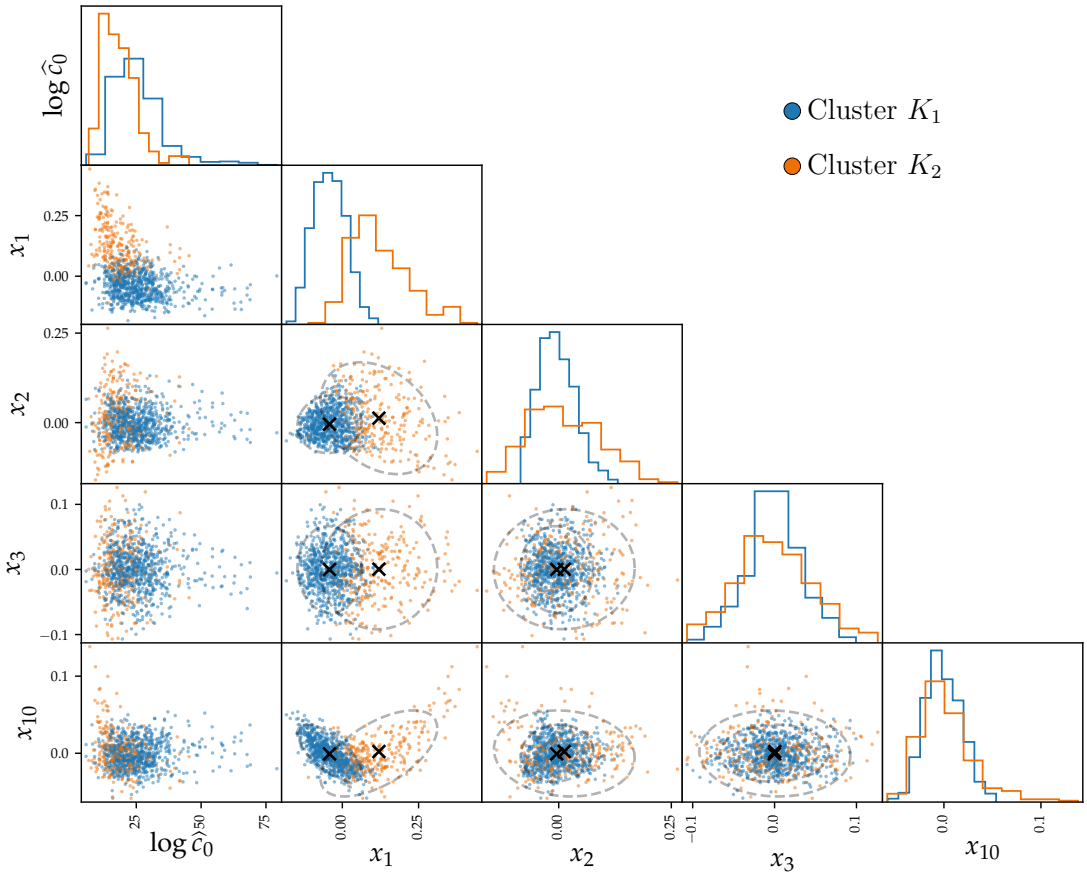
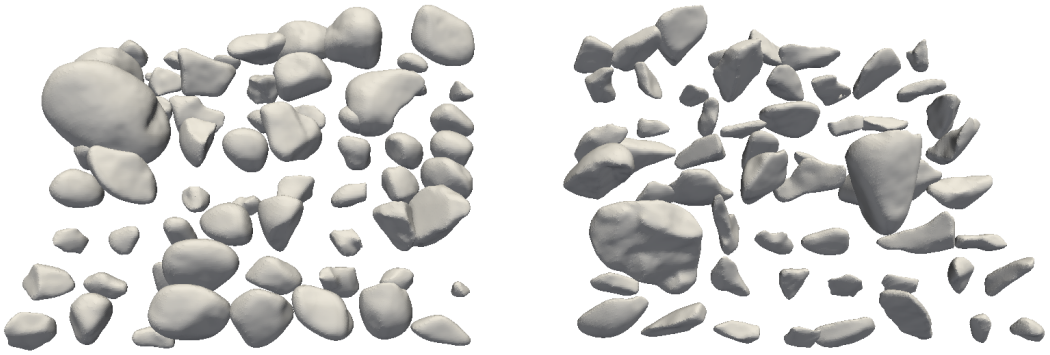
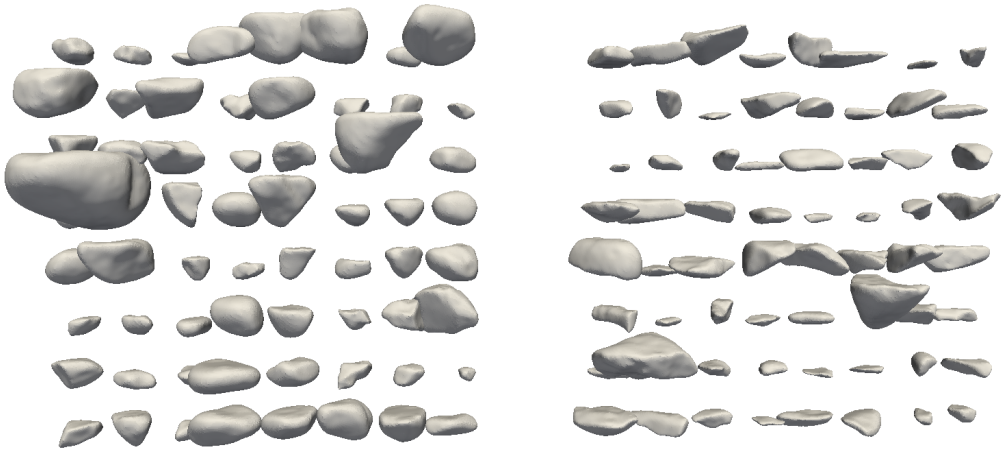


Figure 5.11 Clusters predicted by the GM algorithm. Dashed lines represent 95% confidence ellipses of the Gaussian of each cluster. Centroids are represented with "x".



(a) angle 1



(b) angle 2

Figure 5.12 Comparison of 64 particles from each cluster. Left particles come from K_1 while right particles come from K_2.

5.2.3 Statistical Model

Independence of principal components: The first hypothesis to test on the river population is the normality hypothesis, the reason being that the normality assumption is often used to justify sampling the principal components of the data independently.

To verify if the data of the normalized SH coefficients $\widehat{c}_i$ is normally distributed, both the D'agostino and Shapiro-Wilk tests are executed on all 441 normalized SH coefficients with

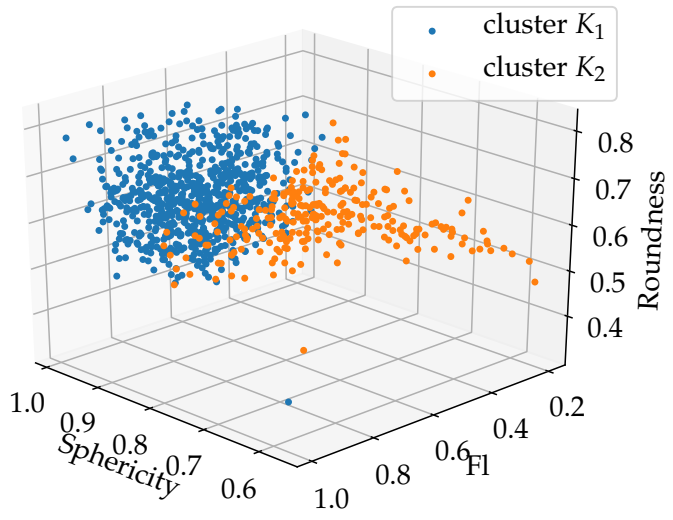
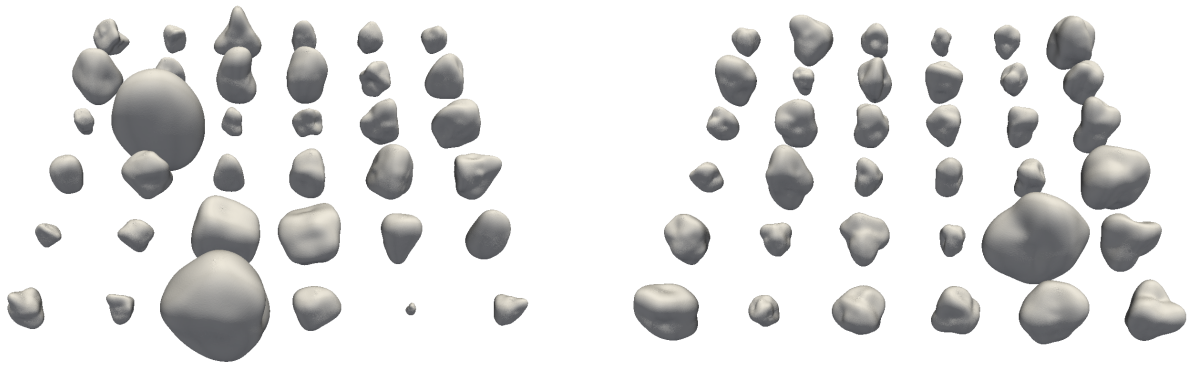


Figure 5.13 Shapes descriptors of the river particles from both clusters.

the significance $\alpha = {}^{0.05}/_{441}$. Dividing 0.05 by 441 ensures that the probability of falsely rejecting *at least* one hypothesis is less than 0.05. This very conservative approach to multiple hypotheses testing is referred to as the Bonferri method [46, page 166]. On our data, each test approximately rejects 98% of the normality hypotheses over all 441 coefficients. This strong empirical evidence forces us to reject the normality assumption. Since the data is not normally distributed, we cannot justify sampling each principal component independently.

Another experiment that can be done to confirm the dependence of principal components is to build a generative model that assumes their independence. The virtual particles sampled from such a model can then be compared to the real ones, enabling one to observe which geometrical patterns are lost when assuming independence of the principal components. The Bootstrap applied on each principal component independently can be used as a simple way to simulate sampling from such a model. Results comparing 36 SH representations of real and virtual river particles are shown in Figure 5.14. The first observation is that the virtual and real particles look similar. This could explain why the independence of the principal components is so often assumed in practice. Nevertheless, a more thorough comparison between real and virtual particles unveils the loss of two important morphologic characteristics: the smoothness of the surface and the presence of flat faces. Indeed, the virtual grains exhibit significantly larger surface ripples which make them look more akin to *popcorn* rather than smooth river particles. Additionally, the flat faces that can be observed on some of the real



(a) Real particles (b) Virtual particles (Bootstrap)

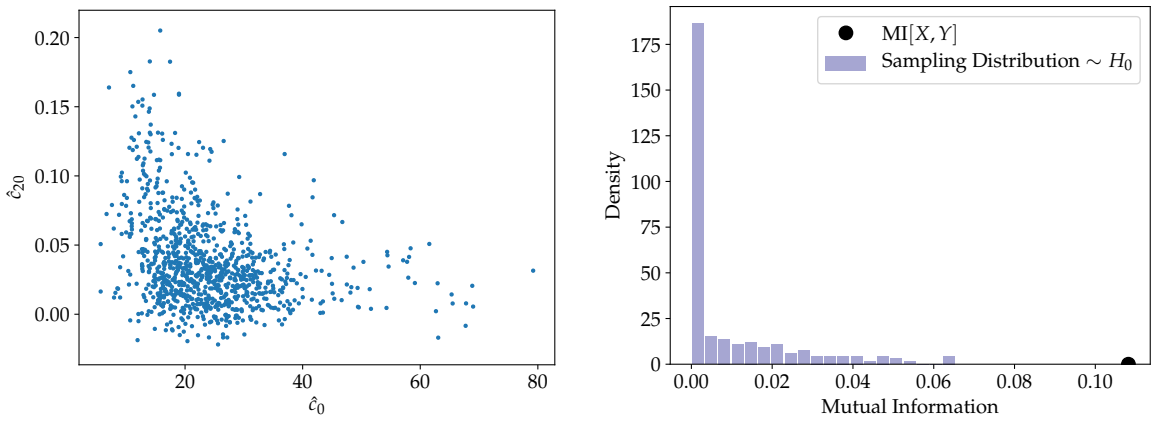
Figure 5.14 Simulating a model where the principal components are assumed to be independent.

river particles are mostly absent from the virtual assembly. This suggests that the principal components must synchronize in a specific way to sculpt flat faces.

Independence between size and geometry: Secondly, the independence between size and geometry (H_0) is to be tested. On the river population, we find that some relative perturbations are not independent of size ($\widehat{c}_0$). For example, Figure 5.15 suggests that the coefficient $\widehat{c}_{20}$ is correlated with the particle size. The computed Spearman correlation is -0.32 with a p-value of 8.4×10^{-24} which is strong evidence for correlation. The Mutual Information score is 0.108 with a p-value of zero. Figure 5.15(b) shows that the MI is located far away from the sampling distribution, which confirms dependence.

Two alternatives can be used when the independence hypothesis is shown to be false. The first approach is to reject it and jointly fit the size and geometry distributions. This was done by Zhou et al. [19] where the authors conditioned the distribution of the principal components on the volume. The alternative is to partition the data into clusters where H_0 approximately holds. Since clustering identifies aggregates of particles that look similar independently of their size, the hypothesis should, at least, hold better on each individual cluster.

The second approach is chosen since clusters in the river population have already been identified in Section 5.2.2. Following some experimentation, we observe that, even though the independence hypothesis fails on the clusters, it holds better than on the whole dataset. To demonstrate it, we run the MI independence test between $\widehat{c}_0$ and the first 100 relative perturbations $\widehat{c}_i$, $i = 1, 2, \ldots, 100$ using on the one hand, the whole dataset, and on the other hand, the two clusters taken separately. The number of independence hypotheses that are



(a) Joint distribution

(b) Sampling distribution for MI

Figure 5.15 Example of the Mutual Information independence test between the size and intrinsic geometry of river particles.

Table 5.3 – Amount of rejected hypothesis of the Mutual Information independence test between $\widehat{c}_0$ and $\widehat{c}_i$, $i = 1, 2, \ldots, 100$.

	Whole data	cluster K_1	cluster K_2
Rejected H_0	40	9	16

rejected with significance $\alpha = 0.05$ are illustrated in Table 5.3. We observe that each cluster leads to fewer rejections of the hypothesis than when considering the whole dataset. Note that the Bonferri method, which consists of using $\alpha = {}^{0.05}/_{100}$ instead of $\alpha = 0.05$, is not useful when applying multiple MI tests due to the extremely low resolution of the p-values returned by the test.

Kernel Density Models: We now discuss the model of the joint probability density of the principal components of river particles. We previously found that normalized SH coefficients do not follow a multivariate Gaussian distribution. For this reason, the KDE is considered to estimate the probability density distribution of the principal components. However, early results of KDE with the K_1 cluster of river particles are not very promising. Figure 5.16 illustrates the selected bandwidths using both Scott's estimator and K-fold cross-validation with k = 20. We observe that the cross-validated bandwidth keeps increasing up to 0.9 as more principal components are considered. A bandwidth close to 0.9 results in an extremely biased estimation of the density distribution since the width of a single kernels is almost equal to the width of the data in every single direction. Note that values of m that go beyond 30 are not included in the graph because the density estimation provided by the

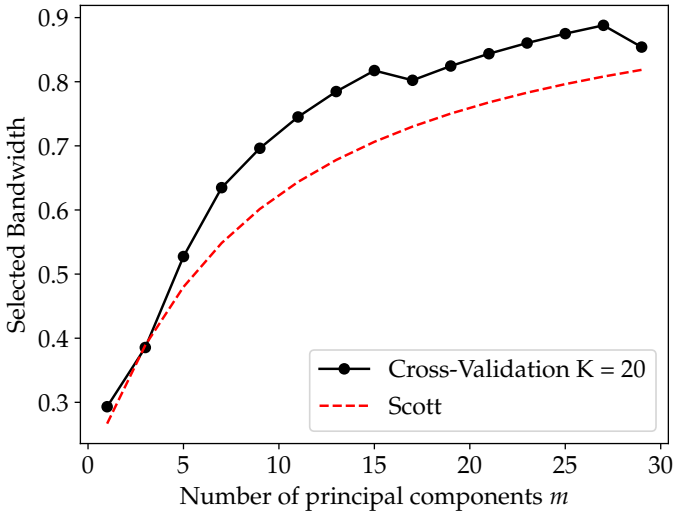


Figure 5.16 Optimal bandwidth selected with K-fold cross validation and Scott's estimator for KDE as a function of the number of principal components.

`neighbors.KernelDensity` Scikit-Learn class was becoming especially noisy for large values of m, which resulted in unstable predictions of h. We suspect that this can be attributed to the fact that the `neighbors.KernelDensity` class does not compute the exact probability density, but estimates it by storing the data points into a KD-tree [58]. However, this remains to be fully confirmed.

As discussed in Section 4.4.3, selecting the bandwidth with K-fold cross-validation forces each kernel to attribute a considerable probability density to their nearest neighbors. It is explained in Appendix D that when working with SH representations of particles, the Euclidean distance between a data point and its nearest neighbor remains very small, even in arbitrarily large dimensions. It is therefore surprising that the K-fold cross-validation selects such high bandwidths. One issue with that reasoning is the KDE algorithm *build-in* norm is that the Mahalanobis norm, not the Euclidean one. To understand this, observe that the probability density attributed by the jth kernel in (4.46) is proportional to

$$\exp\Big(-\frac{1}{2h}\|\boldsymbol{x}-\boldsymbol{x}^{(j)}\|_{\boldsymbol{\Sigma}}^2\Big).$$

It is shown in Appendix D that the Mahalanobis distance to a nearest neighbor keeps arbitrarily increasing as more dimensions are considered, which forces the kernels to increase their bandwidth accordingly.

Another potential issue with KDE is that the data may concentrate near low dimensional manifolds. If such is the case, the KDE would still fail even if enough data were used to

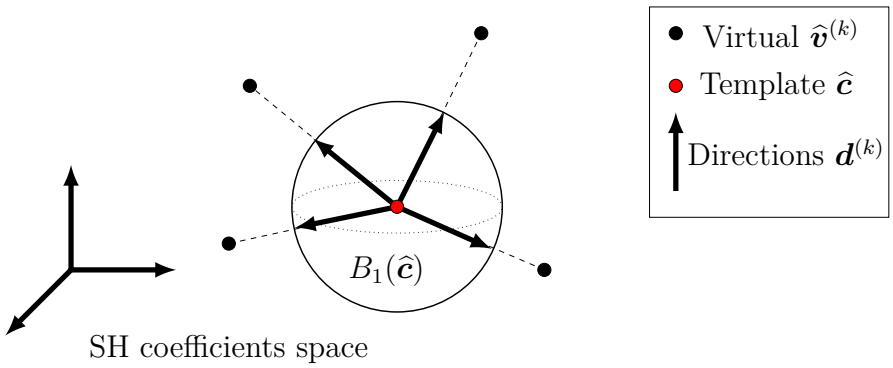


Figure 5.17 Computation of the directions $\boldsymbol{d}^{(k)}$ from a template particle $\widehat{\boldsymbol{c}}$ to the kth virtual particle $\widehat{\boldsymbol{v}}^{(k)}$ generated by the kernel. $B_1(\widehat{\boldsymbol{c}})$ refers to the Euclidean unit ball.

allow for a small optimal bandwidth. This is because, as illustrated in Figure 4.14, the KDE attributes probability density in all directions, even in those that are not tangent to the manifold, and results in samples that lie outside of the manifold.

To the best of our knowledge, no study has been conducted to date that shows that such manifolds exist for populations of real particles. As a preliminary study, the following experiment is conducted: a KDE kernel and a $\mathcal{M}$-KDE kernel are both centered at an arbitrary data point $\widehat{\boldsymbol{c}}$, which we shall refer to as the template particle. Note that we do not consider the index $^{(j)}$ of the particle since it is totally arbitrary. On one hand, the bandwidth of the KDE kernel is chosen arbitrarily, and on the other hand, the $\mathcal{M}$-KDE covariance matrix (4.53) is computed using the 20 nearest Euclidean neighbors of $\widehat{\boldsymbol{c}}$ and without considering isotropic noise ($h = 0$). New particles can be sampled from both kernels which yields the virtual coefficients $\widehat{\boldsymbol{v}}^{(k)}$. Notice that, by construction, any virtual particle sampled with $\mathcal{M}$-KDE lives on a 20-dimensional hyperplane. To get evidence that the data of river particles is indeed concentrated near low-dimensional manifolds, we examine the directions

$$\boldsymbol{d}^{(k)} := \frac{\widehat{\boldsymbol{v}}^{(k)} - \widehat{\boldsymbol{c}}}{\|\widehat{\boldsymbol{v}}^{(k)} - \widehat{\boldsymbol{c}}\|_2}, \tag{5.2}$$

which are illustrated in Figure 5.17. Looking at directions $\boldsymbol{d}^{(k)}$ not only allows one to remove any effect of the bandwidth in the KDE kernel, but it also exposes how both kernels *explore* the boundary $\partial B_1(\widehat{\boldsymbol{c}})$ of the neighborhood $B_1(\widehat{\boldsymbol{c}})$ of the template particle. Note that $B_1(\widehat{\boldsymbol{c}})$ refers to the Euclidean unit ball around $\widehat{\boldsymbol{c}}$. By moving a small distance ϵ along the directions given by both kernels, i.e. $\epsilon\, \boldsymbol{d}^{(k)} + \widehat{\boldsymbol{c}}$, we can see how both kernels explore the boundary $\partial B_\epsilon(\widehat{c})$ around the template. Figure 5.18 shows particles that are obtained by moving a small distance $\epsilon = 0.053$ along each direction. We observe that the directions obtained by

KDE yield particles with multiple ripples making them look non-physical. However, since $\mathcal{M}$-KDE explores the SH coefficient space along directions that are approximately tangent to the manifold, the resulting directions generate smoother and more realistic particles.

This result provides evidence that the SH coefficients are subject to local constraints that dictate which directions yield realistic particles and which do not. For this reason, we hypothesize that the coefficients are concentrated near a manifold $\mathcal{M}_G$, which we shall refer to as the geology manifold. The reason for this terminology is that we suspect these local constraints on variations can be attributed to perturbations that are allowed by geological processes. See Figure 5.19 for an illustration of the geology manifold. Note that the particles B, C, D, and E are all located at a fixed distance $\epsilon = 0.053$ from the template particle A. Applying non-physical perturbations cause the virtual particles B and C to lie outside of the manifold.

We leave as future work the complete implementation of the $\mathcal{M}$-KDE algorithm. This algorithm could be validated on manufactured particles for which we know the exact dimension of the manifolds their SH coefficients live on. The same method would afterwards be applied to the river particles, and in case it succeeds in generating realistic grains, it would provide strong evidence that exploiting manifold structure can improve generative models of virtual particles. Alternatively, we could experiment with advanced generative models, known to exploit manifolds structures in given datasets. For example, Variational Auto-Encoder, Generative Adversarial Networks, $\mathcal{M}$-flows Networks are flexible neural-network based generative models known to be able to estimate low-dimensional structure through the use of latent variables that represent coordinates along the manifolds [28, 35].

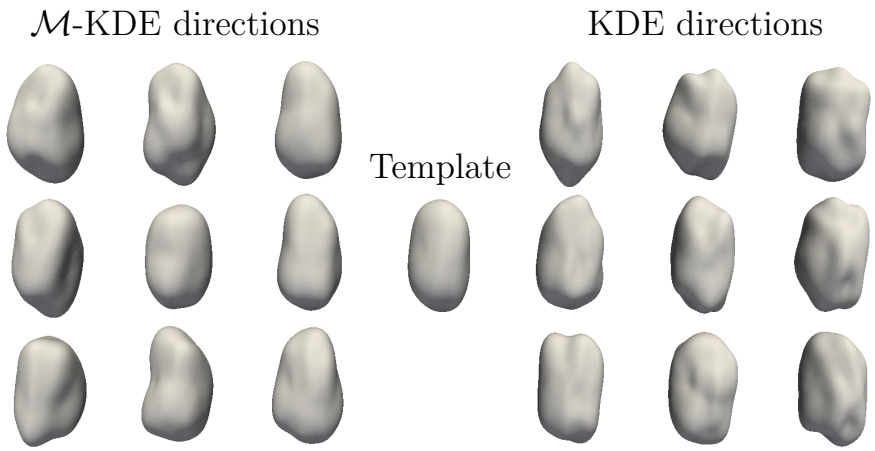


Figure 5.18 Exploration of the boundary $\partial B_\epsilon(\widehat{\boldsymbol{c}})$ around a template particle using the directions computed by $\mathcal{M}$-KDE with 20 neighbors and no noise and KDE with an arbitrary bandwidth. All virtual particles are located at a fixed distance $\epsilon = 0.053$ from the template.

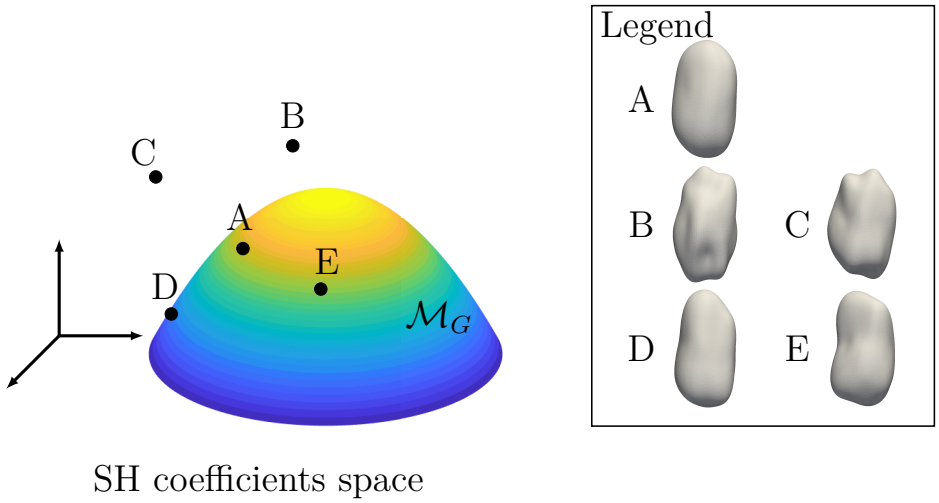


Figure 5.19 Manifold hypohtesis applied on river particles.

Geology manifold expanded around a template particle A. The virtual particles B, C, D, and E are all part of the boundary $\partial B_\epsilon(A)$ with $\epsilon = 0.053$. The particles B and C are sampled outside the manifold, giving them unrealistic shapes, while the virtual particles D and E are sampled close to the manifold, leading to more realistic particles.

5.3 Summary

The main conclusions and contributions from this chapter are as follows:

1. Validation of the GM clustering algorithm by applying it to the manufactured popula-

tions of prolates, oblates, spheroids, cubes, and cylinders. The algorithm is indeed able to perfectly identify the five distinct subpopulations;

2. Validation of the independence hypothesis H_0 (4.41) on the box population, for which it is known to hold;

3. Validation of the KDE statistical model on the box and diamond populations. It was shown that the model is able to generate virtual particles that look similar to the particles from the training data;

4. Use of the GM clustering algorithm to identify two subpopulations of river particles with distinct geometry. In fact, one cluster contained smoother and more spherical particles while the other contained particles with sharper corners and with an overall flatter shape.

5. Use of the MI independence test to prove the dependence between particle size and geometry on the river population. It was also observed that partionning the river particles into the two clusters obtained with GM would yield fewer rejections of H_0 than on the whole data. This is early evidence that clustering can simplify the statistical modeling of particles as shape and geometry could be independently approximated on each cluster separately, but this needs to be confirmed with additional examples;

6. Demonstration that the KDE models fails on high-dimensional data of river particles as cross-validation is forced to select ridiculously large bandwidths, despite the observation from appendix D that particles are close to their nearest neighbors with respect to the Euclidean norm;

7. Empirical evidence that the SH coefficients are subject to local constraints on their directions of variation. To explain this observation, it is hypothesized that the particles are concentrated near lower dimensional manifolds.

The following chapter includes concluding remarks as well as a discussion of futur research avenues.

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

6.1 Summary of Work

In this thesis, we have described a methodology to generate collections of particles based on STM representations of grains obtained with micro-computed tomography. The methodology involves several steps, the first one being the computation of the Spherical Harmonics (SH) coefficients associated with actual particles. Such calculations require linear interpolation of the particle triangulated surface and numerical integration. The SH coefficients are of importance since they allow one to represent each particle in a population with a fixed number of features, making comparisons between grains possible. Additionally, particles are completely determined by their SH coefficients which could make them useful when correlating particle morphology to DEM responses. The following step of the methodology is preprocessing the data by normalizing the SH coefficients and by applying the Principal Component Analysis (PCA). Normalization enables one to store all the information related to particle size into a single coefficient, while PCA reduces the dimensionality of all other coefficients. Afterwards, clustering algorithms can be used to identify subpopulations of particles with similar geometries independently of their size. Clustering has the potential to simplify the statistical modeling of particles by fitting a distinct model to each subpopulation. Finally, generative models are fitted on each cluster, lending one the ability to sample virtual SH coefficients which can be reconstructed into full particles.

Among the specific contributions of this research is the discovery that clustering algorithms can be applied on SH representations of particles to uncover various underlying geometries within a large population of particles. More precisely, the Gaussian Mixture is able to perfectly partition subpopulations of spheroids, oblates, prolates, cubes, and cylinders. Moreover, the same algorithm applied on a population of river particles identifies two subsets of particles with very distinct morphologies.

Another important contribution discussed in this thesis is the use of generative models that go beyond the multivariate Gaussian and Nataf transform, which are used in the geology literature. The Kernel Density Estimation (KDE) method shows promise for the task of generating superquadrics, which have a simpler geometry than real particles. Unfortunately, the same algorithm does not perform well on high dimensional data of river particles, but a modified version called Manifold KDE provides evidence that the SH coefficients are concentrated near low-dimensional manifolds. This observation could drive the next generation of generative models who can exploit those low-dimensional structures.

6.2 Future Research

The discussed methodology is subject to some limitations. The first one is the restriction to star-shaped particles. It was observed that computing the SH representation of the radial parametrization of non star-shaped particles would yield large defects on the surface of the particles. This shortcoming could potentially be overcome by considering the surface parametrization of particles described in Section 1.3 instead of their radial parametrization. The whole methodology could then be extended to representations in terms of surface parametrizations. However, some investigation would be required to determine if the surface parametrization allows for a consistent representation between particles. This is not immediately apparent since the mapping from the unit sphere to the particle surface is not unique.

A second issue is maybe the use of spherical harmonics. Indeed, these are defined globally over the unit sphere in the sense that their support is the whole sphere. Full support implies that each mode may create global defects that must be cancelled out by subsequent harmonics. This could translate into complex correlations between all SH coefficients. Using a basis with compact support, e.g. spherical wavelets, could overcome this challenge. Indeed, since these functions are hierarchical, undesirable defects introduced by a specific wavelet need only be cancelled out by the wavelet descendants. This could reduce correlations between the features of the data set as well as making the data more interpretable, considering that each wavelet is known to work on a specific region of the particle. However, it is not clear how PCA modes obtained from wavelet coefficients would differ from the same modes computed using spherical harmonics. The reason the modes may end up being the same is that the spherical harmonics can be represented with a linear combination of spherical wavelets and vice versa. The PCA algorithm being linear, both representations could end up with the same result. If such is the case, then working with spherical wavelets would be superficial considering all the machine learning algorithms are applied on the principal components and not on the original coefficients.

The next step of our research would be to fully implement the Manifold Kernel Density Estimation algorithm and to apply it to river particles. If done successfully, this method would then need to be compared to the two most popular models in the geology literature: the multivariate Gaussian and the Nataf transform. In case we observe significant improvements, this would give strong evidence that exploiting manifold structures in the framework of spherical harmonics representations of particles can yield better generative models.

It would also be important to study data sets of grains with different geometries than river

particles. Directly available to us are asphalt, rouge, and margelle particles which tend to be less convex and less smooth on average than river particles. Studying those populations could reveal if problems encountered by the statistical models are specific to a type of geometry. For example, as discussed in Section 5.2.3, smooth particles could be harder to generate despite having well behaved SH representations. Other particles that could be considered include Leighton Buzzard Sand (LBS) and Highly Decomposed Granite (HDG), which are widely used in the geology literature.

The research conducted in this thesis covers only a small aspect of a broad projet which aims at running DEM simulations in order to relate the macroscopical behavior of a granular material to specific geometric properties of its particles. It is therefore primordial to eventually run DEM simulations using the virtual assemblies created by our generative model. Considering virtual particles are completely determined by their SH representation, the SH coefficients could be potentially correlated with DEM responses. A similar experiment was done in [18], where the authors ran DEM simulations using virtual particles generated with spherical harmonics representations. Such simulations could be attempted with our generative model.

In the very long term, it would be also pertinent to explore the emerging field of geometric deep learning [62]. This field is inspired by the recent accomplishments of convolutional neural networks in image-based tasks such as object recognition and image generation. Researchers in this domain are currently extending the classical convolutional neural networks so they can work on functions defined over curved manifolds. In the geology literature, particles are mainly represented as functions defined over the unit sphere, which is why these new convolutional networks could be of interest.

REFERENCES

[1] G.-C. Cho, J. Dodds, and J. C. Santamarina, "Particle shape effects on packing density, stiffness, and strength: natural and crushed sands," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 132, no. 5, pp. 591–602, 2006.

[2] B. Sukumaran and A. Ashmawy, "Quantitative characterisation of the geometry of discrete particles," *Geotechnique*, vol. 51, no. 7, pp. 619–627, 2001.

[3] J. Yang and X. Luo, "Exploring the relationship between critical state and particle shape for granular materials," *Journal of the Mechanics and Physics of Solids*, vol. 84, pp. 196–213, 2015.

[4] P. A. Cundall and O. D. Strack, "A discrete numerical model for granular assemblies," *geotechnique*, vol. 29, no. 1, pp. 47–65, 1979.

[5] C. Nouguier-Lehon, B. Cambou, and E. Vincens, "Influence of particle shape and angularity on the behaviour of granular materials: a numerical analysis," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 27, no. 14, pp. 1207–1226, 2003.

[6] P. W. Cleary and M. L. Sawley, "DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge," *Applied Mathematical Modelling*, vol. 26, no. 2, pp. 89–111, 2002.

[7] B. Saint-Cyr, J.-Y. Delenne, C. Voivret, F. Radjai, and P. Sornay, "Rheology of granular materials composed of nonconvex particles," *Physical Review E*, vol. 84, no. 4, p. 041302, 2011.

[8] J. Wang, H. Yu, P. Langston, and F. Fraige, "Particle shape effects in discrete element modelling of cohesive angular particles," *Granular Matter*, vol. 13, no. 1, pp. 1–12, 2011.

[9] H. Wadell, "Volume, shape, and roundness of rock particles," *The Journal of Geology*, vol. 40, no. 5, pp. 443–451, 1932.

[10] F. Altuhafi, C. O'sullivan, and I. Cavarretta, "Analysis of an image-based method to quantify the size and shape of sand particles," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 139, no. 8, pp. 1290–1307, 2013.

[11] S. Stock, "Recent advances in x-ray microtomography applied to materials," *International Materials Reviews*, vol. 53, no. 3, pp. 129–181, 2008.

[12] B. Zhao and J. Wang, "3D quantitative shape analysis on form, roundness, and compactness with μct," *Powder Technology*, vol. 291, pp. 262–275, 2016.

[13] E. J. Garboczi, "Three-dimensional mathematical analysis of particle shape using x-ray tomography and spherical harmonics: Application to aggregates used in concrete," *Cement and Concrete Research*, vol. 32, no. 10, pp. 1621–1638, 2002.

[14] M. Grigoriu, E. Garboczi, and C. Kafali, "Spherical harmonic-based random fields for aggregates used in concrete," *Powder Technology*, vol. 166, no. 3, pp. 123–138, 2006.

[15] X. Liu, E. Garboczi, M. Grigoriu, Y. Lu, and S. T. Erdoğan, "Spherical harmonic-based random fields based on real particle 3D data: improved numerical algorithm and quantitative comparison to real particles," *Powder Technology*, vol. 207, no. 1-3, pp. 78–86, 2011.

[16] J. W. Bullard and E. J. Garboczi, "Defining shape measures for 3D star-shaped particles: Sphericity, roundness, and dimensions," *Powder Technology*, vol. 249, pp. 241–252, 2013.

[17] B. Zhou and J. Wang, "Random generation of natural sand assembly using micro x-ray tomography and spherical harmonics," *Géotechnique Letters*, vol. 5, no. 1, pp. 6–11, 2015.

[18] J.-Y. Nie, D.-Q. Li, Z.-J. Cao, B. Zhou, and A.-J. Zhang, "Probabilistic characterization and simulation of realistic particle shape based on sphere harmonic representation and nataf transformation," *Powder Technology*, vol. 360, pp. 209–220, 2020.

[19] B. Zhou and J. Wang, "Generation of a realistic 3D sand assembly using x-ray micro-computed tomography and spherical harmonic-based principal component analysis," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 41, no. 1, pp. 93–109, 2017.

[20] E. J. Garboczi and J. W. Bullard, "3D analytical mathematical models of random star-shape particles via a combination of x-ray computed microtomography and spherical harmonic analysis," *Advanced Powder Technology*, vol. 28, no. 2, pp. 325–339, 2017.

[21] C. Brechbühler, G. Gerig, and O. Kübler, "Parametrization of closed surfaces for 3-D shape description," *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 154–170, 1995.

[22] L. Shen and F. Makedon, "Spherical mapping for processing of 3D closed surfaces," *Image and Vision Computing*, vol. 24, no. 7, pp. 743–761, 2006.

[23] L. Shen, H. Farid, and M. A. McPeek, "Modeling three-dimensional morphological structures using spherical harmonics," *Evolution: International Journal of Organic Evolution*, vol. 63, no. 4, pp. 1003–1016, 2009.

[24] B. Zhou, J. Wang, and B. Zhao, "Micromorphology characterization and reconstruction of sand particles using micro x-ray tomography and spherical harmonics," *Engineering Geology*, vol. 184, pp. 126–137, 2015.

[25] D. Su and W. Yan, "3D characterization of general-shape sand particles using microfocus x-ray computed tomography and spherical harmonic functions, and particle regeneration using multivariate random vector," *Powder Technology*, vol. 323, pp. 8–23, 2018.

[26] D. Wei, J. Wang, and B. Zhao, "A simple method for particle shape generation with spherical harmonics," *Powder Technology*, vol. 330, pp. 284–291, 2018.

[27] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*, ser. Wiley Series in Probability and Statistics. John Wiley & Sons, 2015.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2016, http://www.deeplearningbook.org.

[29] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[30] Z. Wang and D. W. Scott, "Nonparametric density estimation for high-dimensional data—algorithms and applications," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 11, no. 4, p. e1461, 2019.

[31] E. J. Garboczi and J. W. Bullard, "Contact function, uniform-thickness shell volume, and convexity measure for 3d star-shaped random particles," *Powder technology*, vol. 237, pp. 191–201, 2013.

[32] Z. Qian, E. Garboczi, G. Ye, and E. Schlangen, "Anm: a geometrical model for the composite structure of mortar and concrete using real-shape particles," *Materials and Structures*, vol. 49, no. 1-2, pp. 149–158, 2016.

[33] D. Wei, J. Wang, J. Nie, and B. Zhou, "Generation of realistic sand particles with fractal nature using an improved spherical harmonic analysis," *Computers and Geotechnics*, vol. 104, pp. 1–12, 2018.

[34] P. Vincent and Y. Bengio, "Manifold parzen windows," in *Advances in Neural Information Processing Systems*, 2003, pp. 849–856.

[35] J. Brehmer and K. Cranmer, "Flows for simultaneous manifold learning and density estimation," *arXiv preprint arXiv:2003.13913*, 2020.

[36] J. D. Hiller and H. Lipson, "STL 2.0: a proposal for a universal multi-material additive manufacturing file format," in *Proceedings of the Solid Freeform Fabrication Symposium*, vol. 3. Citeseer, 2009, pp. 266–278.

[37] I. Lackatos, *Proof and refutations*, ser. Cambridge Philosophy Classics. The logic of mathematical discovery, Cambridge: Cambridge University Press, 1976.

[38] K. Atkinson and W. Han, *Theoretical Numerical Analysis*, ser. Texts in Applied Mathematics. Springer, 2005, vol. 39.

[39] G. B. Folland, *Fourier analysis and its applications*. American Mathematical Soc., 2009, vol. 4.

[40] G. Ierley and B. Parker, "PDF notes from sio239: Math Methods for Geophysics," 2008. [Online]. Available: https://igppweb.ucsd.edu/~parker/SIO239/sh.pdf

[41] F. Dai and K. Wang, "Convergence rate of spherical harmonic expansions of smooth functions," *Journal of Mathematical Analysis and Applications*, vol. 348, no. 1, pp. 28–33, 2008.

[42] J. Flaherty, "Notes from Finite Element Analysis." [Online]. Available: http://www.cs.rpi.edu/~flaherje/FEM/fem6.ps

[43] C. Blakely, A. Gelb, and A. Navarra, "An automated method for recovering piecewise smooth functions on spheres free from gibbs oscillations." *Sampling Theory in Signal & Image Processing*, vol. 6, no. 3, 2007.

[44] R. Archibald, A. Gelb, and J. Yoon, "Polynomial fitting for edge detection in irregularly sampled signals and images," *SIAM journal on numerical analysis*, vol. 43, no. 1, pp. 259–279, 2005.

[45] A. Gelb, "The resolution of the gibbs phenomenon for spherical harmonics," *Mathematics of Computation*, vol. 66, no. 218, pp. 699–717, 1997.

[46] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[47] A. N. Pressley, *Elementary Differential Geometry*, ser. Springer Undergraduate Mathematics Series. Springer Science & Business Media, 2010.

[48] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*. Springer, 2003, pp. 35–57.

[49] J. A. Bærentzen, J. Gravesen, F. Anton, and H. Aanæs, *Guide to computational geometry processing: foundations, algorithms, and methods*. Springer Science & Business Media, 2012.

[50] M. Cenanovic, P. Hansbo, and M. G. Larson, "Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement," *arXiv preprint arXiv:1703.05745*, 2017.

[51] MATLAB, *version 9.5.0 (R2018b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[52] D. Poelaert, J. Schniewind, and F. Janssens, "Surface area and curvature of the general ellipsoid," *arXiv preprint arXiv:1104.5145*, 2011.

[53] S. Bektas, "Curvature of the ellipsoid with cartesian coordinates," *Landscape Architecture and Regional Planning*, vol. 2, no. 2, p. 61, 2017.

[54] A. Jaklič, A. Leonardis, and F. Solina, "Superquadrics and their geometric properties," in *Segmentation and recovery of superquadrics*. Springer, 2000, pp. 13–39.

[55] L. I. Smith, "A tutorial on Principal Components Analysis," Tech. Rep., 2002.

[56] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[57] G. Schwarz *et al.*, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[59] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, no. 6, 2004.

[60] S. Pethel and D. Hahs, "Exact test of independence using mutual information," *Entropy*, vol. 16, no. 5, 2014.

[61] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/stats.html#statistical-tests

[62] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[63] M. Jiang, H.-S. Yu, and D. Harris, "A novel discrete model for granular material incorporating rolling resistance," *Computers and Geotechnics*, vol. 32, no. 5, pp. 340–357, 2005.

[64] C. Simon, "Generating uniformly distributed numbers on a sphere," Feb 2015. [Online]. Available: http://corysimon.github.io/articles/uniformdistn-on-sphere/

[65] W. Jarosz, "Efficient Monte Carlo methods for light transport in scattering media," Ph.D. dissertation, UC San Diego, September 2008. [Online]. Available: https://cs.dartmouth.edu/~wjarosz/publications/dissertation/appendixB.pdf

[66] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International Conference on Database Theory*. Springer, 2001, pp. 420–434.

[67] H. Haber, "PDF notes from physics 116c: Mathematical Methods in Physics III," 2012. [Online]. Available: http://scipp.ucsc.edu/~haber/ph116C/SphericalHarmonics_12.pdf

APPENDIX A PROOFS

A.1 Reconstruction loss

Our goal is to express the reconstruction loss in geometrical space. As a reminder, the coefficients $\widehat{w}_i^{(j)}$ are the reconstructions of the relative perturbations $\widehat{c}_i^{(j)}$ (4.19). We start by writing the loss (4.20) without matrix notation.

$$
\begin{aligned}
R(\mathbf{P}, \boldsymbol{\mu}) &= \frac{1}{N_s} \sum_{j=1}^{N_s} \sum_{i=1}^{d-1} \left| \widehat{c}_i^{(j)} - \widehat{w}_i^{(j)} \right|^2 \\
&= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{|c_0^{(j)}|^2} \sum_{i=1}^{d-1} \left| c_0^{(j)} \widehat{c}_i^{(j)} - c_0^{(j)} \widehat{w}_i^{(j)} \right|^2 \\
&= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{|c_0^{(j)}|^2} \sum_{i=1}^{d-1} \left| c_i^{(j)} - w_i^{(j)} \right|^2 \qquad \left(\text{we define } w_i^{(j)} := c_0^{(j)} \widehat{w}_i^{(j)} \right) \\
&= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{|c_0^{(j)}|^2} \left(\left| c_0^{(j)} - c_0^{(j)} \right|^2 + \sum_{i=1}^{d-1} \left| c_i^{(j)} - w_i^{(j)} \right|^2 \right)
\end{aligned}
$$

In the last step, we simply added zero to the expression. Next, we use Parseval's equality to express the loss in the geometric space:

$$
\begin{aligned}
R(\mathbf{P}, \boldsymbol{\mu}) &= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{4\pi \langle r^{(j)} \rangle^2} \left(|c_0^{(j)} - c_0^{(j)}|^2 + \sum_{i=1}^{d-1} |c_i^{(j)} - w_i^{(j)}|^2 \right) \\
&= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{\langle r^{(j)} \rangle^2} \left\langle \Bigg(\underbrace{\sum_{i=0}^{d-1} c_i^{(j)} Y_i(\theta, \phi)}_{\text{Original Surface : } r^{(j)}(\theta,\phi)} - \underbrace{\left(c_0^{(j)} Y_0(\theta, \phi) + \sum_{i=1}^{d-1} w_i^{(j)} Y_i(\theta, \phi) \right)}_{\text{Reconstructed surface : } r_w^{(j)}(\theta,\phi)} \Bigg)^2 \right\rangle \\
&= \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{\left\langle \left(r^{(j)} - r_w^{(j)} \right)^2 \right\rangle}{\langle r^{(j)} \rangle^2},
\end{aligned}
$$

which concludes the proof.

A.2 Cumulative Variance vs Reconstruction Loss

We demonstrate that the cumulative variance (4.26) and the reconstruction loss (4.20) are related by an affine relation. The idea is to express the reconstruction loss in terms of the principal components $x_k^{(j)}$ that are ignored when fixing $m < d-1$.

$$\begin{aligned} R(\mathbf{Q}, \boldsymbol{\mu}) &= \frac{1}{N_s} \sum_{j=1}^{N_s} \sum_{i=m+1}^{d-1} |x_i^{(j)}|^2 \\ &= \sum_{i=m+1}^{d-1} \frac{1}{N_s} \sum_{j=1}^{N_s} |x_i^{(j)}|^2 \\ &= \sum_{i=m+1}^{d-1} \sigma_i^2 \\ &= \sum_{i=1}^{d-1} \sigma_i^2 - \sum_{i=1}^{m} \sigma_i^2 \\ &= \Big(1 - \mathrm{CV}(i)\Big) \sum_{i=1}^{d-1} \sigma_i^2. \end{aligned}$$

This relation illustrates how choosing a high cumulative variance relates to selecting a low reconstruction loss. Indeed, as CV approaches unity, the reconstruction goes to zero. The number of principal components is often heuristically chosen to yield a CV between 95% and 99%, which ensures that $R(\mathbf{Q}, \boldsymbol{\mu}) \leq 5\% \sum_{i=1}^{d-1} \sigma_i^2$.

APPENDIX B FORMULAS FOR SPHERICAL HARMONICS

This appendix provides useful formulas associated with the spherical harmonics. These formulas are assembled here for those who wish to reproduce this research. The formula from Section B are taken from [25], while the formulas in Sections B and B are taken from [13].

B.1 Derivatives

We show a list of first-order derivatives of the spherical harmonics.

$$\frac{\partial Y_\ell^m(\theta,\phi)}{\partial\theta} = \begin{cases} -m\sqrt{2}C(\ell,m)\,P_\ell^{|m|}(\cos\phi)\,\sin(m\theta) & m>0, \\ 0 & m=0, \\ -m\sqrt{2}C(\ell,m)\,P_\ell^{|m|}(\cos\phi)\,\cos(-m\theta) & m<0, \end{cases} \tag{B.1}$$

$$\frac{\partial Y_\ell^m(\theta,\phi)}{\partial\phi} = \begin{cases} \sqrt{2}C(\ell,m)\,\dfrac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi}\,\cos(m\theta) & m>0, \\ C(\ell,0)\dfrac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi} & m=0, \\ \sqrt{2}C(\ell,m)\,\dfrac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi}\,\sin(-m\theta) & m<0, \end{cases} \tag{B.2}$$

where

$$\frac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi} = \frac{-1}{\sin\phi}\left((\ell+1)\cos\phi P_\ell^{|m|}(\cos\phi) - (\ell-|m|+1)P_{\ell+1}^{|m|}(\cos\phi)\right). \tag{B.3}$$

The second-order derivatives of the spherical harmonics are given by:

$$\frac{\partial^2 Y_\ell^m(\theta,\phi)}{\partial\theta^2} = \begin{cases} -m^2\sqrt{2}C(\ell,m)\,P_\ell^{|m|}(\cos\phi)\,\cos(m\theta) & m>0, \\ 0 & m=0, \\ -m^2\sqrt{2}C(\ell,m)\,P_\ell^{|m|}(\cos\phi)\,\sin(-m\theta) & m<0, \end{cases} \tag{B.4}$$

$$\frac{\partial^2 Y_\ell^m(\theta,\phi)}{\partial\phi\partial\theta} = \begin{cases} -m\sqrt{2}C(\ell,m)\,\dfrac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi}\,\sin(m\theta) & m>0, \\ 0 & m=0, \\ -m\sqrt{2}C(\ell,m)\,\dfrac{\partial P_\ell^{|m|}(\cos\phi)}{\partial\phi}\,\cos(-m\theta) & m<0, \end{cases} \tag{B.5}$$

$$\frac{\partial^2 Y_\ell^m(\theta,\phi)}{\partial\phi^2} = \begin{cases} \sqrt{2}C(\ell,m)\,\dfrac{\partial^2 P_\ell^{|m|}(\cos\phi)}{\partial\phi^2}\,\cos(m\theta) & m>0, \\ C(\ell,0)\,\dfrac{\partial^2 P_\ell^{|m|}(\cos\phi)}{\partial\phi^2} & m=0, \\ \sqrt{2}C(\ell,m)\,\dfrac{\partial^2 P_\ell^{|m|}(\cos\phi)}{\partial\phi^2}\,\sin(-m\theta) & m<0, \end{cases} \tag{B.6}$$

where

$$\begin{aligned} \frac{\partial^2 P_\ell^{|m|}(\cos\phi)}{\partial\phi^2} = \frac{1}{\sin^2\phi}\Big(\; & \left(\ell+1+(\ell+1)^2\cos^2\phi\right)P_\ell^{|m|}(\cos\phi) \\ & - 2\cos\phi(\ell-|m|+1)(\ell+2)P_{\ell+1}^{|m|}(\cos\phi) \\ & + (\ell-|m|+1)(\ell-|m|+2)P_{\ell+2}^{|m|}(\cos\phi)\;\Big). \end{aligned} \tag{B.7}$$

B.2 Inertia Tensor Components

The following formulas allow one to compute the inertia tensor of a shape described by the surface function $r(\theta,\phi)$. These formulas can be derived by expressing (3.7) in spherical coordinates.

$$
\begin{aligned}
I_{11} &= \frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin\phi\,(1-\sin^2\phi\cos^2\theta)d\phi d\theta \\
I_{22} &= \frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin\phi\,(1-\sin^2\phi\sin^2\theta)d\phi d\theta \\
I_{33} &= \frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin^3\phi\,d\phi d\theta \\
I_{12} &= -\frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin^3\phi\cos\theta\sin\theta\,d\phi d\theta \\
I_{13} &= -\frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin^2\phi\cos\phi\cos\theta\,d\phi d\theta \\
I_{23} &= -\frac{1}{5}\int_0^{2\pi}\int_0^{\pi} r^5(\theta,\phi)\sin^2\phi\cos\phi\sin\theta\,d\phi d\theta.
\end{aligned}
\tag{B.8}
$$

B.3 Curvature of Radial Functions

This section explains how to compute the curvature of any smooth surface $r(\theta,\phi)$ assuming one can compute its first and second order derivatives. The formulas were verified and adapted from [13], but we remark that a different convention for the angles θ and ϕ was used and a few typos were present. In the formulas, all vectors shall be expressed in the classical orthonormal basis. First, the position vector of the surface is

$$
\boldsymbol{r}(\theta,\phi) = \Big(r(\theta,\phi)\,\cos\theta\sin\phi, r(\theta,\phi)\,\sin\theta\sin\phi, r(\theta,\phi)\,\cos\phi\Big). \tag{B.9}
$$

The first step is to compute the tangent vectors on the surface

$$
\boldsymbol{r}_\theta = \Big(r_\theta\,\cos\theta\sin\phi - r\sin\phi\sin\theta, r_\theta\,\sin\theta\sin\phi + r\cos\theta\sin\phi, r_\theta\,\cos\phi\Big), \tag{B.10}
$$

$$
\boldsymbol{r}_\phi = \Big(r_\phi\,\cos\theta\sin\phi + r\cos\phi\cos\theta, r_\phi\,\sin\theta\sin\phi + r\sin\theta\cos\phi, r_\phi\,\cos\phi - r\sin\phi\Big). \tag{B.11}
$$

The second step is to compute the normal vector

$$
\boldsymbol{n} = \frac{\boldsymbol{r}_\phi \times \boldsymbol{r}_\theta}{||\boldsymbol{r}_\phi \times \boldsymbol{r}_\theta||}, \tag{B.12}
$$

whose components are

$$
\begin{aligned}
n_1 &= S^{-1}(rr_\theta\sin\theta - rr_\phi\sin\phi\cos\phi\cos\theta + r^2\sin^2\phi\cos\theta), \\
n_2 &= S^{-1}(-rr_\theta\cos\theta - rr_\phi\sin\phi\cos\phi\sin\theta + r^2\sin^2\phi\sin\theta), \\
n_3 &= S^{-1}(rr_\phi\sin^2\phi + r^2\cos\phi\sin\phi),
\end{aligned}
\tag{B.13}
$$

where $S = r\sqrt{r_\theta^2 + r_\phi^2 \sin^2\phi + r^2 \sin^2\phi}$ is the surface Jacobian. We compute the partial derivatives of the normal vector as

$$\frac{\partial n_i}{\partial \theta} = S^{-1}\left(a_i - b_i\Big(\frac{r_\theta}{r} + \frac{c}{S^2}\Big)\right), \tag{B.14}$$

$$\frac{\partial n_i}{\partial \phi} = S^{-1}\left(d_i - b_i\Big(\frac{r_\phi}{r} + \frac{e}{S^2}\Big)\right), \tag{B.15}$$

where

$$\begin{aligned}
a_1 &= r_\theta^2 \sin\theta + r r_{\theta\theta} \sin\theta + r r_\theta \cos\theta - r_\theta r_\phi \sin\phi\cos\phi\cos\theta - r r_{\theta\phi} \sin\phi\cos\phi\cos\theta \\
&\quad + r r_\phi \sin\phi\cos\phi\sin\theta + 2 r r_\theta \sin^2\phi\cos\theta - r^2\sin^2\phi\sin\theta, \\
a_2 &= -r_\theta^2\cos\theta - r r_{\theta\theta}\cos\theta + r r_\theta \sin\theta - r_\phi r_\theta \sin\phi\cos\phi\sin\theta \\
&\quad - r r_{\theta\phi}\sin\phi\cos\phi\sin\theta - r r_\phi \sin\phi\cos\phi\cos\theta + 2 r r_\theta \sin^2\phi\sin\theta + r^2\sin^2\phi\cos\theta, \\
a_3 &= r_\phi r_\theta \sin^2\phi + r r_{\theta\phi}\sin^2\phi + 2 r r_\theta \sin\phi\cos\phi, \\
\\
b_1 &= r r_\theta \sin\theta - r r_\phi \sin\phi\cos\phi\cos\theta + r^2\sin^2\phi\cos\theta, \\
b_2 &= -r r_\theta\cos\theta - r r_\phi \sin\phi\cos\phi\sin\theta + r^2\sin^2\phi\sin\theta, \\
b_3 &= r r_\phi \sin^2\phi + r^2\sin\phi\cos\phi, \\
\\
c &= r^2(r_\theta r_{\theta\theta} + r_\phi r_{\theta\phi}\sin^2\phi + r r_\theta \sin^2\phi), \\
\\
d_1 &= r_\theta r_\phi \sin\theta + r r_{\theta\phi}\sin\theta - r_\phi^2 \sin\phi\cos\phi\cos\theta - r r_{\phi\phi}\sin\phi\cos\phi\cos\theta \\
&\quad - r r_\phi \cos^2\phi\cos\theta + r r_\phi \sin^2\phi\cos\theta + 2 r r_\phi \sin^2\phi\cos\theta + 2r^2\sin\phi\cos\phi\cos\theta, \\
d_2 &= -r_\theta r_\phi\cos\theta - r r_{\theta\phi}\cos\theta - r_\phi^2\sin\phi\cos\phi\sin\theta - r r_{\phi\phi}\sin\phi\cos\phi\sin\theta \\
&\quad - r r_\phi\cos^2\phi\sin\theta + r r_\phi\sin^2\phi\sin\theta + 2 r r_\phi\sin^2\phi\sin\theta + 2r^2\sin\phi\cos\phi\sin\theta, \\
d_3 &= r_\phi^2\sin^2\phi + r r_{\phi\phi}\sin^2\phi + 4 r r_\phi \sin\phi\cos\phi + r^2\cos^2\phi - r^2\sin^2\phi, \\
\\
e &= r^2(r_\theta r_{\theta\phi} + r_\phi r_{\phi\phi}\sin^2\phi + r r_\phi \sin^2\phi + r_\phi^2\sin\phi\cos\phi + r^2\sin\phi\cos\phi).
\end{aligned} \tag{B.16}$$

To compute the curvature on the surface, we must calculate the first and second fundamental forms of the surface [47]. The components of the first fundamental form can be computed

via

$$\begin{aligned} E &= \langle \boldsymbol{r}_\phi, \boldsymbol{r}_\phi \rangle = r_\phi^2 + r^2, \\ F &= \langle \boldsymbol{r}_\theta, \boldsymbol{r}_\phi \rangle = r_\theta r_\phi, \\ G &= \langle \boldsymbol{r}_\theta, \boldsymbol{r}_\theta \rangle = r_\theta^2 + r^2 \sin^2 \phi. \end{aligned} \tag{B.17}$$

The components of the second fundamental form can be computed with

$$\begin{aligned} L &= -\langle \boldsymbol{r}_\phi, \boldsymbol{n}_\phi \rangle, \\ M &= -\langle \boldsymbol{r}_\theta, \boldsymbol{n}_\phi \rangle = -\langle \boldsymbol{r}_\phi, \boldsymbol{n}_\theta \rangle, \\ N &= -\langle \boldsymbol{r}_\theta, \boldsymbol{n}_\theta \rangle. \end{aligned} \tag{B.18}$$

The Weingarten matrix is therefore given as

$$\mathbf{W} = \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} L & M \\ M & N \end{bmatrix}. \tag{B.19}$$

The eigenvalues of $\mathbf{W}$ are the principal curvatures κ_1, κ_2. Its determinant and trace provide the Gaussian and mean curvatures:

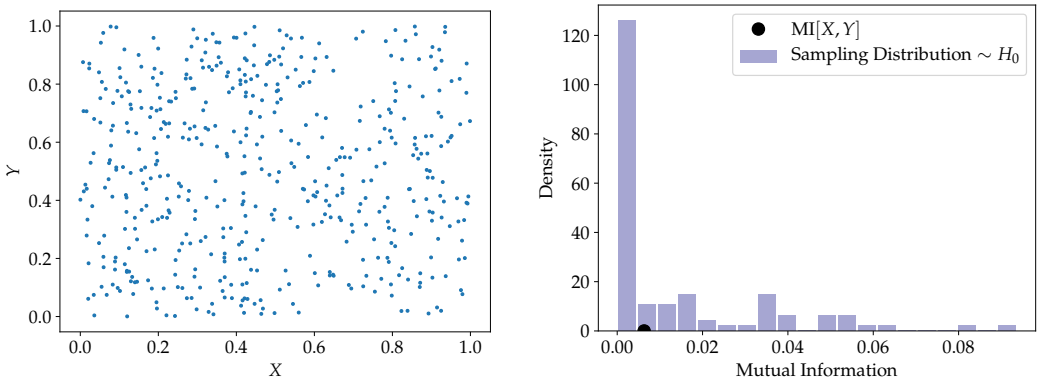
$$H = \frac{LG + NE - 2MF}{2(EG - F^2)}, \qquad K = \frac{LN - M^2}{EG - F^2}. \tag{B.20}$$

APPENDIX C CORRELATION AND INDEPENDENCE TESTS

In this appendix, we shall verify the Spearman and the Mutual Information (MI) tests on simple bivariate distributions. The significance level $\alpha = 5\%$ was used for every test and each dataset was generated with 500 samples.

C.1 Uniform distributions

Let $X, Y \sim U(0, 1)$ be sampled independently, see Figure C.1.



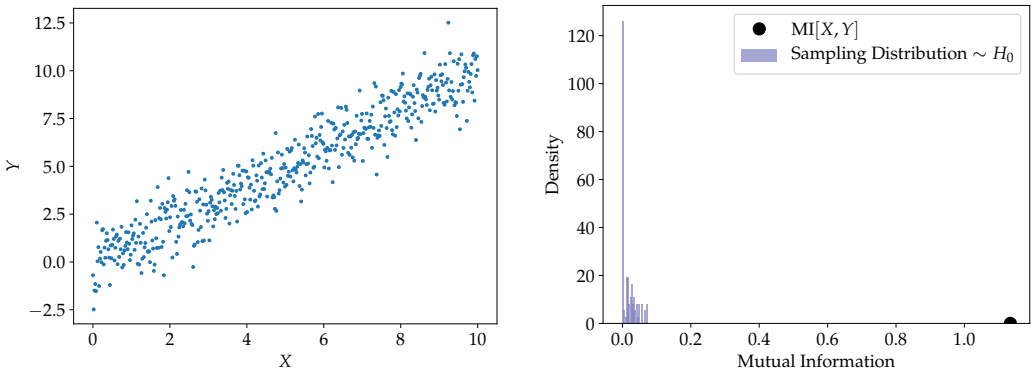
(a) Joint distribution of X and Y (b) Sampling distribution for MI

Figure C.1 Independence test for a Uniform distribution.

The Spearman correlation is -0.03 with a p-value of 0.5, which means that the data is uncorrelated. The mutual information is 0.006 with a p-value of 0.4, which implies that the data is independent.

C.2 Linear Relationship

Let $X \sim U(0, 10)$ and $Y = X + \epsilon$ with $\epsilon \sim N(0, 1)$, see Figure C.2.



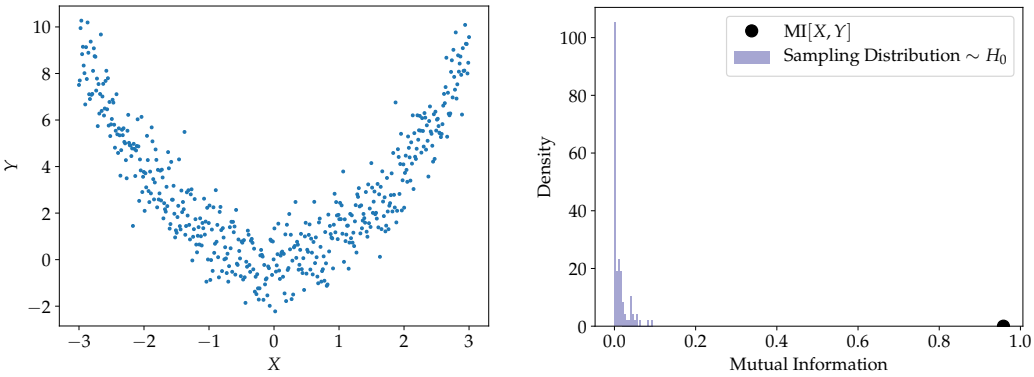
(a) Joint distribution of X and Y (b) Sampling distribution for MI

Figure C.2 Independence test for linear relationship.

We compute a Spearman correlation of 0.95 with a p-value of 1.17×10^{-263}, which shows very strong correlation. The mutual information is 1.134 with a p-value of zero. A look at the Figure C.2 (b) demonstrates that the Mutual Information score is located far away from the sampling distribution so we can safely state that X and Y are independent.

C.3 Quadratic Relationship

Let $X \sim U(0, 10)$ and $Y = X^2 + \epsilon$ where $\epsilon \sim N(0, 1)$, see Figure C.3.



(a) Joint distribution of X and Y (b) Sampling distribution for MI

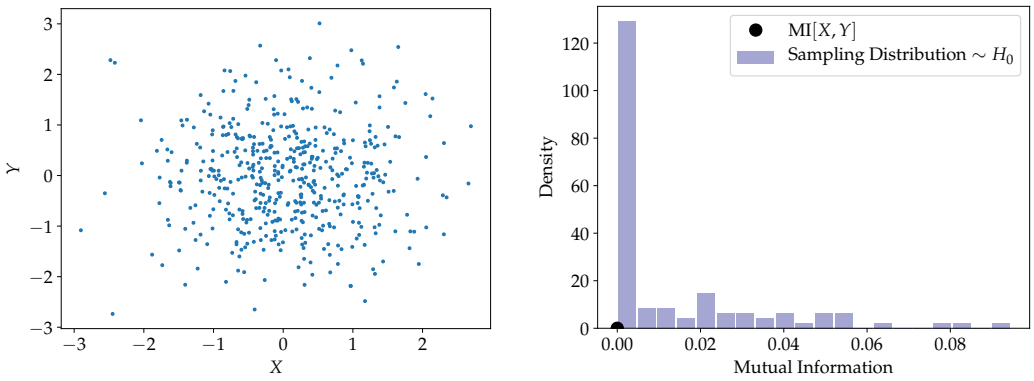
Figure C.3 Independence test for quadratic relationship.

We get a correlation of -0.01 with a p-value of 0.958, which suggests that the data is non-

correlated. The Mutual Information score is 0.958 with a p-value of zero.

C.4 Independent Gaussians

Let $X, Y \sim N(0, 1)$ be independent variables, see Figure C.4.



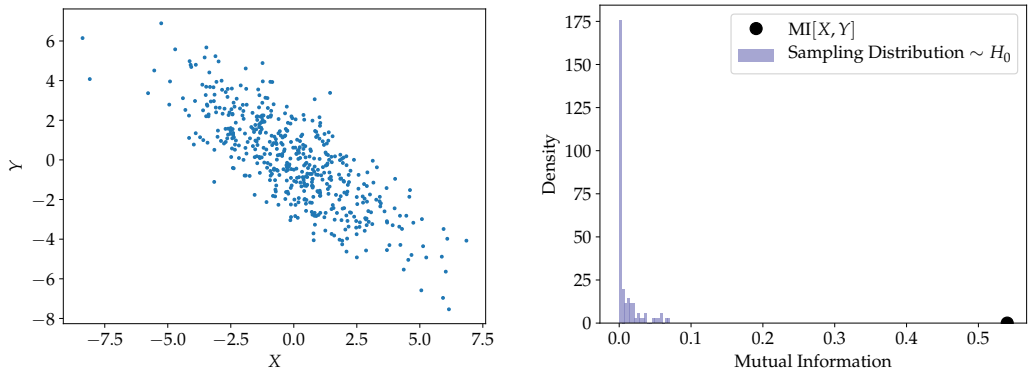
(a) Joint distribution of X and Y (b) Sampling distribution for MI

Figure C.4 Independence test for independent Gaussians.

The Spearman correlation is -0.019 with a p-value of 0.439, which indicates that the data is not correlated. The mutual information is 0 with a p-value of 0.439. The two test jointly show evidence that the data is independent.

C.5 Correlated Gaussians

Let $(X, Y) \sim N(\mathbf{0}, \Sigma)$ be correlated variables, see Figure C.5.



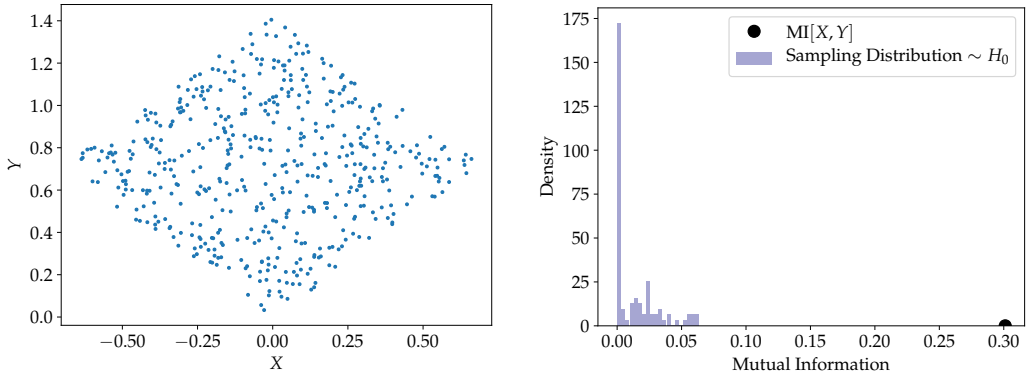
(a) Joint distribution of X and Y (b) Sampling distribution for MI

Figure C.5 Independence test for correlated Gaussians.

In that case, we measure a correlation of -0.77 with a p-value of 1.23×10^{-99}, which indicates that the data is negatively correlated. The Mutual Information score is 0.54 with a p-value of 0, which states that the data independent.

C.6 Square-Shaped Distribution

We can sample X, Y from a diamond centered at the origin, see Figure C.6. These variables are dependent by construction.



(a) Joint distribution of X and Y

(b) Sampling distribution for MI

Figure C.6 Independence test for square-shaped distribution.

The observed Spearman correlation is -0.004 with a p-value of 0.925, which suggests that the data is un-correlated. The mutual information is 0.301 with a p-value of zero, which confirms dependence.

APPENDIX D NORMS IN $\ell^2(\mathbb{N})$

Two decades ago, multiple studies demonstrated that, for high dimensional data, the concepts of proximity and nearest neighbor may become ambiguous. For example, the authors of [66] argued that the meaningfulness of said concepts becomes highly dependent on the choice of the norm. They provided theoretical and empirical evidence that using fractional norms compared with Manhattan or Euclidean norms can improve the performance of clustering and K-Nearest Neighbors algorithms for high dimensional tasks. In this appendix, we examine these notions by comparing how the Euclidean and Mahalanobis norms behave when restricted to the Hilbert space $\ell^2(\mathbb{N})$,

$$\ell^2(\mathbb{N}) = \Big\{ \boldsymbol{c} \in \mathbb{R}^\infty \,\Big|\, \sum_{i=1}^{\infty} |c_i|^2 < \infty \Big\}, \tag{D.1}$$

the space in which the SH coefficients belong.

D.1 Ill-definition of the Mahalanobis norm

In this section, we demonstrate that the Mahalanobis norm is not always defined in the space $\ell^2(\mathbb{N})$. By construction, the infinite Euclidean norm

$$\|\boldsymbol{c}\|_{\ell^2}^2 := \sum_{i=1}^{\infty} |c_i|^2, \tag{D.2}$$

is always finite on the SH coefficients. The Mahalanobis norm unfortunately does not share this property because it can be shown to be infinite for some sequences in $\ell^2(\mathbb{N})$. As a reminder, the Mahalanobis norm is defined as

$$\|\boldsymbol{c}\|_{\boldsymbol{\Sigma}}^2 = \boldsymbol{c}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{c}, \tag{D.3}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the data. For example, let us sample the SH coefficients from an infinity of independent uniform distributions, $c_i \sim U([0, \frac{1}{i}])$, $i = 1, 2, \dots$. All

sequences sampled from this distribution are in $\ell^2(\mathbb{N})$ since

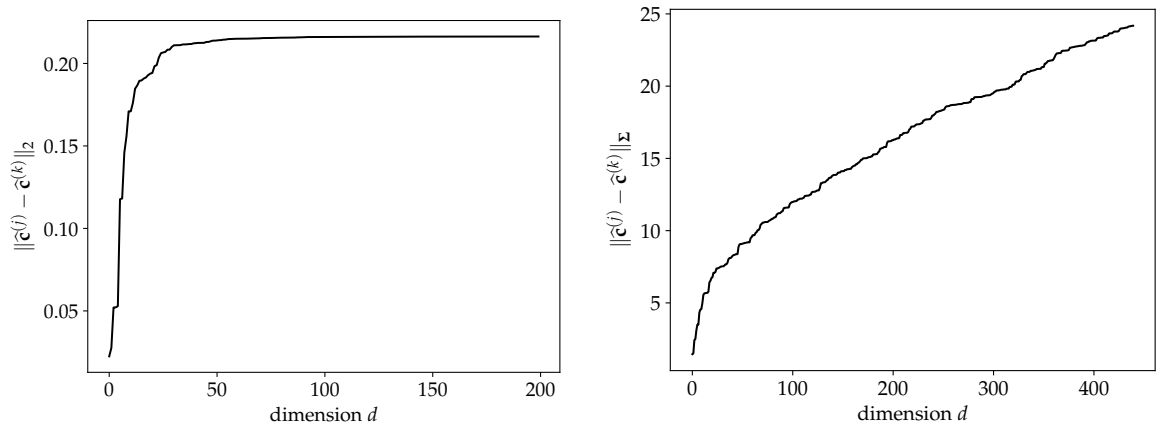
$$\begin{aligned} \|\boldsymbol{c}\|_{\ell^2}^2 &= \sum_{i=1}^{\infty} |c_i|^2 \\ &\leq \sum_{i=1}^{\infty} \frac{1}{i^2} \\ &= \frac{\pi^2}{6} < \infty. \end{aligned} \tag{D.4}$$

The covariance matrix of this distribution is diagonal with diagonal elements $\boldsymbol{\Sigma}(i,i) = \dfrac{1}{12i^2}$. Note that the variances diminish as one goes further in the sequence. Though this example is contrived, the reduction of variance always occurs when studying datasets of real particles. Let $\boldsymbol{c} = (1, 1/2, 1/3, \cdots, 1/i, \cdots)$, its Mahalanobis norm is infinite,

$$\begin{aligned} \|\boldsymbol{c}\|_{\boldsymbol{\Sigma}}^2 &= \sum_{i=1}^{\infty} \frac{|c_j|^2}{\boldsymbol{\Sigma}(i,i)} \\ &= \sum_{i=1}^{\infty} \frac{1}{i^2 \boldsymbol{\Sigma}(i,i)} \\ &= \sum_{i=1}^{\infty} 12 \frac{i^2}{i^2} \\ &= \sum_{i=1}^{\infty} 12 = \infty \end{aligned} \tag{D.5}$$

This shows that the Mahalanobis norm can potentially be undefined when working with an infinity of SH coefficients. In this example, the Mahalanobis norm behaves badly because of the division by $\boldsymbol{\Sigma}(i,i)$ which cancels out the property that the amplitude of the coefficients tends rapidly to zero.

The previous analysis considers infinity of SH coefficients which is not possible in practice. However, similar behavior of the Euclidean and Mahalanobis norms can be observed when using a steadily increasing number of SH coefficients. In following experiment, two random river particles are selected and the distance between their relative perturbations $\widehat{c}_i, i \neq 0$ is computed using only d dimensions. Note that the relative perturbations also live in $\ell^2(\mathbb{N})$ since they are simply a scaled version of the SH coefficients. Figure D.1 illustrates the results for various values of d using the Euclidean and Mahalanobis norms. Figure D.1(a), shows that after $d = 50$ dimensions, the Euclidean norm reaches a plateau, which is expected since



(a) Euclidean distance

(b) Mahalanobis distance

Figure D.1 Distance computed between two arbitrary river particles for various.

$$\lim_{d\to\infty} \sum_{i=1}^{d} |c_i^{(j)} - c_i^{(k)}|^2 = \|\boldsymbol{c}^{(j)} - \boldsymbol{c}^{(k)}\|_{\ell^2}^2 < \infty \tag{D.6}$$

In Figure D.1(b), we observe that the Mahalanobis distance between the data points keeps increasing with respect to the dimension d.

D.2 Proximity and Nearest Neighbors

Typically, as the number of dimensions increases, the data points become *sparsely distributed* in feature space, making it harder to identify meaningful notions of proximity. Proximity can be investigated by studying computations of the nearest neighbors of an arbitrary query point $\widehat{\boldsymbol{c}}$. One common measure used to characterize the meaningfulness of proximity is the relative contrast [66], which is computed using the minimum and maximum distances between the query point $\widehat{\boldsymbol{c}}$ and all data points $\widehat{\boldsymbol{c}}^{(j)}$, $j = 1, 2, \ldots N_s$, that is

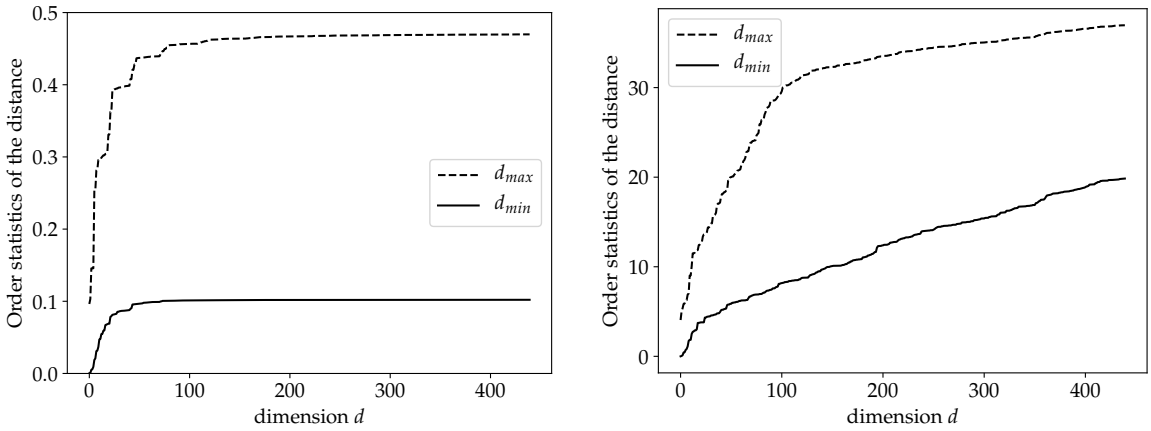
$$d_{\min} := \min_{1\le j\le N_s} \|\widehat{\boldsymbol{c}} - \widehat{\boldsymbol{c}}^{(j)}\| \tag{D.7}$$

$$d_{\max} := \max_{1\le j\le N_s} \|\widehat{\boldsymbol{c}} - \widehat{\boldsymbol{c}}^{(j)}\|. \tag{D.8}$$

The relative contrast is then defined as

$$\mathrm{RC}(\widehat{\boldsymbol{c}}) = \frac{d_{\max} - d_{\min}}{d_{\min}} = \frac{1 - {}^{d_{\max}}/_{d_{\min}}}{{}^{d_{\max}}/_{d_{\min}}}, \tag{D.9}$$

which is a strictly decreasing function of the ratio ${}^{d_{\max}}/_{d_{\min}}$. The nearest neighbors search of



(a) Euclidean norm

(b) Mahalanobis norm

Figure D.2 Minimal and Maximal distances from a query point as a function of dimensionality.

a query point $\widehat{\boldsymbol{c}}$ is said to be meaningful if the relative contrast $\mathrm{RC}(\widehat{\boldsymbol{c}})$ is large, meaning that the nearest particle appears much closer than the one that appears farthest. Unmeaningful nearest neighbor search occurs when the contrast is poor, which happens when the nearest and farthest neighbors are almost at the same distance from $\widehat{\boldsymbol{c}}$.

The following experiment is conducted to study the relative contrasts using both Euclidean and Mahalanobis norms: a random river particle is chosen as query points and the minimum and maximal distances $d_{\max}$ and $d_{\min}$ are calculated for various dimensions, see Figure D.2. By studying Figure D.2(a), we see that the distance to the nearest neighbor $d_{\min}$ converges quickly to the small value 0.1. Similar behavior has been observed for other river particles, which suggests that the river particles are very close to their nearest neighbors with respect to the Euclidean norm. Comparing the minimal distance to $d_{\max}$, the contrast of this specific query point converges approximately to 3.5. Results from Figure D.2(b) are quite different. First of all, the minimal distance $d_{\min}$ keeps increasing as d increases. This means that neighboring particles of $\widehat{\boldsymbol{c}}$ are *pulled away* from the query point as dimensionality grows. Note that we found that this behavior also occurs for other query points. When considering the maximal dimension, the observed contrast is around 0.85 which is smaller than the one obtained with the Euclidean norm. This is an indicator that the notion of nearest neighbor is better defined with the Euclidean norm than with the Mahalanobis norm.