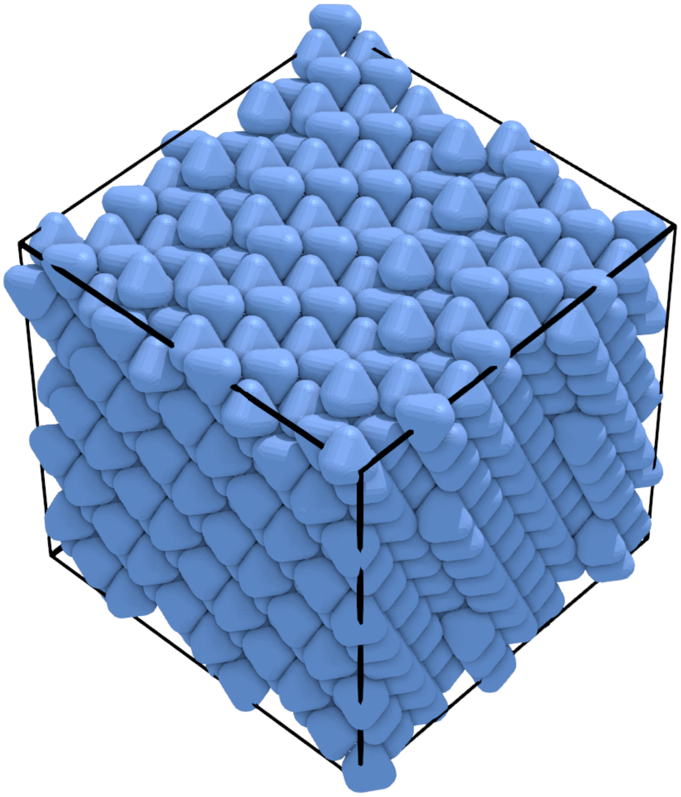
Universiteit Utrecht

Department of Physics

Crystal phase identification of "odd shaped" particles with increasing roundness using machine learning

Bachelor Thesis

Cyril Delaporte



Supervisors:
Prof. dr. ir. Marjolein Dijkstra
Robin van Damme
Gabriele Coli
Debye Institute for Nanomaterials Science

June 12, 2019

Abstract

Crystalline structure identification is a widely studied topic in condensed matter physics. Understanding how the structure of materials are determined by the shape of the particles it is build from is important for the understanding of the properties of these materials and can potentially help us tweak these properties to develop useful applications. In this research, we investigate how we can use machine learning tools for the identification of crystal structures by looking at systems of rounded polyhera. The bond order parameters of these systems are obtained from Floppy Box Monte Carlo simulations that produce large, high dimensional data sets. This data is analyzed with use of unsupervised machine learning, specifically dimensionality reduction techniques PCA and diffusion maps. We find that these techniques can identify different crystal structures for simple systems. For more complicated ones, a clear distinction of all crystal structures is hard to make with the naked eye. Additional techniques, like clustering algorithms, are needed to make a complete identification in more complex systems. Although the differences in results are minimal, diffusion maps reduces dimensionality slighlty better than PCA.

Contents

1 Introduction

The organization of individual building blocks into ordered structures is found everywhere in nature at all length scales. Crystal structures are found in atomic systems as well as in molecular materials, nanoparticles and colloids. Understanding the relationship between the macroscopic structure of a material and the properties of its particles is a widely studied topic in condensed matter physics with useful applications. For instance, nanoscale colloidal crystals have been shown to assemble into ordered superlattices with interesting mechanical properties, while micron-sized particles can be used to build materials that interact with visible light [1].

A particularly interesting aspect in this field are so called “packing problems” where self-assembly of a system is determined by the shape of the particles [2]. Packing problems of polyhedral shapes and the crystalline structures they can form as well as the recent advances in the synthesis of faceted nanoparticles and colloids have attracted interest the phase behaviour of such “odd-shaped” particle systems [1, 3, 4]. Discoveries in this field have shown to be useful and may play a role in tomorrows materials [5]. Furthermore, the rounding of a particle, due to suface charges or stabilization by a polymeric surface coating also plays a role in the packing of a system [6].

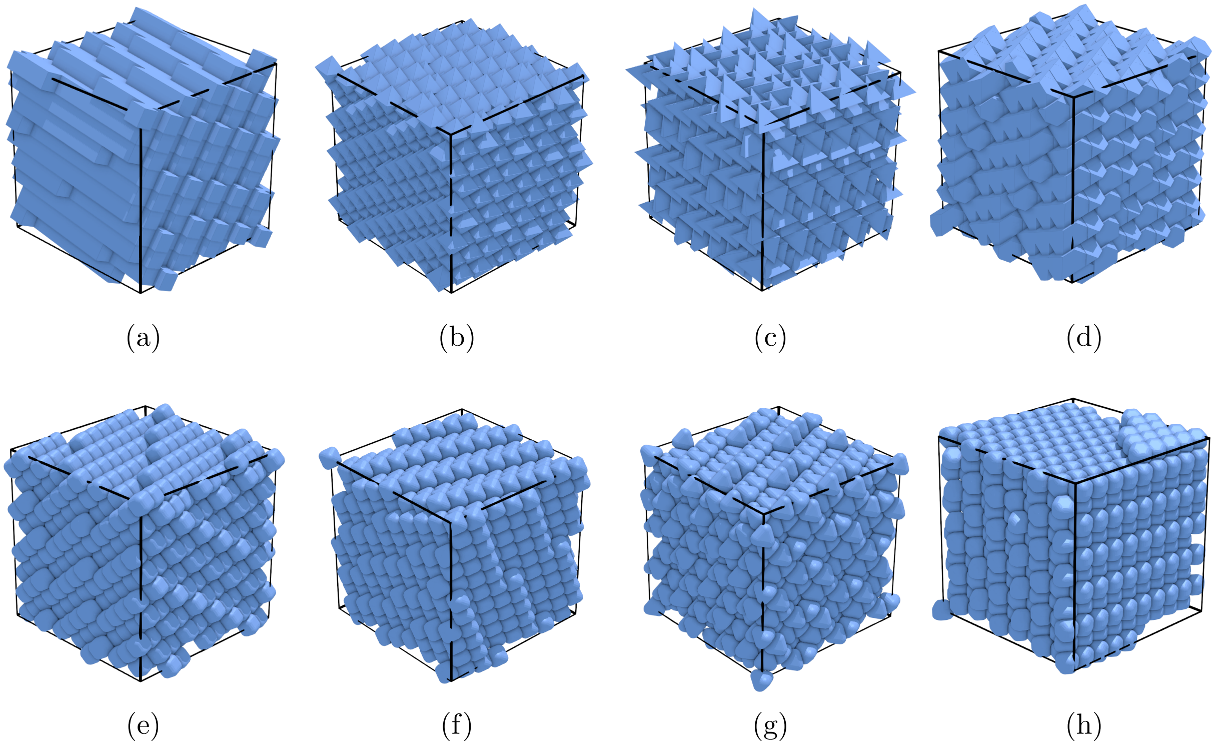


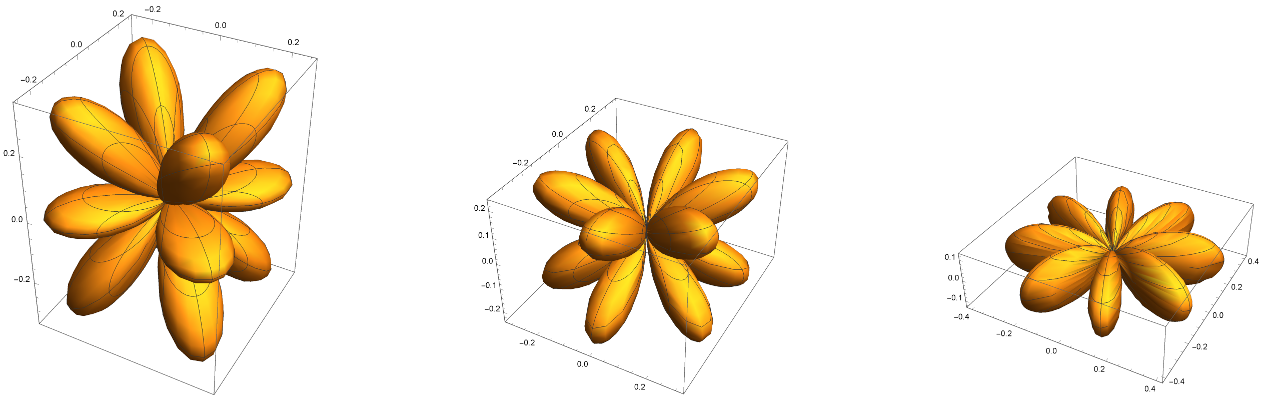
Figure 1: Some snapshots of the simulations used in this research. Figures 1a, 1b, 1c and 1d are close packed systems of cubes, octahedra, tetrahedra and truncated tetrahedra respectively. Figures 1e, 1f, 1g and 1h show the same systems of cubes, octahedra, tetrahedra and truncated tetrahedra, this time the individual particles are rounded.

In the process of studying these systems, scientists produce enormous amounts of data from the computationally expensive process of performing many numerical particle simulations which moreover have to be analysed. For crystal structure identification, tools called bond order parameters are used as a measure to quantify the types of crystal structures that occur in the studied systems. More on bond order parameters will be elaborated in the theory section (section 2). Still, this analysis is difficult and labor-intensive, partly due to the wide variety of crystal structures that can be found in self assembling systems [7].
As computer hardware and software continue to develop at high rates, high-performance methods are needed to study the generated data. Techniques and tools from data science and machine learning prove themselves remarkably useful in identifying and extracting trends and patterns within large data sets. One can, for instance, "teach" artificial neural-networks to identify crystal structures in a system if the types of structures are known beforehand [8]. However, typically it is not known which structures are present in a data set before analysis. For these type of problems, unsupervised machine learning techniques are used. One of these techniques that will be used in this thesis are dimensionality reduction techniques, specifically Principal Component Analysis (PCA) and Diffusion Maps. With dimensionality reduction techniques high dimensional output data is reduced to a point where the relevant information hidden in the data can be extracted. This sounds promising for finding structures in the high dimensional parameter space from particle simulations.

In this thesis crystal structures that are found in systems of rounded polyhedra are studied where the roundedness is characterized by a parameter $r \in [0,1]$ ($r = 0$ being an ideal polyhedron to $r = 1$ a sphere). This is done by running particle simulations from which the bond order parameters are obtained. This leaves us with large, high dimensional data sets, which is a typical problem where machine learning proves itself to be exceptionally useful. Here, unsupervised learning techniques are used. Specifically, the bond order parameters from these simulations using dimensionality reduction techniques PCA and Diffusion Maps. The results from this analysis are will be discussed and evaluated. In this research we investigate if PCA and diffusion maps are suitable techniques to identify crystal structures and discuss the differences and results for the two machine learning techniques.

2 Theory

This thesis can be roughly separated into two parts. The first part consists of the particle simulations that produces the required data. In the second part, the data sets are analysed by means of the machine learning methods principal component analysis and diffusion maps. In this section, the necessary technicalities for the simulations and analysis are elaborated. First, the derivation of bond order parameters used in this research are discussed and how they characterize the crystal structures. Next the machine learning techniques that are used in this research are described. Details on the simulation itself and how the output data is analyzed is discussed in the methods section (section 3).



(a) $Y_{l=4}^{m=2}(\theta,\phi)$ (b) $Y_{l=4}^{m=3}(\theta,\phi)$ (c) $Y_{l=4}^{m=4}(\theta,\phi)$

Figure 2: Real parts of the spherical harmonics plotted for $l = 4$ and $m = 2, 3, 4$ as an illustration

2.1 Bond Order parameters

Crystals are ordered particles in periodic lattices. Depending on the type of crystal, the neighbourhood of every particle has a certain symmetry. Bond order parameters can characterize these structures by making use of these symmetries. This method was first introduced in 1982 by Steinhardt et al. and have shown to be useful in studying bond orientational order in liquids and glasses [9]. The bond order parameters are based on the properties of spherical harmonics $Y_l^m(\theta, \phi)$. In figure 2 a few spherical harmonics are plotted to give an idea how these functions can be useful to characterize local structure.

The bond order parameters are constructed as follows: for each particle i a set of bonds to its closest neighbours is defined $\vec{\mathbf{r}}_{ij}$ (with j the closest neighbours of particle i). The bonds in the system are expanded in terms of spherical harmonics $Y_l^m(\vec{\mathbf{r}}_{ij})$ which are then averaged over the neighbourhood of particle i. The choice of this neighbourhood definition is not unique and specific methods have proven to be more useful in identifying different types of crystal structures [10].

Here a definition constructed by Voronoi-cell weighting is maintained. The advantage of this definition is that the geometry of the particle neighbourhoods is symmetric and parameter free, in contrast to other definitions. Voronoi-cell weighting works as follows. First a Voronoi diagram is constructed. An example for a 2D Voronoi cell is described here. Consider a plane with points (particles) at random positions. For each particle a Voronoi cell is constructed composed of line segments and vertices. The line segments are points that are equidistant to two nearest particles and vertices are points equidistant to at least three particles (see figure 3). The contribution of each neighbour is weighted by an associated area factor $A(f)/A$, where $A(f)$ is the length (surface area in 3D) of line segment f separating two neighbouring particles.

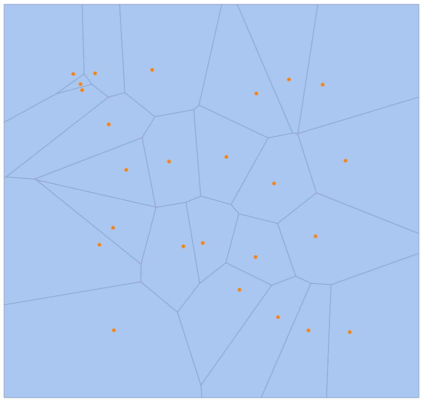


Figure 3: Illustration of Voronoi diagram in 2D for a set of random particles denoted by the red dots.

Hence, $A = \sum_{f\in\mathcal{F}(a)} A(f)$ is the total area of the Voronoi cell of a particle. The so called Minkowski metric is defined as

$$q_{lm}(i) = \sum_{f\in\mathcal{F}(a)} \frac{A(f)}{A} Y_l^m(\theta_f, \phi_f)$$

which is used to construct the rotationally invariant bond order parameters (also see [11]):

$$\bar{q}_l(i) = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |q_{lm}(i)|^2}, \tag{1}$$

and

$$\bar{w}_l(i) = \frac{\sum\limits_{m_1+m_2+m_3=0} \begin{vmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{vmatrix} \bar{q}_{lm_1}(i)\bar{q}_{lm_2}(i)\bar{q}_{lm_3}(i)}{\left(\sum\limits_{m=-l}^{l} |\bar{q}_{lm}(i)|^2\right)^{3/2}}. \tag{2}$$

2.2 Machine Learning

Machine learning techniques can be roughly classified into four categories: supervised learning, unsupervised learning, semi-supervised learning and reinforced learning. The categories are distinguished by the type of available data and the objective of the procedure (see figure 4 for an overview). Here, a brief elaboration is given to distinguish the different types and their purposes.

Supervised learning uses labeled data sets where both the input and output are known. The goal of its learning procedure is to find a quantitative model to relate input and output such that it can predict outcomes for new, unknown data input.

Unsupervised learning is used for unlabeled data sets. The basic idea is to find an underlying structure or relationships within the data based only on the inputs. Unsupervised learning can further be classified into clustering algorithms and dimensionality reduction techniques.

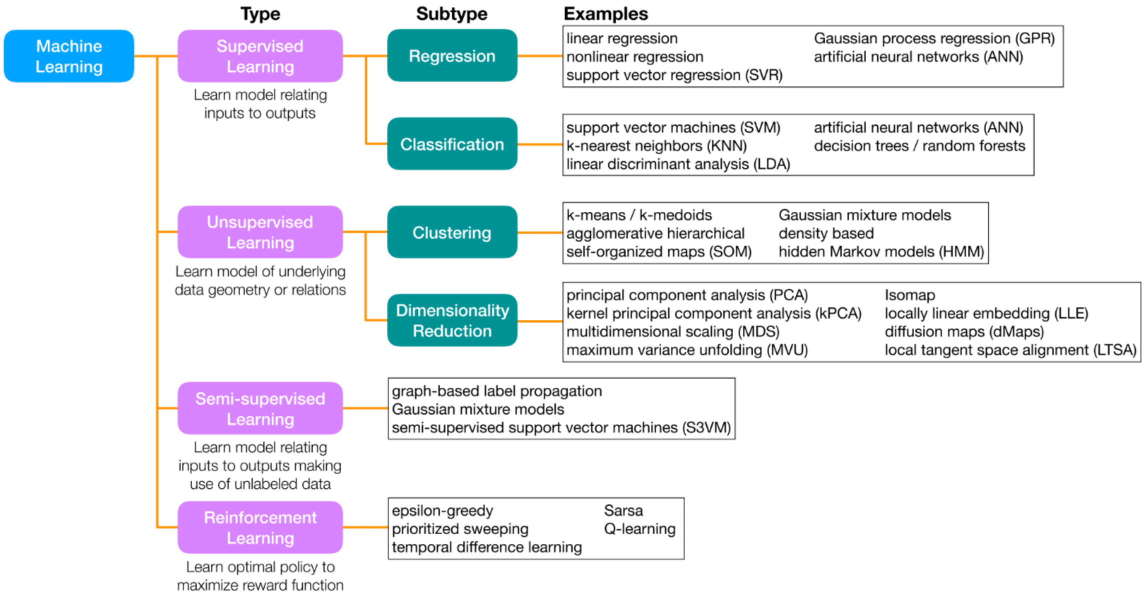

Type
Subtype
Examples
Machine Learning
Supervised Learning
Learn model relating inputs to outputs
Regression
linear regression
nonlinear regression
support vector regression (SVR)
Gaussian process regression (GPR)
artificial neural networks (ANN)
Classification
support vector machines (SVM)
k-nearest neighbors (KNN)
linear discriminant analysis (LDA)
artificial neural networks (ANN)
decision trees / random forests
Unsupervised Learning
Learn model of underlying data geometry or relations
Clustering
k-means / k-medoids
agglomerative hierarchical
self-organized maps (SOM)
Gaussian mixture models
density based
hidden Markov models (HMM)
Dimensionality Reduction
principal component analysis (PCA)
kernel principal component analysis (kPCA)
multidimensional scaling (MDS)
maximum variance unfolding (MVU)
Isomap
locally linear embedding (LLE)
diffusion maps (dMaps)
local tangent space alignment (LTSA)
Semi-supervised Learning
Learn model relating inputs to outputs making use of unlabeled data
graph-based label propagation
Gaussian mixture models
semi-supervised support vector machines (S3VM)
Reinforcement Learning
Learn optimal policy to maximize reward function
epsilon-greedy
prioritized sweeping
temporal difference learning
Sarsa
Q-learning

Figure 4: An overview of machine learning types and their algorithms from a topical review on machine learning in soft materials engineering [12].

Clustering is the process of finding points within a data set that are more similar than others and divides the data into clusters. With dimensionality reduction, also known as manifold learning, low-dimensional parametrizations are determined within high-dimensional data sets. Dimensionality reduction can again be classified into linear and nonlinear techniques. In this thesis, both will be used for our research.
With semi-supervised learning, one tries to make optimal use of both labeled and unlabeled data to achieve a better model performance than using either type alone.
Reinforced learning differs from the other types of learning. It seeks an optimal policy for interacting with a dynamic situation to maximize a reward function rather than finding an input-output model or looking for structure in a data set.

2.2.1 Principal Component Analysis

Principal component analysis (PCA) is a linear unsupervised dimensionality reduction technique. This technique is closely related to other techniques like the Karhunen-Love transform, factor analysis and proper orthogonal decomposition. These techniques are linear in the sense that they are constrained to collapse the high-dimensional data onto linear hyperplanes. For PCA, the idea is to search for a set of orthogonal vectors (principal components) that contain the maximum variance in the data points. Ordering these vectors according to the fraction of variance often yields a small number of components and leaves an effective low-dimensionality of the data set.

Mathematically, one considers N observations of d dimensions each, i.e. $\{x_i(t)\}_{i=1}^{d}$, with $t = 1, \cdots, N$. These observations are stored in a $d \times N$ observation matrix
$\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_d})^T$ and mean center $\mathbf{X}$ to insure that the decomposition is independent of

the arbitrary location of the data cloud in its coordinate space (i.e. $\sum_{j=1}^{N} X_{ij} = 0, \forall i$). The $d \times d$ covariance matrix is constructed as follows

$$\mathbf{C} = \frac{1}{N-1}\mathbf{X}\mathbf{X}^T,$$

where C_{ij} is the measure of the covariance between measurement dimensions i and j. Next, for each observation, a vector $\mathbf{W} = \mathbf{V}^T\mathbf{X}$ is found with $\mathbf{V}$ an orthonormal $d \times d$ matrix such that the covariance matrix is diagonal in the new basis. Matrix $\mathbf{V}$ possesses by definition the right eigenvectors of covariance matrix $\mathbf{C}$. If a data set can appropriately be decomposed in orthogonal vectors, a gap in the eigenvalue spectrum is found which separate the first $k < d$ principal components. This is the part where the dimensionality is reduced. The relevant eigenvectors of $\mathbf{V}$ are left over (corresponding to the first k eigenvalues). We can now project the original observation matrix onto these vectors,

$$\mathbf{W}_{proj} = \mathbf{V}_{proj}^T\mathbf{X},$$

where $\mathbf{W}_{proj}$ is a matrix of size $k \times N$ and where the columns contain the projection of each data point onto the low-dimensional PCA subspace.

2.2.2 Diffusion Maps

Diffusion Maps (dMaps) is an unsupervised nonlinear dimensionality reduction technique. Where PCA achieves dimensionality reduction by finding components with maximum variance in the data, dMaps can make low-dimensional projections onto curvilinear manifolds which makes its a generally more applicable method.

The method consists of constructing a random walk over the high-dimensional point cloud followed by a harmonic analysis to find its slowest relaxation modes.
The methods consists in a couple of steps. Consider N data points $\{\mathbf{x}_i\} \in \mathcal{R}^K$ in a K-dimensional space with $i = 1, \cdots, N$. The distances $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ between all N points are computed and stored in a $N \times N$ distance matrix $\mathbf{d}$, with $\|\cdot\|$ an appropriate distance metric. In many cases, as for our own case, the Euclidean metric will serve as a good measure. Next, a Gaussian kernel with standard deviation ε is added to the distance matrix to form

$$\mathbf{A} = \begin{pmatrix} e^{-d_{1,1}^2/2\varepsilon} & e^{-d_{1,2}^2/2\varepsilon} & \dots & e^{-d_{1,N}^2/2\varepsilon} \\ e^{-d_{2,1}^2/2\varepsilon} & e^{-d_{2,2}^2/2\varepsilon} & \dots & e^{-d_{2,N}^2/2\varepsilon} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-d_{N,1}^2/2\varepsilon} & e^{-d_{N,2}^2/2\varepsilon} & \dots & e^{-d_{N,N}^2/2\varepsilon} \end{pmatrix}, \tag{3}$$

where $A_{ij} \in [0, 1]$ can be seen as the non normalized hopping probabilities of the random walk from $\mathbf{x}_i$ to $\mathbf{x}_j$ with characteristic step size ε. An appropriate value of ε has to be chosen and is dependent of the structure of the data set. Now, matrix $\mathbf{A}$ is normalized by its row sums to construct the Markov matrix $\mathbf{M} = \mathbf{D}^{-1}\mathbf{A}$ with $\mathbf{D}$ diagonal matrix with elements $D_{ij} = \sum_j A_{ij}$. M_{ij} can now be seen as the normalized hopping probabilities of the random walk from $\mathbf{x}_i$ to $\mathbf{x}_j$. Matrix $\mathbf{M}$ can now be diagonalized to find the eigenvalues $\{\lambda\}_{i=1}^{N}$ and corresponding eigenvectors $\{\psi\}_{i=1}^{N}$. Markov matrices hold the property that it always contains a trivial eigenvalue and its associated eigenvector that correspond to a stationary random walk, i.e. $\lambda_1 = 1$ and $\psi_1 = \mathbf{1}$. Looking at the eigenvalues in non-ascending order, the remaining eigenvalues leave a gap between λ_{k+1} and λ_{k+2} which indicated the division between the slow and fast diffusion modes. This is where the dimensionality is reduced $K \to k$. The eigenvectors corresponding to the slow diffusion modes provide a good parametrization for the low-dimensional manifold, each point $\mathbf{x}_i$ is projected onto the i^{th} component of the k leading non-trivial eigenvectors, the so called diffusional map embedding,

$$\mathbf{x_i} \to \Big(\psi_2(i), \psi_3(i), ..., \psi_{k+1}(i)\Big). \tag{4}$$

3 Methods

In this research, crystal structures of systems with “odd shaped” particles are studied. Specifically, systems with increasingly rounded cubes, octahedra and tetrahedra are investigated. The roundedness of the particles, represented by parameter r, is varied in steps of 0.01 between (and including) 0 and 1.

Crystals are composed of a small group of particles called unit cells. A unit cell is repeated through the whole crystal structure along its lattice vectors and completely reflects the structure and symmetry of the crystal. The number of particles per unit cell N of the densest packing depends on the shape of the particles we chose for our system. All shapes used in this research are eventually rounded up to perfect spheres, which have a densest packing with a (primitive) unit cell of $N = 1$ (face centered cubic system). Cubes (simple cubic lattice) and octahedra are known for having $N = 1$ particles per unit cell for their densest packing [2]. The densest packing for tetrahedra is known to have $N = 4$ particles per unit cell, however in a paper on spherotetrahedra, a dense packing with a unit cell with $N = 2$ is also found [6]. It is plausible that for a specific rounding of the considered shapes, another number of particles per unit cell N exists (although this is unlikely for the cubes and octahedra). If we would be interested in the densest packed systems for these particles, the number of unit cells N would have to be found for each rounding of the particles. However, in this research the focus lies more on the identification of the crystal structures with use of machine learning. Therefore, the number of particles per unit cell is fixed for the systems considered in this research, specifically $N = 1$ for the rounded cubes, $N = 1$ for the rounded octahedra and $N = 2$ for the rounded tetrahedra.

Furthermore in this section, the particle simulation is described, which grant the bond order parameters that are used for the machine learning analysis. Also a short description of how this analysis is done for each dimensionality reduction method. The output of the simulation, i.e. the bond order parameters $\bar{q}_l$ and $\bar{w}_l$ with $l \in [1, 12]$ are obtained as well as the packing fraction for each simulated system. This output provides the necessary input for the techniques described in the theory (section 2.2.1 and 2.2.2) that will be used to make the analysis. A brief description of how this is done will also be provided in this section. Furthermore the analysis is done in Python and the scripts with annotations will be provided along with this report.

3.1 Simulation

For the simulation part of this research, the floppy-box Monte Carlo (FBMC) method [13] is used. Here, a description for the setup for the algorithm is given.
The FBMC algorithm is based on a standard Monte Carlo (MC) simulation for an ensemble with a fixed number of particles, pressure and temperature (NPT or isothermal-isobaric ensemble). There are a few key points in which the FBMC is essentially different from a standard MC simulation. First, the system is simulated by creating one single unit cell with a small number of particles which is then copied to create the full system. Furthermore, additionally to the standard MC moves the shape and size of the box is also varied in each MC step. Finally the term fixed pressure should be specified here. The algorithm is meant to simulate close packed systems which is accomplished by increasing the pressure in the system towards higher densities. The pressure is therefore not fixed at one value but the algorithm fixes the pressure to an value that increases during the course of the simulation. A detailed description of the method will be elaborated in this section.

Consider N particles in a box with the positions to the particles centre $\mathbf{r}_i$ and orientation $\mathbf{q}_i$, where $i = 1, 2, ..., N$. The orientation is obtained by applying a rotation matrix to an initial orientation. The box is defined by three vectors $\mathbf{v}_j$, with $j = 1, 2, 3$. One of the corners of the box is centred in the origin and all of the vectors are given with respect to a standard Cartesian coordinate system. The packing fraction of the system is given by $\phi = 1/V \sum_{j=1}^{N} V_j$ with V_j the volume of particle j and $V \equiv |\mathbf{v}_1 \cdot (\mathbf{v}_2 \times \mathbf{v}_3)|$ the volume of the simulation box. The total potential energy of the system is given by $U = U(\mathbf{v}^3, \mathbf{r}^N, \mathbf{q}^N)$. Since we consider only hard-particle interactions U can only take two values,

$$\beta U = \begin{cases} 0 & \text{when there is no overlap between particles,} \\ \infty & \text{when two particles overlap,} \end{cases} \tag{5}$$

with $\beta = 1/k_bT$, k_b being the Boltzmann constant and T the temperature. One of the differences to a standard Monte Carlo simulation is that the number of particles in the box is small, typically $N < 12$. In addition to to the standard MC moves, translation and rotation, scaling and deformation of the box is also applied to the box where the perturbations for these moves are constructed randomly. These four moves all have to be done while maintaining the NPT ensemble and while satisfying the overlap restriction. Therefore an acceptance criterion is defined which has to be satisfied. The probability of a move being accepted written as $acc(o \rightarrow n)$ with "o'' and "n'' represent respectively the old and new state. The acceptance criterion for these moves is given by

$$acc(o \rightarrow n) = min\left(1, exp\left[-\beta(U_n - U_0) + P(V_n - V_0) + (N+1)\log\left(\frac{V_n}{V_0}\right)\right]\right), \tag{6}$$

where P is the pressure and subscripts n and o indicating the new and old values in the system.

The order in which these moves are made is random but requires a certain probability distribution in order to make an efficient sampling. For $N = 1$, efficient sampling is found for roughly 70% rotation, 15% scaling, 15% deformation and no translation is applied. For $N \geq 2$ we find efficient sampling for typically 35% translation, 35% rotation, 15% scaling and 15% deformation moves.

After each of these moves, two criteria have to be satisfied. No overlap is allowed and the acceptance criteria in equations 6 and **??** have to be satisfied. Checking for overlaps in the system needs its own algorithm and can be computationally expensive and time consuming. There are several methods to check hard-article overlap. In this simulation the Gilbert-Johnson-Keerthi (GJK) overlap algorithm is used [14]. Since this overlap check can be rather time consuming, it is first ensured the move is accepted on basis on pressure and volume (equations 6 and **??**) before doing the overlap check.

For this overlap check, as well as for the moves that are applied a, set of scaled coordinates is introduced. Consider $\mathbf{r}_j = M\mathbf{s}_j$, where $\mathbf{s}_j \in [0,1)^3$ is the scaled coordinates with off course $j = 1, ..., N$ and matrix $\mathbf{M} = (\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3)$ where the vectors are the columns of the matrix.

The next step is to construct the whole system by making periodic images of the box, to check for overlaps in the system and moreover to make enough nearest neighbours to calculate the bond order parameters. Checking for overlap in the whole system is essentially equivalent to checking for a particle in a box and another particle in the box, its own periodic images and other particles' periodic images. This is done for every particle making sure they are not double checked. For efficiency, a minimum amount of periodic images have to be constructed without going at the cost of the overall speed of algorithm.

Consider a particle in the origin with largest outscribed-sphere R_O. For two particles with a distance $2R_O$ from each other, no overlap takes place. A sufficiently large self-image list is constructed by considering a cube with vertices $\mathbf{c}_n = 2R_O(\pm\hat{\mathbf{x}} \pm \hat{\mathbf{y}} \pm \hat{\mathbf{z}})$ with $n = 1, ..., 8$ and $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ Carthesian unit vectors, which envelops the sphere at the origin with radius $2R_O$. The inverse matrix M^{-1} is applied to the the vertices of the cube $\mathbf{c}_n$ to obtain parallelepiped $\mathbf{p}_n = M^{-1}\mathbf{c}_n$. The upper bounds to the number of images that need to be checked in each direction are constructed using $\mathbf{p}_n$

$$\begin{aligned} N_1 &= \lceil \max_n(\mathbf{p}_n \cdot \hat{\mathbf{x}}) \rceil; \\ N_2 &= \lceil \max_n(\mathbf{p}_n \cdot \hat{\mathbf{y}}) \rceil; \\ N_3 &= \lceil \max_n(\mathbf{p}_n \cdot \hat{\mathbf{z}}) \rceil; \end{aligned} \tag{7}$$

where $\lceil \cdot \rceil$ indicates the ceiling function ($\lceil a \rceil$ gives the smallest integer larger than a). For images that fall outside off the rectangle $[-N_1, N_1] \times [-N_2, N_2] \times [-N_3, N_3]$, no overlap with the particle at the origin is possible. The equivalent set of images can now be established as $P_{im} = i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3$ with $i = -N_1, ..., N_1$; $j = -N_2, ..., N_2$; $j = -N_3, ..., N_3$; and $i + j + k \neq 0$. When this list is checked for overlap the Monte Carlo cycle is over. A default of 10^7 MC cycles is used to find to try and determine the crystal structure. When the MC cycles are finished, the bond order parameters $\bar{q}_l$ and $\bar{w}_l$ with $l \in [1, 12]$ of the crystal structure is calculated and stored.

This routine is repeated for $0 \leq r \leq 1$ with steps of 0.001, i.e. per system, two data sets (one for $\bar{q}_l$ and one for $\bar{w}_l$) with 1001 12-dimensional points are obtained. In the following sections, a short description of the analysis process of these data sets.

3.2 Principal component analysis

Principal component analysis is relatively straight forward to implement by following the theory. Here, a short description is given.

First, the data set is stored in the 12×1001 observation matrix $\mathbf{X}$. The 12×12 covariance matrix $\mathbf{C}$ is now constructed from which we obtain the eigenvalues λ_d and eigenvectors $\mathbf{w}_d$ with $d \in [1, 12]$ as described in section 2.2.1. The eigenvalues can now be plotted to find the eigenvalue gap and find the reduced dimension $k < d$. Furthermore, the eigenvectors $\mathbf{w}_d$ corresponding to the eigenvalues λ_d with $d \in [1, k]$ are plotted as a function of $\mathbf{x}_i$ with index $i \in [0, 1000]$. $\mathbf{x}_i$ can essentially be interpreted as roundness parameter $r \in [0, 1]$. Furthermore, it is possible to find out what bond order parameters $\bar{q}_l$ and $\bar{w}_l$ contribute most to the eigenvectors $\mathbf{w}_k$ by projecting them upon the directions of the bond order parameters $\bar{q}_l$ (and $\bar{w}_l$) in the original data set (mathematically these are just the eigenvectors of the 12×12 identity matrix).

3.3 Diffusion maps

Implementing diffusion maps takes a few more steps. A short description is given here.

Like described in the theory section 2.2.2, the distances $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ between all 12-dimensional pair points $\mathbf{x}_i$ and $\mathbf{x}_j$ with indices $i, j \in [0, 1000]$ are calculated and stored in the 1001×1001 matrix $A_{ij} = e^{-d_{ij}^2/2\varepsilon}$. Now, an appropriate value of ε is to be determined. This is done by plotting the sum of matrix $\mathbf{A}$ as a function of the $\log_{10} \varepsilon$ which will show a flat curve for too small and too large values of epsilon. By trial and error, the best value of epsilon is found under condition that it is chosen within the range of ε where the slope of the sum of $\mathbf{A}$ as a function of $\log_{10} \varepsilon$ is non zero. Now that an appropriate value of ε is chosen, Markov matrix $\mathbf{M}$ is constructed (see theory section 2.2.2) and eigenvalues λ_i and eigenvectors ψ_i are obtained with $i \in [0, 1000]$. The eigenvalues λ_i are plotted to find the gap between that reduces the dimensionality $12 \to k$. The eigenvectors ψ_d corresponding to the eigenvalues λ_d with $d \in [1, k]$ are plotted as a function of $\mathbf{x}_i$ with index $i \in [0, 1000]$. Again, $\mathbf{x}_i$ can essentially be interpreted as roundness parameter $r \in [0, 1]$.

4 Results

In this section, both the results obtained from the simulation as well as the results obtained from the analysis with the machine learning techniques are reported for each of the studied systems. The results are reported and evaluated per system in the following order; first the simulation output, secondly the PCA results, then the results for diffusion maps and finally, for the rounded cubes and octahedra system, snapshots of the simulation are discussed.

4.1 Cubes

In figure 5, the bond order parameters (BOP's) $\bar{q}_1$ to $\bar{q}_{12}$ are plotted as a function of roundedness for the system of rounded cubes for FBMC simulations with unit cell $N = 1$. Note that only 6 of the 12 bond order parameters can be seen in the graph. In fact, only the parameters $\bar{q}_2$, $\bar{q}_4$, $\bar{q}_6$, $\bar{q}_8$, $\bar{q}_{10}$ and $\bar{q}_{12}$ (all even BOP's) show curves. The other BOP's are all of order of magnitude 10^{-16}. Most importantly, we notice the jump in the BOP's just before $r = 0.7$. This jump indicates a phase transition between two crystal structures. We therefore also expect to find this transition in the results for PCA and diffusion maps.

Furthermore, a “messy” part in the region $0 < r < 0.2$ is noticed (also more devious data points are found through the whole range). We have to keep in mind that the FBMC simulation initially looks for the densest packing. However, if for a system several different dense packed structures exist, there is a good chance the FBMC simulation gives *a* dense packing and not *the* densest one. For a research where the densest packing is studied it would be important to run the simulation a number of times and chose the densest packing for each run to be considered in the study. In this case however, we are not so much interested in the densest packing but more in whether machine learning can identify different structures and phase transitions. The simulations are also particularly time consuming (some of them can take up to a week) so for these reasons we have decided to do the machine learning analysis for one single run per system. Moreover it can also be useful and interesting to see how PCA and diffusion maps react to noisy data.

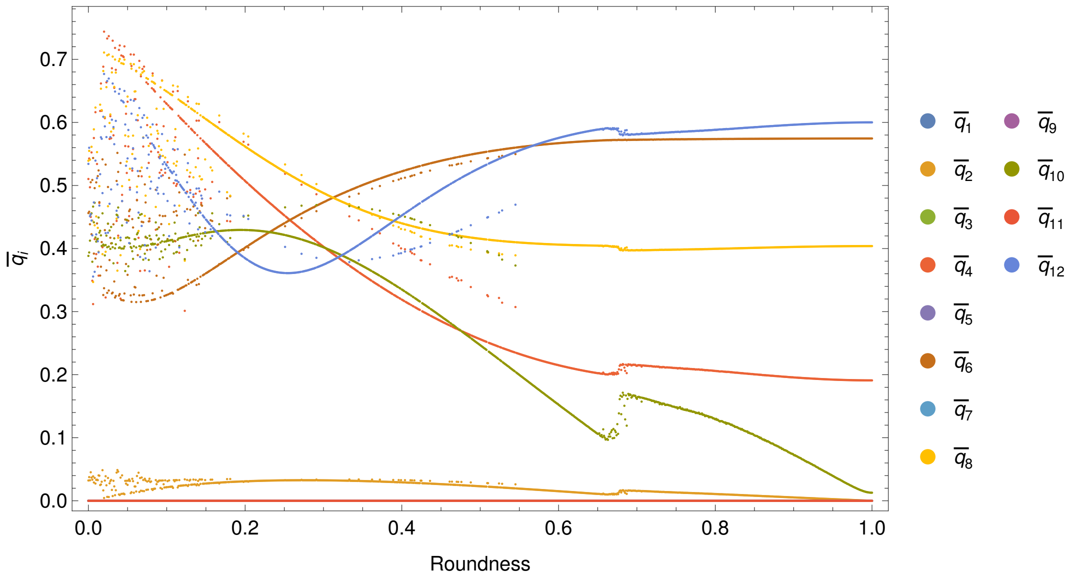
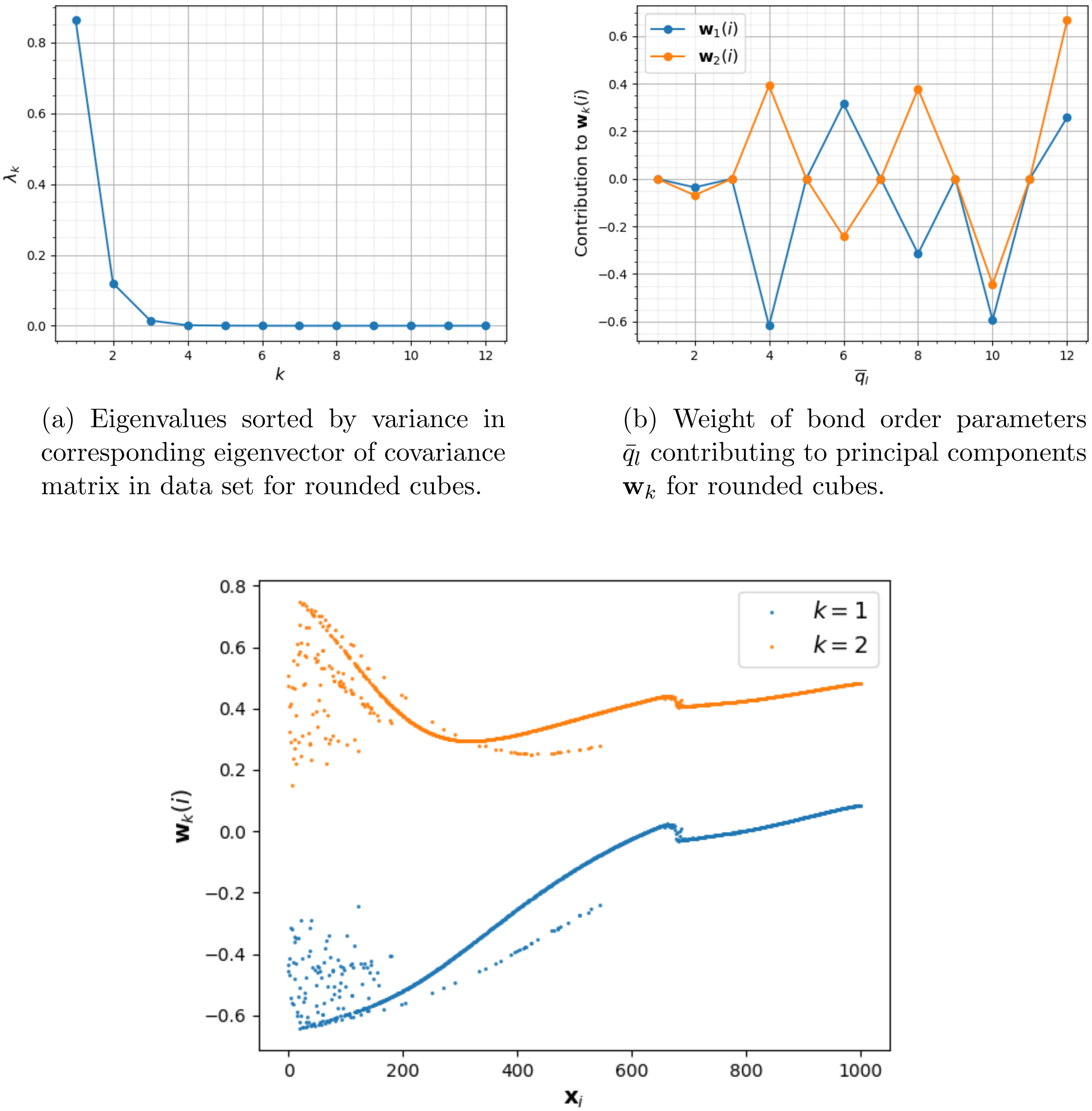


Figure 5: Bond order parameters q_1 to q_{12} plotted as a function of the roundedness for rounded cubes with unit cell $N = 1$.

4.1.1 PCA

In figure 6 we see the PCA results for the rounded cubes system. To measure how the dimensionality is reduced for each system, the 12 normalized eigenvalues λ_k of the covariance matrix of the BOP data set for the rounded cubes are plotted non-ascending order in figure 6a. Like expected from the theory (section 2.2.1), we find a gap in the value of these eigenvalues which quantify the effective dimensionality to where the original data set is reduced. In this case we see one high value, a lower one and the rest effectively zero. This indicates that the first eigenvector contains predominantly most of the variance in the original data set. The second eigenvector still contains some variance in the original data set but with these two vectors, the whole data set can effectively be reconstructed.
Figure 6b shows which bond order parameters $\bar{q}_l$ contribute to the eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA. We see that $\bar{q}_4$ and $\bar{q}_{10}$ contribute most to eigenvector $\mathbf{w}_1$ where $\bar{q}_6$, $\bar{q}_8$ and $\bar{q}_{12}$ have a smaller contribution. For eigenvector $\mathbf{w}_2$, $\bar{q}_{12}$ has the biggest contribution where $\bar{q}_4$, $\bar{q}_6$, $\bar{q}_8$ and $\bar{q}_{10}$ have a smaller contribution.

In figure 6c the eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are plotted respectively as function of $\mathbf{x}_i$ (roundedness). Notice the similarity of the curve with the BOP plots in figure 5. This is not a surprise since we have seen in figure 6b how $\mathbf{w}_1$ and $\mathbf{w}_2$ are constructed from the BOP's. The jump in the plot at $\mathbf{x}_i$ with $i \approx 700$ (corresponding to roundedness parameter $r = 0.7$) can be clearly seen in both $\mathbf{w}_1$ and $\mathbf{w}_2$ like we expected from 6b.



(a) Eigenvalues sorted by variance in corresponding eigenvector of covariance matrix in data set for rounded cubes.

(b) Weight of bond order parameters $\bar{q}_l$ contributing to principal components $\mathbf{w}_k$ for rounded cubes.

(c) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for rounded cubes.

Figure 6: Results from PCA for rounded cubes.

4.1.2 dMaps

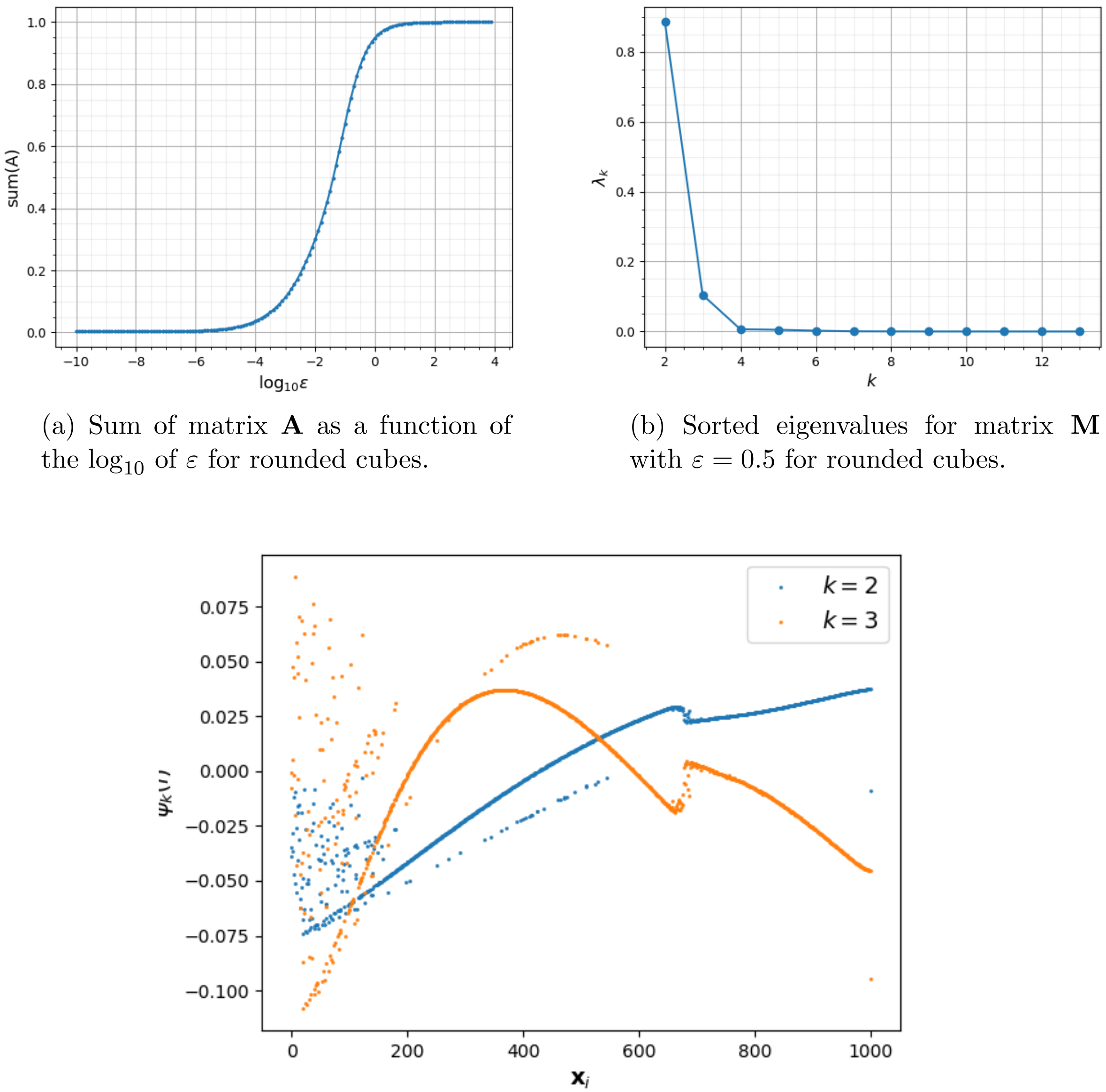
Figure 7 shows the results obtained from diffusion maps on the rounded cubes system. Before applying diffusion maps on a data set, an appropriate value for ε has to be found. Like explained in the theory 2.2.2 the value of ε is a measure for the connectedness of the datapoints. For too small values of ε the data points are not connected to other data points at all, but for too large values, all data points are connected to all other data points. To find an appropriate value, the sum of matrix $\mathbf{A}$ is taken as a function of the $\log_{10}$ of ε (see figure 7a). The flat regions in this plot correspond to the too small and too large regions for epsilon (the sum of $\mathbf{A}$ does not decrease/grow any more for a smaller/larger ε/connectedness between data points). Hence, ε has to be chosen somewhere in between these regions for diffusion maps to have any meaning. By trial and error, we have found that the upper part of the slope gives the best results. For this data set, a value of $\varepsilon = 0.5$ is chosen.

In figure 7b the first 12 out of 1001 (normalized) eigenvalues of matrix $\mathbf{M}$ (see theory 2.2.2) are sorted and plotted for the measure of dimensionality reduction for this system. Again, like with PCA, a clear gap between the first and second eigenvalue can be identified after which all other eigenvalues are (practically) zero. In fact, the dimensionality reduction plots for PCA and diffusion maps look very much alike. This will be discussed more in another section.

In figure 7c the nontrivial eigenvectors $\psi_2(i)$ and $\psi_3(i)$ of matrix $\mathbf{M}$ are plotted as a function of $\mathbf{x}_i$. Again, the similarities with the curves in the bond order parameter plots in 5 are strong. In fact the plots in 7c seem to look exactly like the plots found with PCA in figure 6c except for that $\psi_2(i)$ is flipped. Also the actual values of the eigenvectors seem to be translated and scaled, but their form is obviously similar, which is remarkable. Again the jump in the plot just before $\mathbf{x}_{700}$ (corresponding to roundedness parameter $r = 0.7$) can be clearly seen in like we expected from 6b.

4.1.3 Simulation snapshots around phase transition

A clear gap can be identified for roundedness parameter r in $0.660 < r < 0.679$ in the BOP plots (figure 5), results from PCA (figure 6c) and diffusion maps (figure 7c). This suggests a phase transition in this region and it is therefore worth while to trace back these simulation runs and to take a look at the simulation snapshots (see figure 8). However, due to the way the crystals are constructed by the FBMC simulation (orientation of unit cells, etc.) it is extremely hard find similarities or to distinct the structures from each other. It is very well possible two crystals don't look similar from the snapshots (say 8h and 8i) but in fact are similar (after a slight rotation around one/all of the axes or translation).

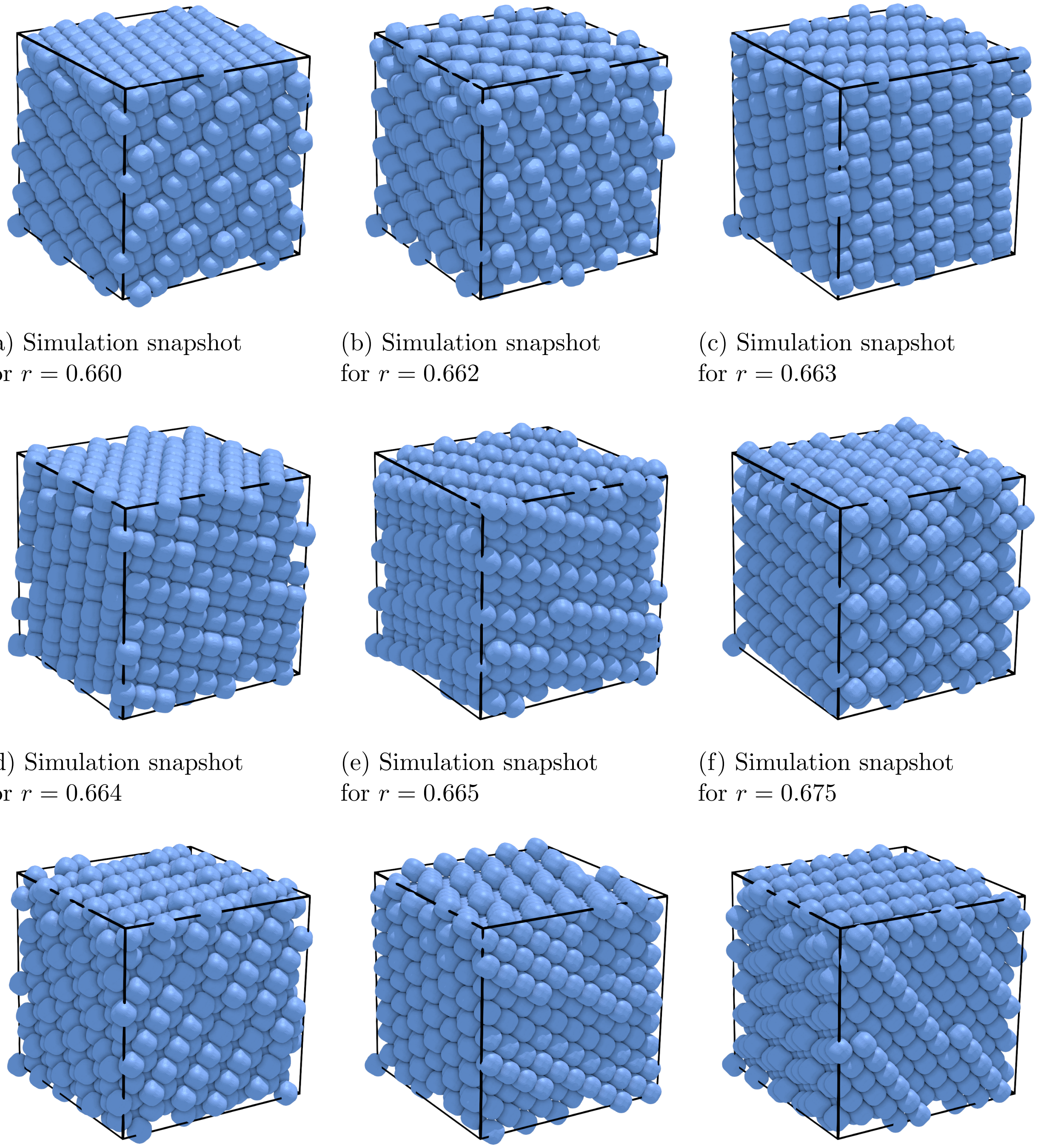


(a) Sum of matrix **A** as a function of the $\log_{10}$ of ε for rounded cubes.

(b) Sorted eigenvalues for matrix **M** with $\varepsilon = 0.5$ for rounded cubes.

(c) Eigenvectors $\psi_k(i)$ of matrix **M** plotted as a function of i with $\varepsilon = 0.5$ in data set for rounded cubes.

Figure 7: Results from diffusion maps for rounded cubes.



(a) Simulation snapshot for $r = 0.660$

(b) Simulation snapshot for $r = 0.662$

(c) Simulation snapshot for $r = 0.663$

(d) Simulation snapshot for $r = 0.664$

(e) Simulation snapshot for $r = 0.665$

(f) Simulation snapshot for $r = 0.675$

(g) Simulation snapshot for $r = 0.677$

(h) Simulation snapshot for $r = 0.678$

(i) Simulation snapshot for $r = 0.679$

Figure 8: A few snapshots from the simulation of rounded cubes from $r = 0.660$ to $r = 0.679$.

4.2 Octahedra

In figure 9 the bond BOP's $\bar{q}_1$ to $\bar{q}_{12}$ are plotted as a function of roundedness for octahedra. Again only the parameters $\bar{q}_2$, $\bar{q}_4$, $\bar{q}_6$, $\bar{q}_8$, $\bar{q}_{10}$ and $\bar{q}_{12}$ (all even BOP's) are non-zero and actually show curves in their plots. The plots look less noisy than the BOP plots for the rounded cubes (figure 5). In these plots, two clear jumps can be distinguished for roughly around $r = 0.3$ and $0.8 < r < 0.9$. The jumps here are slightly different than for the jumps in the plots for rounded cubes (figure 5). It not only suggests a phase transition, there also seems to be a degeneracy for the regions $r = 0.3$ and $0.8 < r < 0.9$ where there are two competing phases. Again, this is also expected to be found in the results for PCA and diffusion maps.

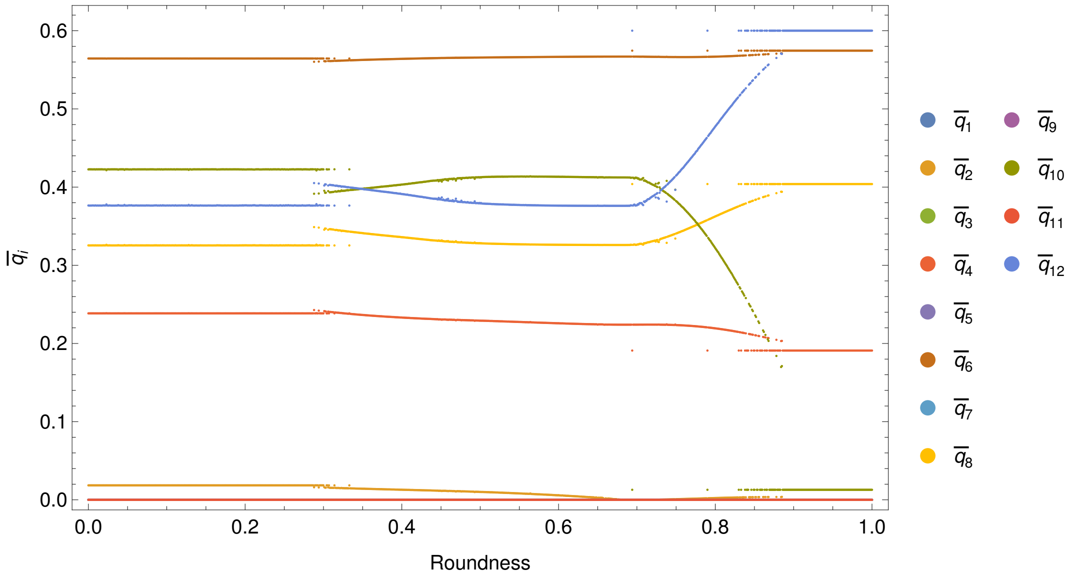
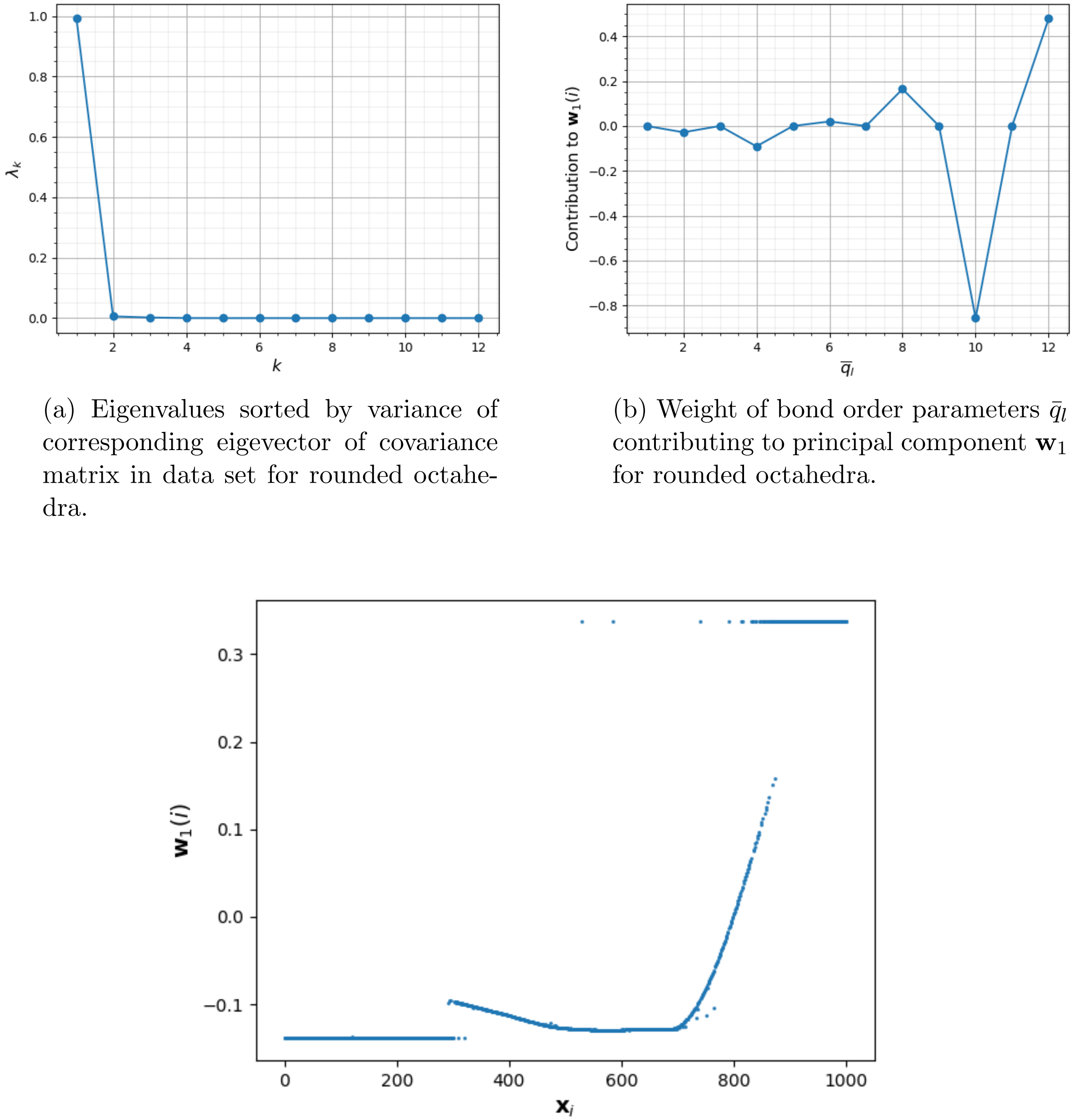


Figure 9: Bond order parameters q_1 to q_{12} plotted as a function of the roundedness for rounded octahedra with unit cell $N = 1$.

4.2.1 PCA

Figure 10 shows the PCA results for the rounded octahedra system. Again we want to measure how PCA reduces the dimensionality for this system. In figure 10a, the 12 normalized eigenvalues of the covariance matrix of the BOP data set for the rounded octahedra, which are plotted and sorted in non-ascending order. This time, an even clearer gap is found between the first eigenvalue and the rest of the eigenvalues. As we see in figure 10a, the first eigenvalue takes value 1 which means the variance in the original data set can be fully expressed with one eigenvector and that the whole data set can be effectively expressed with this vector.
Figure 6b shows how this eigenvector $\mathbf{w}_1$ is constructed from the BOP's. We see that $\bar{q}_{10}$ is the largest contributor to $\mathbf{w}_1$ by far, $\bar{q}_{12}$ also contributes a bit and $\bar{q}_4$ and $\bar{q}_8$ even less.

In figure 10c eigenvector $\mathbf{w}_1$ is plotted. Again, the similarities with the BOP plots (figure 9) is clear. In fact, the shape of the curve looks exactly like $\bar{q}_{10}$, which is in agreement with the plots in figure 10b. Like expected from the jumps in the BOP plots (figure 9), the jumps around $r = 0.3$ and $0.8 < r < 0.9$ are very clear and the same degeneracy for these regions of r can be distinguished as in the BOP plots.



(a) Eigenvalues sorted by variance of corresponding eigevector of covariance matrix in data set for rounded octahedra.

(b) Weight of bond order parameters $\bar{q}_l$ contributing to principal component $\mathbf{w}_1$ for rounded octahedra.

(c) Eigenvector $\mathbf{w}_1$ found by PCA as a function of roundedness $\mathbf{x}_i$ for rounded cubes.

Figure 10: Results from PCA for rounded octahedra.

4.2.2 dMaps

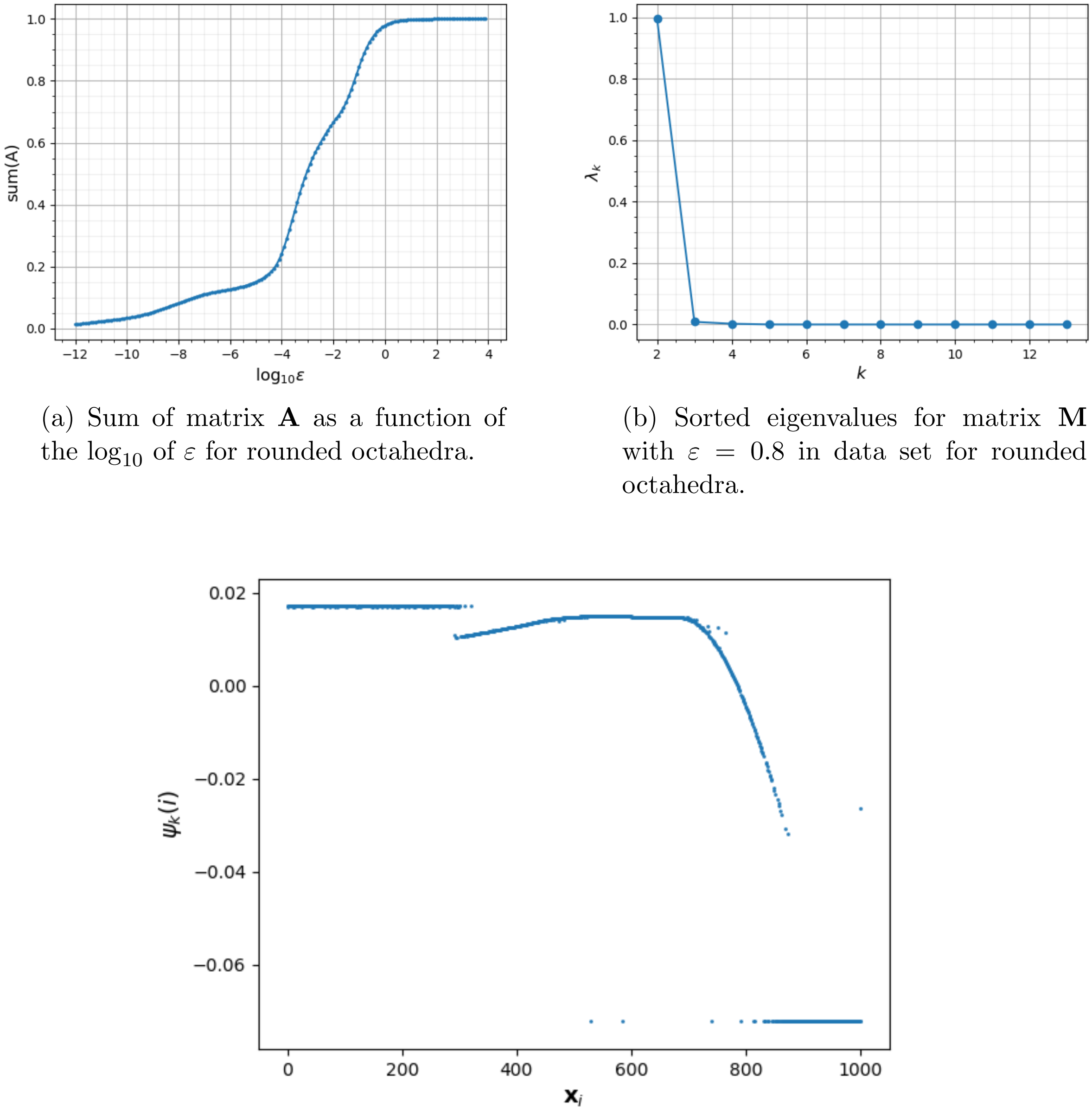
Figure 11 shows the results obtained from diffusion maps on the rounded octahedra system. Again, the appropriate value is found by taking the sum of matrix $\mathbf{A}$ as a function of the $\log_{10}$ of ε (see figure 7a). A value in the upper part of the slope is chosen, i.e. $\varepsilon = 0.8$.

Figure 11b shows the first 12 out of 1001 (normalized) eigenvalues of matrix $\mathbf{M}$ (see theory 2.2.2), which are sorted and plotted for the measure of dimensionality reduction for this system. Just like with PCA, the first non-trivial eigenvalue takes value 1 wheres the other eigenvalues are (practically) zero which means we can express the data set with the first non-trivial eigenvector $\psi_2(i)$.

This eigenvector is plotted in 11c as a function of $\mathbf{x}_i$. Again, the curve in this plot is the same as the curve in the plot we find with PCA. The jumps in the plot correspond to the jumps the BOP plots (figure 9). The apparent phase transitions seem to be unanimous in the BOP plots, the results PCA and diffusion maps, i.e. around roundedness parameter $r = 0.3$ and $0.8 < r < 0.9$.

4.2.3 Simulation snapshots around phase transition

A clear gaps can be identified for roundedness parameter around $r \approx 0.3$ and around $r \approx 0.83$ in the BOP plots (figure 10b), results from PCA (figure 10c) and diffusion maps (figure 11c). This suggests phases transition in these regions and it is therefore worth while to trace back these simulation runs and to take a look at the simulation snapshots (see figures 12 and 13). However, due to the way the crystals are constructed by the FBMC simulation (orientation of unit cells, etc.) it is extremely hard find similarities or to distinct the structures from each other. It is very well possible two crystals don't look similar from the snapshots (say 8h and 8i), due to a slight rotations around one/all of the axes or translations, but in fact are similar.

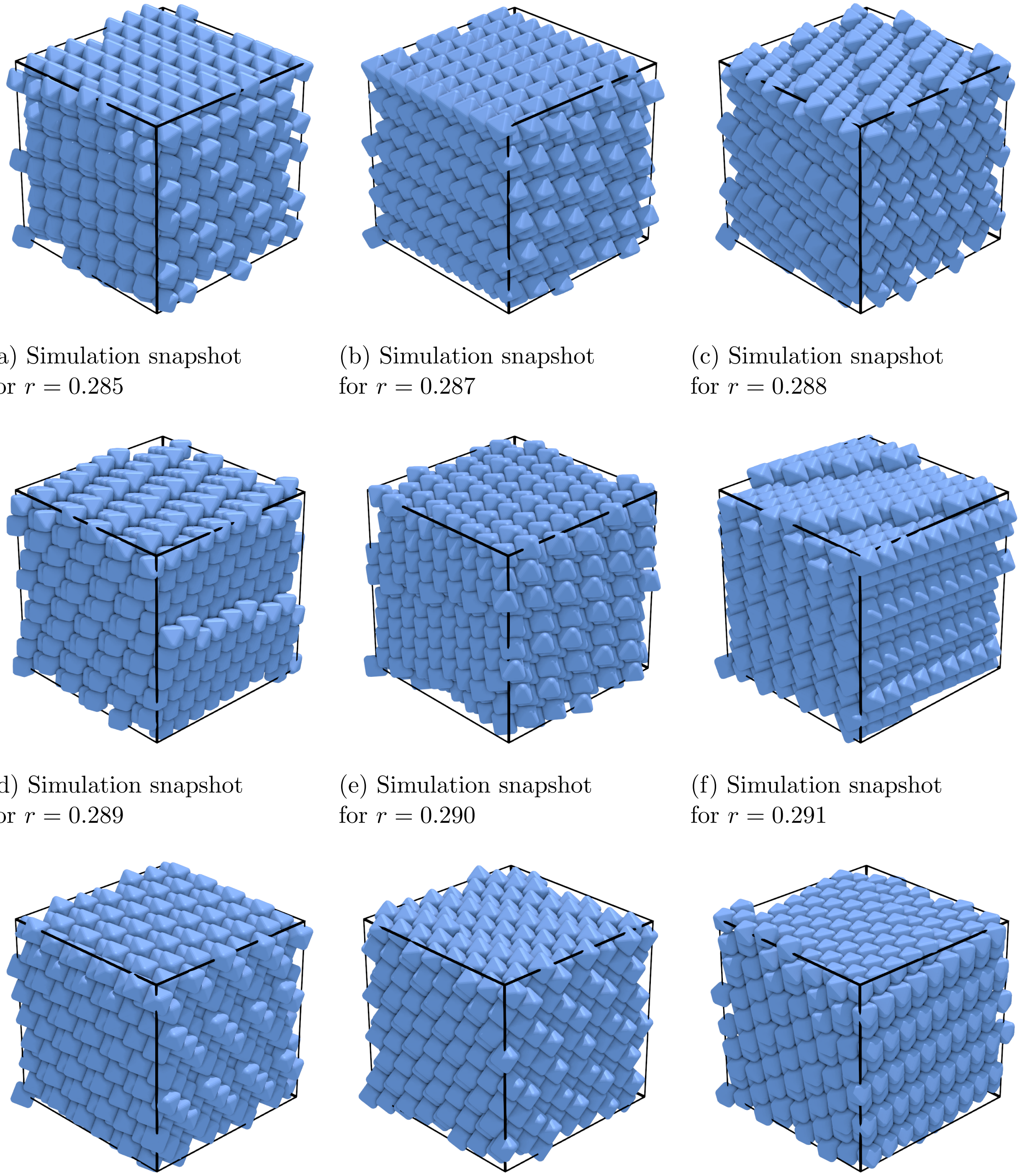


(a) Sum of matrix **A** as a function of the $\log_{10}$ of ε for rounded octahedra.

(b) Sorted eigenvalues for matrix **M** with $\varepsilon = 0.8$ in data set for rounded octahedra.

(c) Eigenvectors $\psi_k(i)$ of matrix **M** plotted as a function of i with $\varepsilon = 0.8$ in data set for rounded octahedra.

Figure 11: Results from diffusion maps for rounded octahedra.



(a) Simulation snapshot for $r = 0.285$

(b) Simulation snapshot for $r = 0.287$

(c) Simulation snapshot for $r = 0.288$

(d) Simulation snapshot for $r = 0.289$

(e) Simulation snapshot for $r = 0.290$

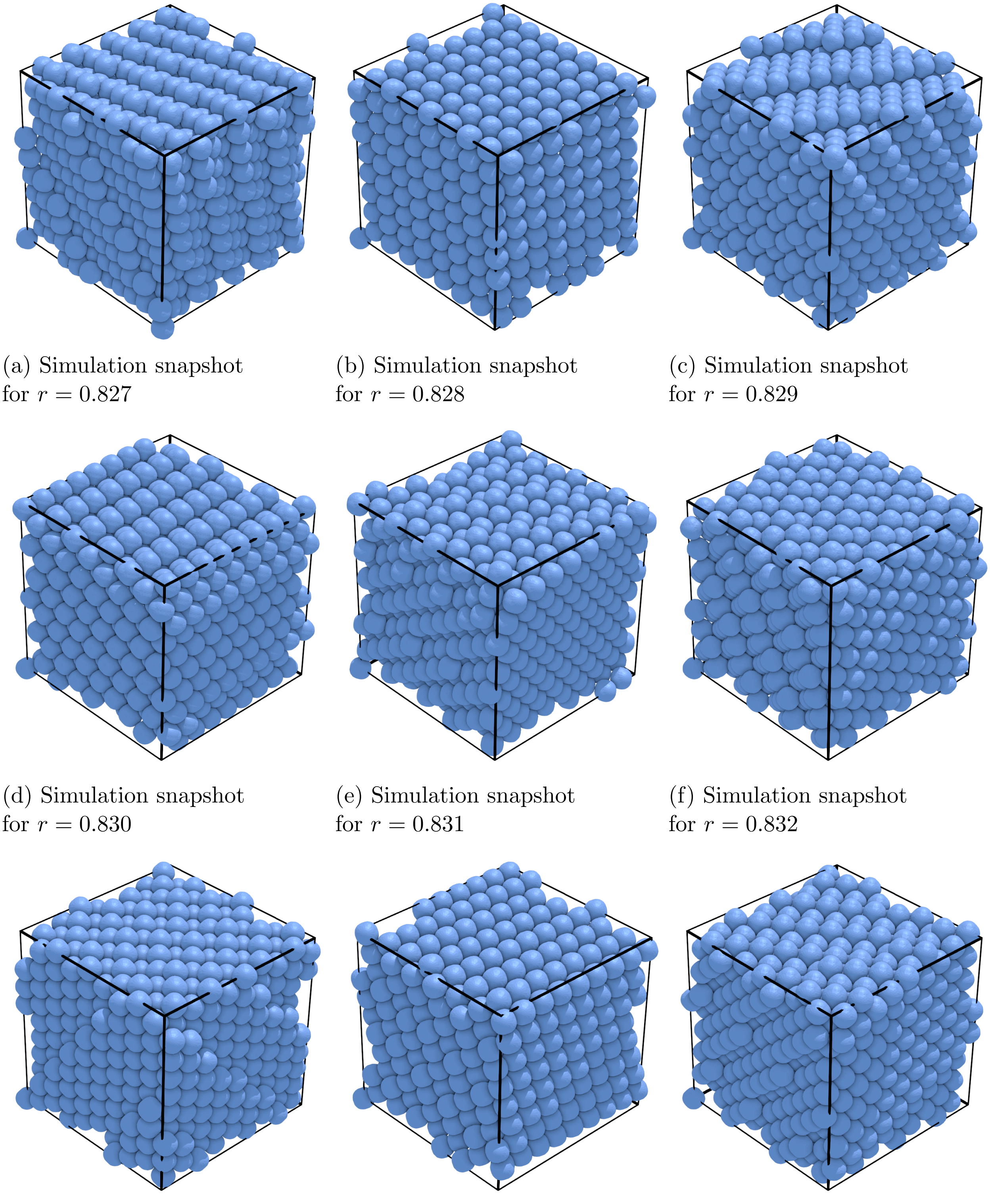
(f) Simulation snapshot for $r = 0.291$

(g) Simulation snapshot for $r = 0.292$

(h) Simulation snapshot for $r = 0.293$

(i) Simulation snapshot for $r = 0.294$

Figure 12: A few snapshots from the simulation of rounded octahedra from $r = 0.285$ to $r = 0.294$.



(a) Simulation snapshot for $r = 0.827$

(b) Simulation snapshot for $r = 0.828$

(c) Simulation snapshot for $r = 0.829$

(d) Simulation snapshot for $r = 0.830$

(e) Simulation snapshot for $r = 0.831$

(f) Simulation snapshot for $r = 0.832$

(g) Simulation snapshot for $r = 0.833$

(h) Simulation snapshot for $r = 0.834$

(i) Simulation snapshot for $r = 0.835$

Figure 13: A few snapshots from the simulation of rounded octahedra from $r = 0.827$ to $r = 0.835$.

4.3 Tetrahedra

In figure 14 the BOP's $\bar{q}_1$ to $\bar{q}_{12}$ are plotted as a function of roundedness for tetrahedra with unit cell $N = 2$. The plot is a lot less ordered then the BOP plots for rounded cubes and rounded octahedra. As opposed to the BOP's for the rounded cubes and octahedra, the order of magnitude of all BOP's are around 10^{-1} except for $\bar{q}_2$ which is at most in order of magnitude 10^{-2} and $\bar{q}_1$ which is at most in order of magnitude 10^{-8}. In other words, almost all of the BOP's actually show curves which makes the plot a lot less ordered. Furthermore, what we have seen in figure 9 with the BOP's for the octahedra, where for some region of r a degeneracy in BOP's is found, also seems to occur for the tetrahedra. For some regions of r, even three different trends can be identified! Figure 14 shows that the system of rounded tetrahedra is much more complicated than the previously studied systems, it it hard to tell where exactly phase transitions are expected and where not since jumps are seen all over the place. Maybe some dimensionality reduction might come in handy. Why this system also yields uneven BOP's has most likely to do with symmetry properties of the spherical harmonics and the differences in symmetry between the structures in systems with tetrahedra and with cubes/octahedra. More study is needed for better understanding. Unfortunately this is not considered in this research.

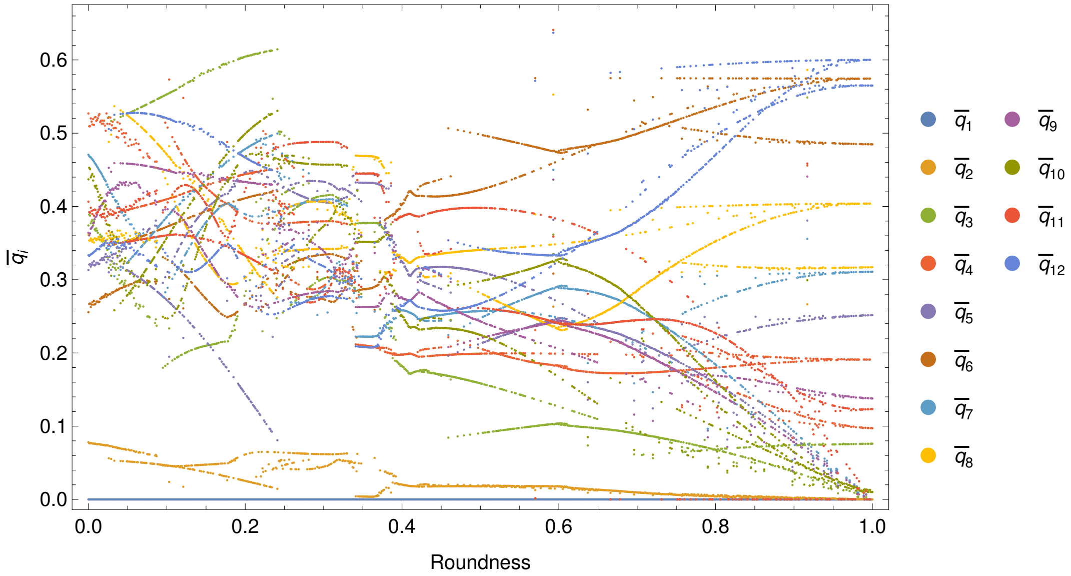


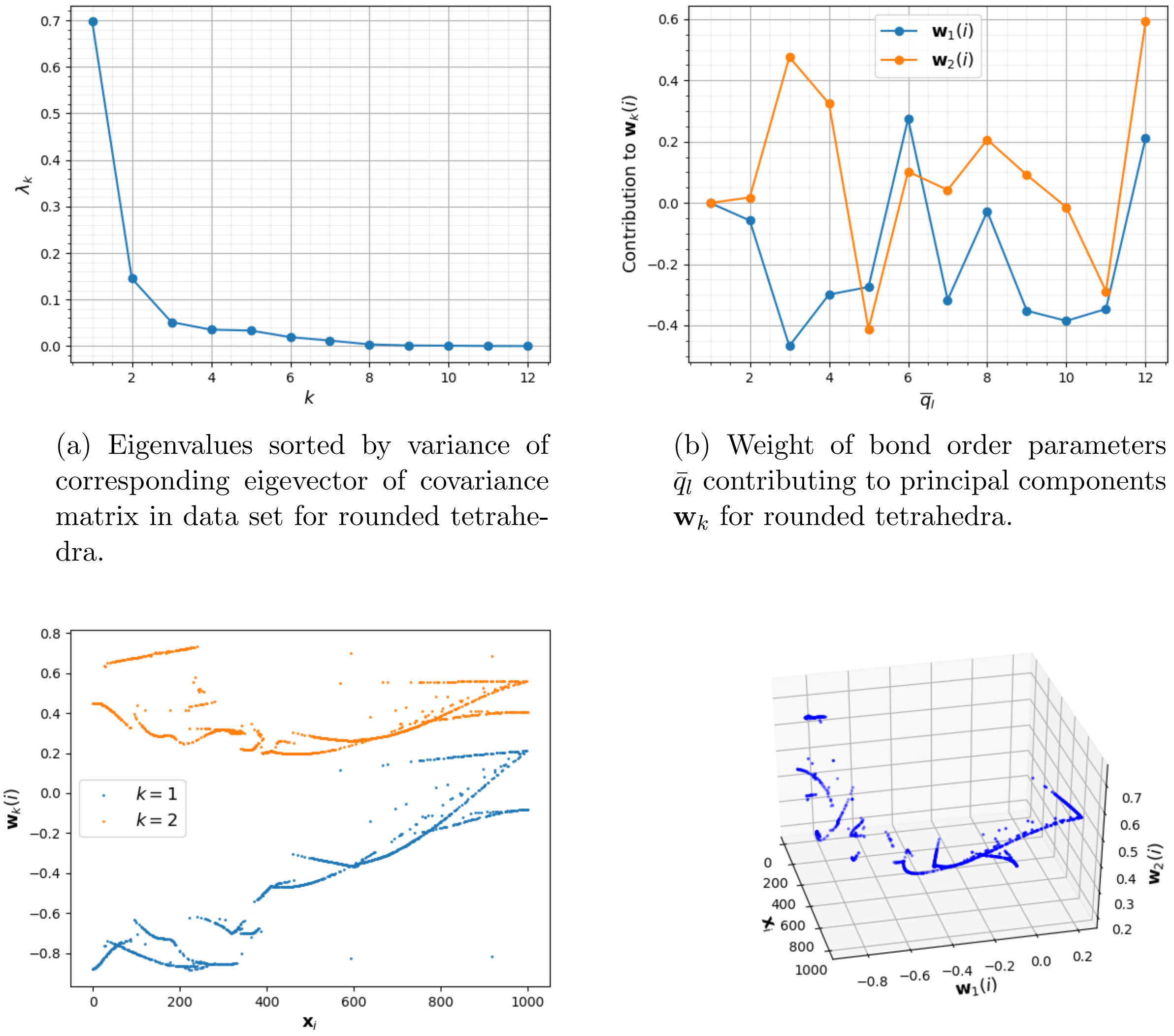
Figure 14: Bond order parameters q_1 to q_{12} plotted as a function of the roundedness for rounded tetrahedra with unit cell $N = 2$.

4.3.1 PCA

Figure 15 shows the PCA results for the rounded tetrahedra system. To measure how PCA reduces the dimensionality for this system, the 12 normalized eigenvalues are plotted in non-ascending order in figure 15a. There is again a clear gap between the first and second eigenvalue, suggesting that most of the “information” from the data set can be expressed with the first eigenvector $\mathbf{w}_1$. The second eigenvalue is smaller and the $3rd$ to the $7th$ are all really small but visibly not zero. We see that even for an evidently more complex system the rounded cubes or octahedra, the dimensionality in such data sets can still be reduced significantly with PCA, even though we see it is more complex by the fact that the eigenvalues are non zero up to the $7th$ eigenvalue.

Figure 15b shows how the first two eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are constructed from the BOP’s. Notice that the uneven BOP’s contribute no less than the uneven BOP’s to the eigenvectors found by PCA, in contrary to the cubes and octahedra (in figures 6b and 10b) where the contribution from the uneven BOP’s is zero. This is perfectly in line with what we see in the BOP plot in figure 15 but still remarkable. We see that both $\mathbf{w}_1$ and $\mathbf{w}_2$ are a complex blend of the BOP’s $\bar{q}_2$ to $\bar{q}_{12}$.

In figure 10c, the eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are plotted as a function of $\mathbf{x}_i$ (roundedness r). Unlike with the rounded cubes and octahedra we cannot clearly identify any of curves the BOP’s make in the plots for $\mathbf{w}_1$ and $\mathbf{w}_2$. Which makes perfect sense since $\mathbf{w}_1$ and $\mathbf{w}_2$ are a complex blend of all of the BOP’s as we have seen in figure 15b (also because the plots in figure 14 are very messy oof course). There are however some regions that have similarities to the BOP plots. For instance, the three curves we see in both $\mathbf{w}_1$ and $\mathbf{w}_2$ in the last section for roughly $r > 0.8$ ($\mathbf{x}_i$ with $i > 800$) can also be found in figure 14 for several BOP’s (for $\bar{q}_3$, $\bar{q}_5$, $\bar{q}_8$ and $\bar{q}_{12}$). We can already identify more clearly jumps in the roundedness spectrum than with the BOP plots only. However, it is still not sufficiently clear to cluster the lines/regions by hand. It might be useful to try some clustering algorithms to find if it can identify the clusters in the plot. Further research for this is needed.



(a) Eigenvalues sorted by variance of corresponding eigevector of covariance matrix in data set for rounded tetrahedra.

(b) Weight of bond order parameters $\bar{q}_l$ contributing to principal components $\mathbf{w}_k$ for rounded tetrahedra.

(c) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for rounded tetrahedra.

(d) 3D plot of eigenvectors of matrix $\mathbf{w}_1$ and $\mathbf{w}_2$ found with PCA plotted as a function of $\mathbf{x}i$ with for rounded tetrahedra.

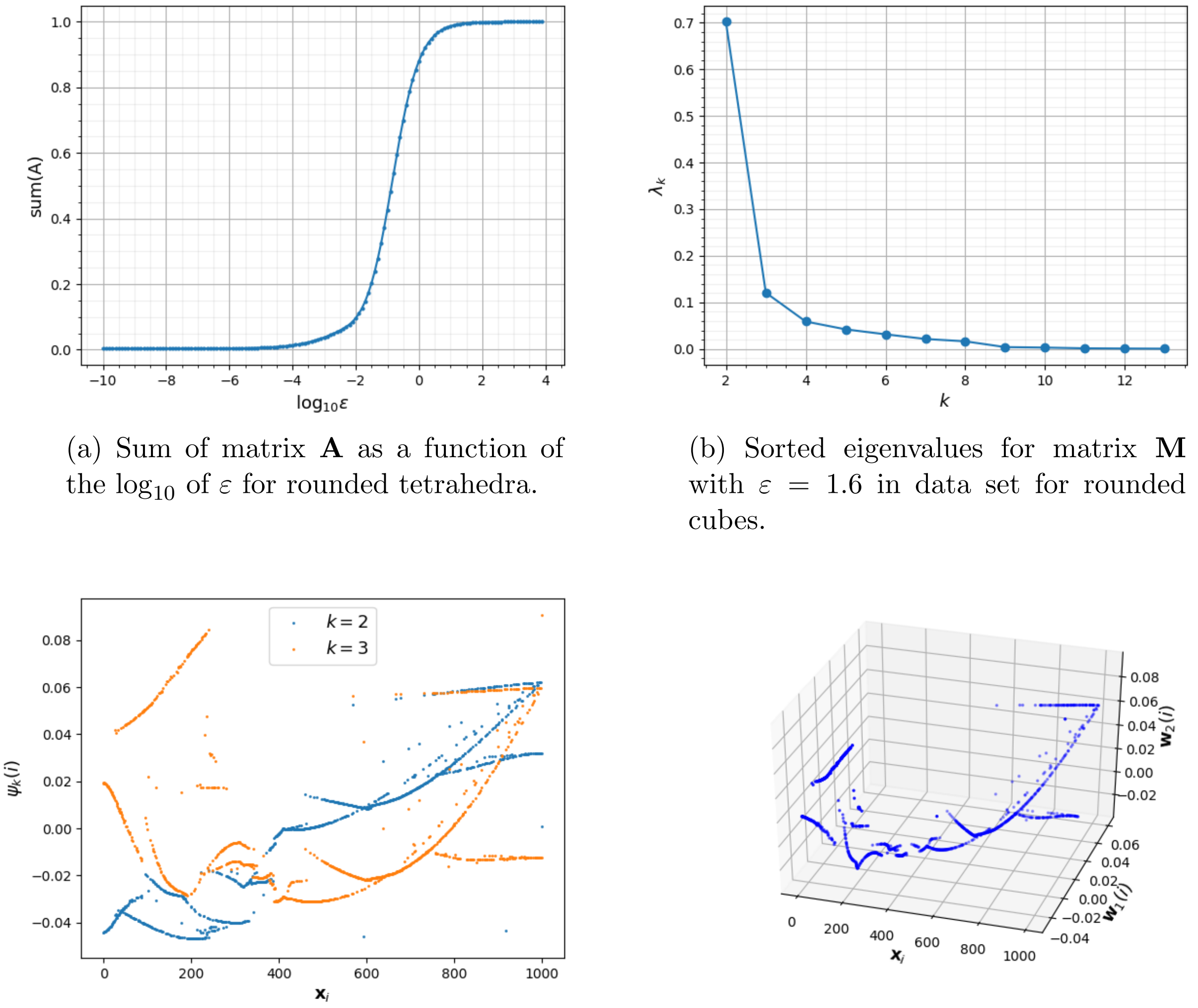
Figure 15: Results from PCA for rounded tetrahedra with unit cell $N = 2$.

4.3.2 dMaps

Figure 16 shows the results obtained from diffusion maps on the rounded tetrahedra system. We find the appropriate value by taking the sum of matrix $\mathbf{A}$ as a function of the $\log_{10}$ of ε (see figure 11a). A value in the upper part of the slope is chosen, i.e. $\varepsilon = 1.6$.

Figure 16b shows the first 12 out of 1001 (normalized) non-trivial eigenvalues of matrix $\mathbf{M}$, which are sorted and plotted for the measure of dimensionality reduction for this system. We see that the first non-trivial eigenvalue is around 0.7, the second eigenvalue drops to almost 0.1 and the from the $3rd$ to the $7th$, the values are small but non zero. From this we can say that most of the data can be expressed with the first non-trivial eigenvector ψ_2, eigenvector ψ_3 plays a small role and ψ_k with $k > 3$ can effectively be neglected. The eigenvalue plot again looks very similar to the eigenvalue plot for PCA. More on this will be discussed in section 4.4.

The eigenvectors ψ_2 and ψ_3 are plotted as a $2D$ plot in figure 16c and as a $3D$ plot in figure 16d as a function of $\mathbf{x}_i$. It is hard to say if we can identify any of the bond order parameter plots from figure 14. One could argue however about similarities with the eigenvector plots from PCA in figure 15c. These plots gives a much better indication on where phase transitions can be found than the BOP plots (figure 14) and also shows more clearly where to find jumps than the eigenvector plot from PCA. Counting by hand results in 8 to 14 different phases depending on how a cluster is defined. A clustering algorithm would still be a good solution for this problem.



(a) Sum of matrix **A** as a function of the $\log_{10}$ of ε for rounded tetrahedra.

(b) Sorted eigenvalues for matrix **M** with $\varepsilon = 1.6$ in data set for rounded cubes.

(c) Eigenvectors $\psi_k(i)$ of matrix **M** plotted as a function of i with $\varepsilon = 1.6$ in data set for rounded tetrahedra.

(d) 3D plot of eigenvectors of matrix **M** $\psi_1(i)$ and $\psi_2(i)$ plotted as a function of $\mathbf{x}i$ with $\varepsilon = 1.6$ in data set for rounded tetrahedra.

Figure 16: Results from diffusion maps for rounded tetrahedra.

4.4 (Non)linear dimensionality reduction

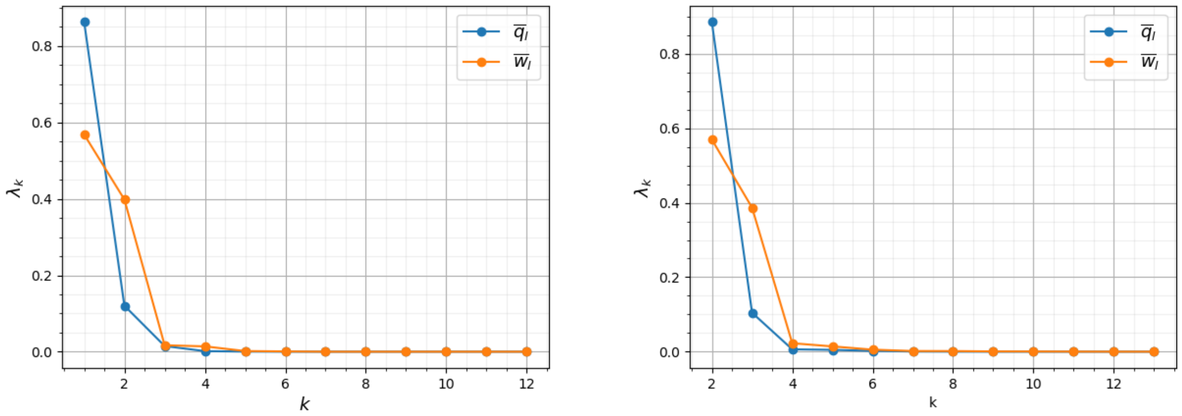
Both dimensionality reduction techniques, principal component analysis and diffusion maps, have now been reviewed for three different systems of rounded polyhedra. To find which method is better suited to analyse these types of problems, it is useful to take a closer look and make a comparison between these different methods.

Firstly, to compare the implementation of the methods, PCA is very fast and straightforward to implement. The covariance matrix can be found directly from the data set, from which we can directly find the eigenvalues and eigenvectors which is where we find the information of our data set. Diffusion maps is not difficult to implement. It is however much slower for a few reasons. First of all, it takes some time to construct matrix $\mathbf{D}$ (a large matrix which contains all distances between the data points). Secondly, an appropriate value of ε has to be found. This is not difficult to do, but it does take some time if ε has to be found by hand. A standardized way to find ε would be helpful and less arbitrary, more research for this is needed. And finally, since matrix $\mathbf{M}$ is so large, it takes some time to find the eigenvalues and eigenvectors from where we extract the information of the data set.

Since PCA is linear and diffusion maps a non-linear dimensionality reduction method, we would expect diffusion maps to give better results (dMaps is more general since it is not restricted to projections on linear planes but can preform projections onto curvilinear-manifolds). In section 4.4 the dimensionaltity reduction plots for PCA and diffusion maps for each system are plotted next to each other for comparison. The plots give the dimensionality reduction for the $\bar{q}_l$ as well as the $\bar{w}_l$ bond order parameters.
For the rounded cubes in the figures 17 we can see that, for the $\bar{q}_l$ BOP's, the first eigenvalue is slightly higher and the second eigenvalue slightly lower in the plot for dimensionality reduction by diffusion maps than by PCA. This indicates a more reduced dimensionality for diffusion maps. For the $\bar{w}_l$ bond order parameters however, the difference is minimal.
For the rounded octahedra in the figures 18 we see that PCA and diffusion maps give effectively the same results for the $\bar{q}_l$ bond order parameters. For the $\bar{q}_l$ BOP's, the first eigenvalue is slighlty higher and second eigenvalue slightly lower for the for dimensionality reduction by diffusion maps than by PCA. This indicates a better reduced dimensionality preformed by diffusion maps on the $\bar{w}_l$ data set. For the rounded tetrahedra in the figures 19 we see that both in $\bar{q}_l$ and $\bar{w}_l$ the dimensionality seem better reduced for diffusion maps than for PCA. Another advantage of PCA is that we can easily find out which bond order parameter contributes to the eigenvectors that are found. This makes PCA a lot more intuative.

Conserning the eigenvector plots for $\mathbf{w}_k$ and ψ_k for PCA and diffusion maps. For the rounded octahedra the plots are effectively the same (except for a scaling factor). For the cubes, diffusion maps does produce a nicer plot (since we can put and ψ_2 in the same figure), but does not help us find phase transition better than PCA (or even the BOP plots from the simulation output). For the tetrahedra, diffusion maps does seem to give slightly better results to identify different structures. It is still hard to count distinct clusters by hand, so a clustering algorithm is still needed to do this properly. It might also be the case that a clustering algorithm can find the different types of structures just fine after performing PCA, which would make diffusion maps redundant. All of this needs more research.

Cubes

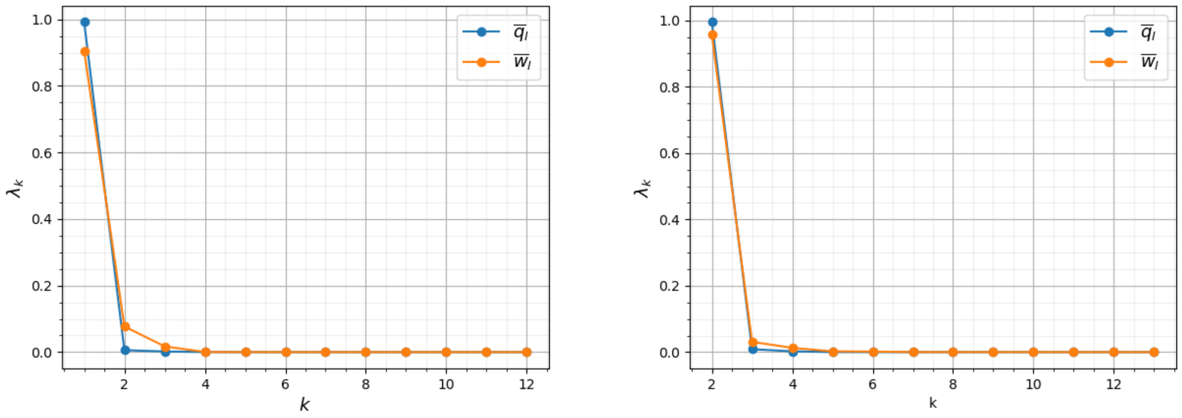


(a) Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameter.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 17: Dimensionality reduction performed with PCA 17a and diffusion maps 17b for the data sets of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ for rounded cubes.

Octahedra

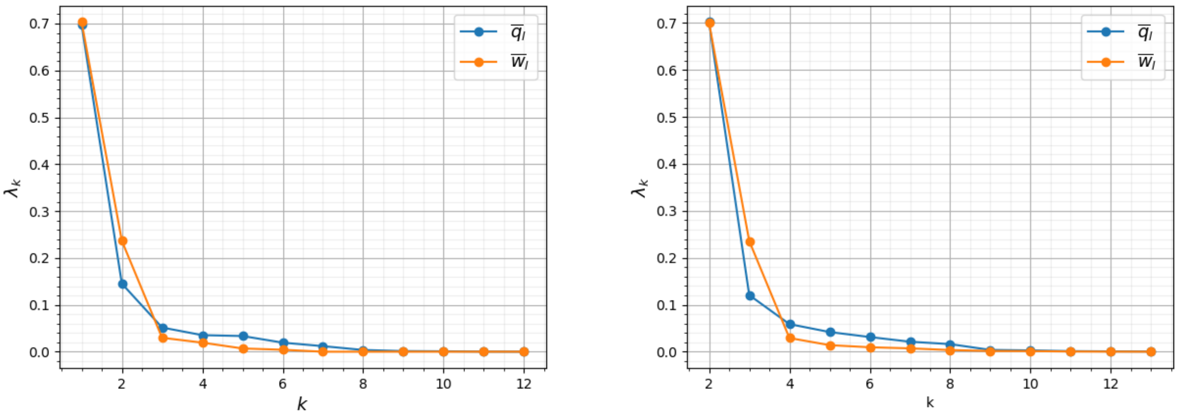


(a) Normalized eigenvalues found with PCA sorted in non-ascending order for the data set with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.8$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 18: Dimensionality reduction performed with PCA 18a and diffusion maps 18b for the data sets of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ for rounded octahedras.

Tetrahedra



(a) Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 1.6$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.5$ for $\bar{w}_l$, sorted in non-ascending order.

Figure 19: Dimensionality reduction performed with PCA 19a and diffusion maps 19b for the data sets of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ for rounded tetrahedra.

4.5 Note

We find a remarkable result that can be seen in section 4.4 where the eigenvalues from the analysis of the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters are plotted. For every studied system (both with PCA and diffusion maps) we see that the second eigenvalue for the $\bar{w}_l$ BOP's is always higher than the second eigenvalue for the $\bar{q}_l$ BOP data sets. Extra figures are added in the appendix where we find the same results. This might be an indication more information can be extracted from analysis on the $\bar{w}_l$ bond order parameters for the study of rounded polyhedra.

5 Conclusion

In this research, we've looked at crystal structures in systems of increasingly rounded cubes, octahedra and tetrahedra by means of machine learning techniques. This has been done by analysing the bond order parameters of these systems, obtained with FBMC simulations with the dimensionality reduction techniques principal component analysis and diffusion maps.

For the system of rounded cubes we find that both PCA and diffusion maps work well to find the phase transitions, i.e. to find the crystal structure in the system. Although the differences in the results are small, diffusion maps does reduce dimensionality better than PCA. However, we already knew from the jump in the bond order parameters that we should find a transition at this point.

The same goes for the system of rounded octahedra, both PCA and diffusion maps give results from which we can determine the different crystal structures. However this was already clear from the gaps in the bond order parameters. For this system there is no advantage in choosing diffusion maps over PCA.

The system of rounded tetrahedra is much more complicated. Whereas the cubes and octahedra only need 6 BOP's to quantify the crystal structures in the systems, the system of rounded tetrahedra need 10 BOP's to quantify its crystal structures. Both PCA and diffusion maps do reduce the complexity the system significantly to a point where the crystal structures can be rougly identified. Diffusion maps does this slightly better than PCA, but it is still difficult to identify all structures by hand. A good extension of this research would be to add a clustering algorithm to complete this phase identification.

Diffusion maps reduces the dimensionality of the data sets slightly better for each studied system. However, improvement in crystal structure identification is small. Also, PCA is much faster and more intuitive than diffusion maps which makes PCA the preffered technique for this specific problem. It might however be useful for more complex systems to try diffusion maps anyways.

References

[1] Amir Haji-Akbari et al. “Disordered, quasicrystalline and crystalline phases of densely packed tetrahedra”. In: *Nature* 462 (Dec. 2009), 773 EP -. URL: https://doi.org/10.1038/nature08641.

[2] S Torquato and Yang Jiao. “Dense Packings of the Platonic and Archimedean Solids”. In: *Nature* 463 (Feb. 2010), p. 1106. DOI: 10.1038/nature08847.

[3] Orlin D. Velev and Shalini Gupta. “Materials Fabricated by Micro and Nanoparticle Assembly. The Challenging Path from Science to Engineering”. In: *Advanced Materials* 21 (May 2009). DOI: 10.1002/adma.200801837.

[4] Bartosz A. Grzybowski et al. “Self-assembly: from crystals to cells”. In: *Soft Matter* 5 (6 2009), pp. 1110–1128. DOI: 10.1039/B819321P. URL: http://dx.doi.org/10.1039/B819321P.

[5] Sharon C. Glotzer Michael J. Solomon. “Anisotropy of building blocks and their assembly into complex structures”. In: *Nature Materials* 6 (Aug. 2007).

[6] Shuixiang Li Weiwei Jin Peng Lu. “Evolution of the dense packings of spherotetrahedral particles: from ideal tetrahedra to spheres”. In: *Scientific Reports* (Oct. 2015).

[7] Sharon c. Glotzer Pablo F. Damasceno Michael Engels. “Predictive Self-Assembly of Polyhedra into Complex Structures”. In: *Science* 337 (July 2012).

[8] Frank Smallenburg Laura Filion Emanuele Boattini Michel Ram. “Neural-network-based order parameters for classification of binary hard-sphere crystal structures”. In: *Molecular Physics* (2018).

[9] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. “Bond-orientational order in liquids and glasses”. In: *Phys. Rev. B* 28 (2 June 1983), pp. 784–805. DOI: 10.1103/PhysRevB.28.784. URL: https://link.aps.org/doi/10.1103/PhysRevB.28.784.

[10] Wolfgang Lechner and Christoph Dellago. “Accurate determination of crystal structures based on averaged local bond order parameters”. In: *The Journal of chemical physics* 129 (Oct. 2008), p. 114707. DOI: 10.1063/1.2977970.

[11] Walter Mickel et al. “Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter”. In: *The Journal of Chemical Physics* 138.4 (2013), p. 044501. DOI: 10.1063/1.4774084. eprint: https://doi.org/10.1063/1.4774084. URL: https://doi.org/10.1063/1.4774084.

[12] Andrew L Ferguson. “Machine learning and data science in soft materials engineering”. In: *Journal of Physics: Condensed Matter* 30.4 (Dec. 2017), p. 043002. DOI: 10.1088/1361-648x/aa98bd. URL: https://doi.org/10.1088%2F1361-648x%2Faa98bd.

[13] Joost de Graaf et al. “Crystal-structure prediction via the Floppy-Box Monte Carlo algorithm: Method and application to hard (non)convex particles”. In: *The Journal of Chemical Physics* 137.21 (2012), p. 214101. DOI: 10.1063/1.4767529. eprint: https://doi.org/10.1063/1.4767529. URL: https://doi.org/10.1063/1.4767529.

[14] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. “A fast procedure for computing the distance between complex objects in three-dimensional space”. In: *IEEE Journal on Robotics and Automation* 4.2 (Apr. 1988), pp. 193–203. ISSN: 0882-4967. DOI: 10.1109/56.2083.

A Figures

In this appendix, all the figures used in this research, plus additional figures are reorted. Figures of additional systems that have not been discussed in the research itself are added, as well as the plotted simulation output for the $\bar{q}_l$, $\bar{w}_l$ bond order parameters and packing fractions. Also the plots for the $\bar{w}_l$ bond order parameters output for PCA and diffusion maps are shown.

A.1 Cubes

A.1.1 Simulation output

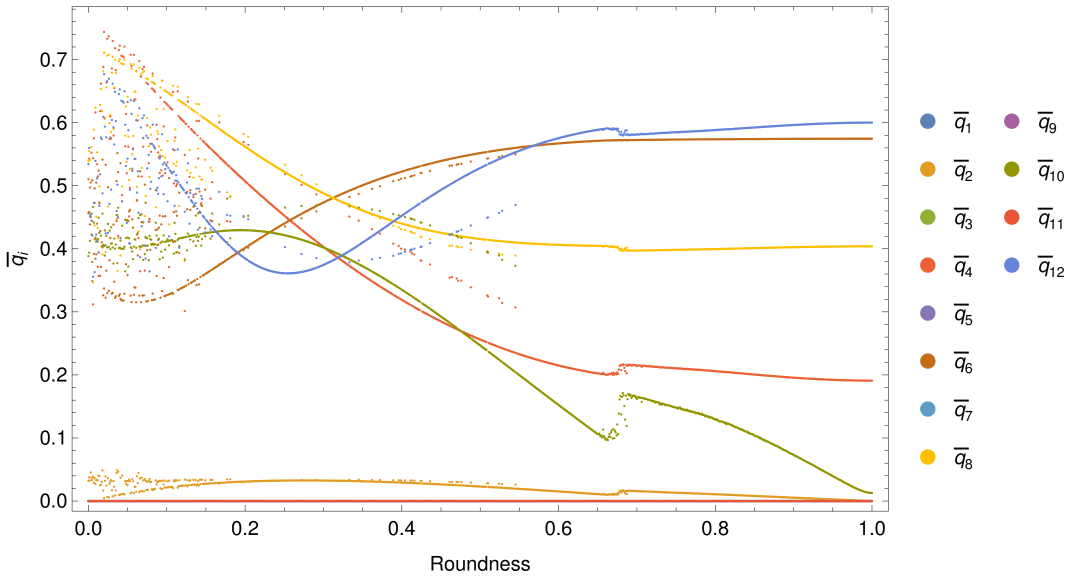


Figure 20: Bond order parameters $\bar{q}_1$ to $\bar{q}_{12}$ as a function of roundedness parameter r for increasingly rounded cubes.

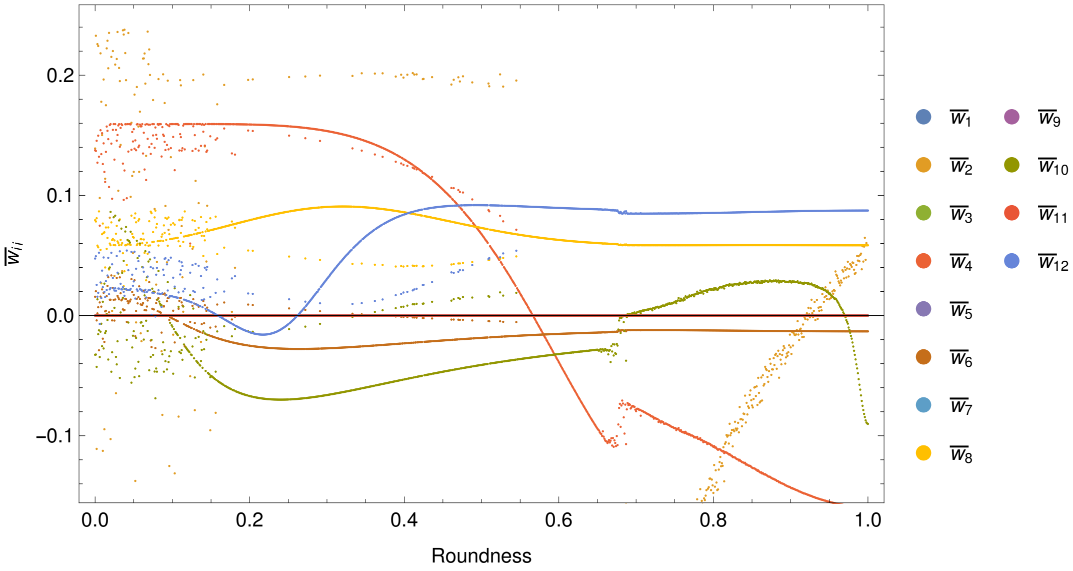


Figure 21: Bond order parameters $\bar{w}_1$ to $\bar{w}_{12}$ as a function of roundedness parameter r for increasingly rounded cubes.

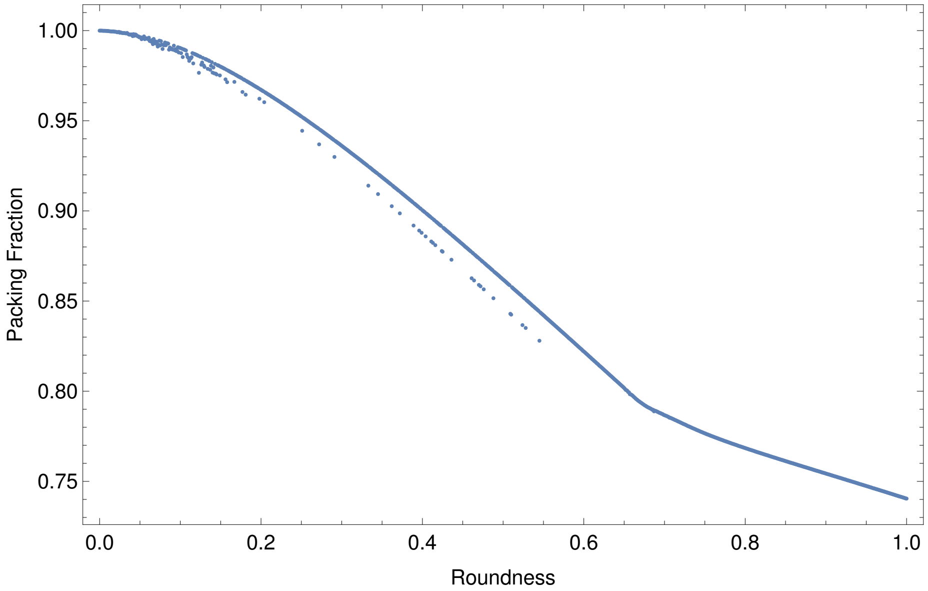


Figure 22: Packing fraction as a function of roundedness parameter r for increasingly rounded cubes.

A.1.2 PCA

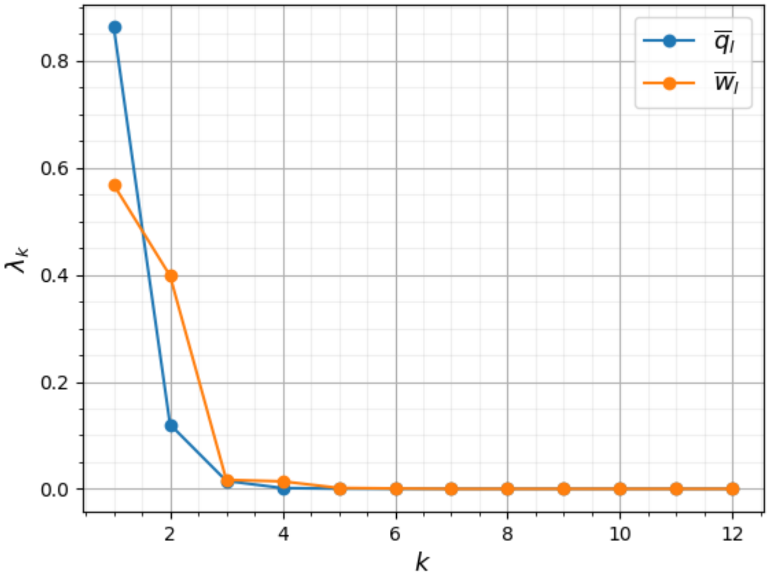
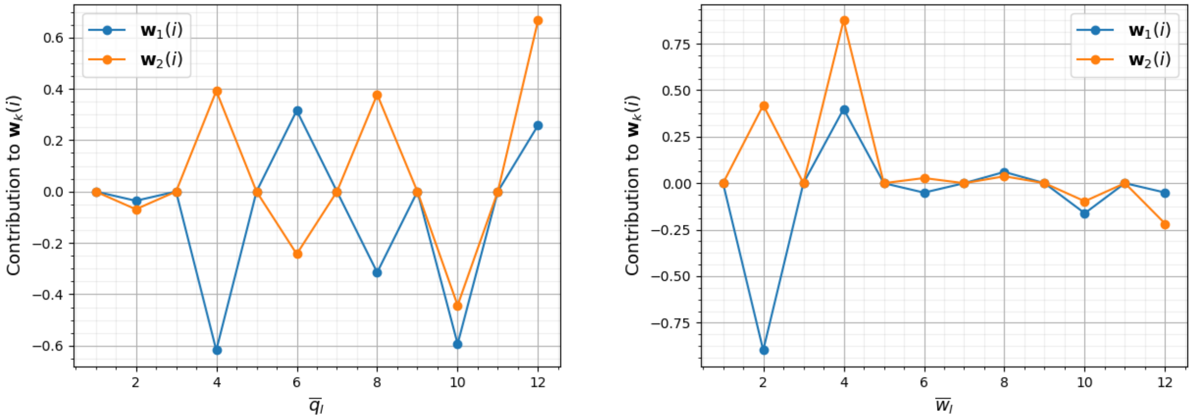


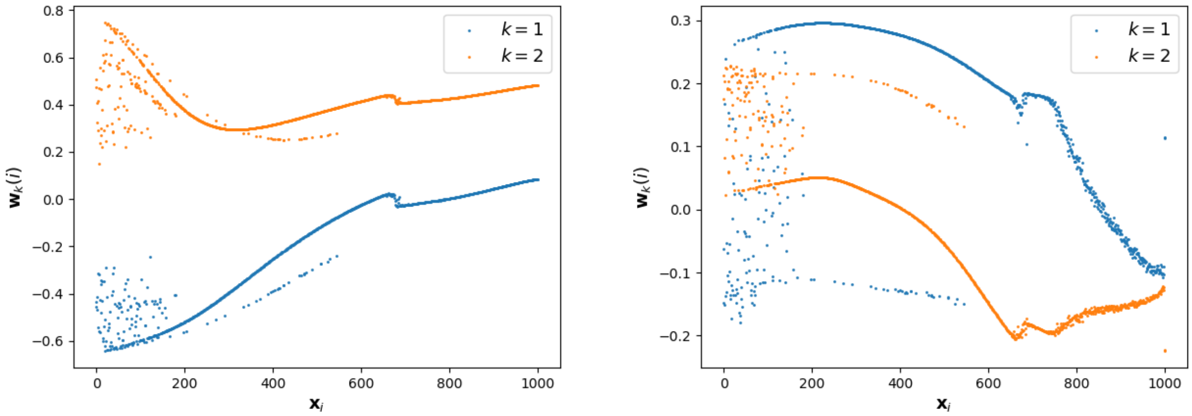
Figure 23: Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters for rounded cubes.



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded cubes.

(b) Weight of bond order parameters $\bar{w}_l$ contributing to principal component for rounded cubes.

Figure 24: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded cubes.

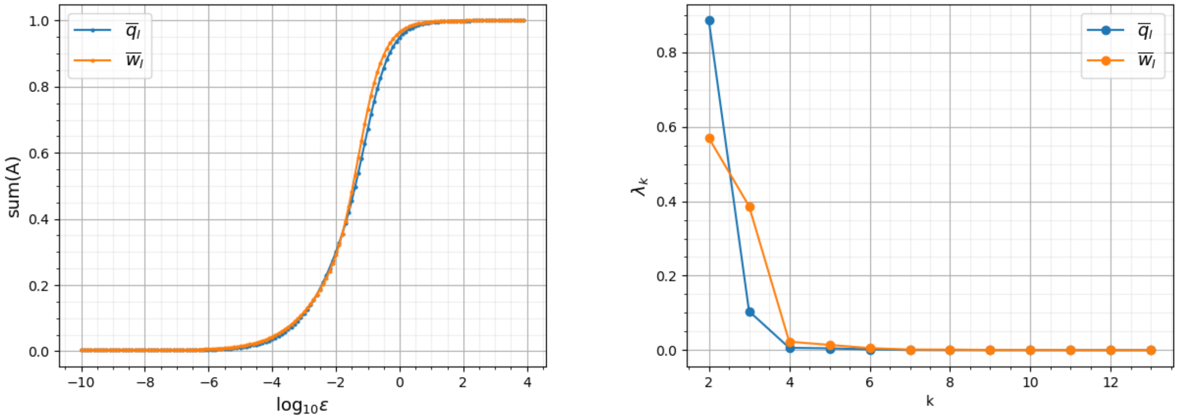


(a) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded cubes.

(b) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded cubes.

Figure 25: Eigenvectors found with PCA plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 25a) and $\bar{w}_l$ (figure 25b).

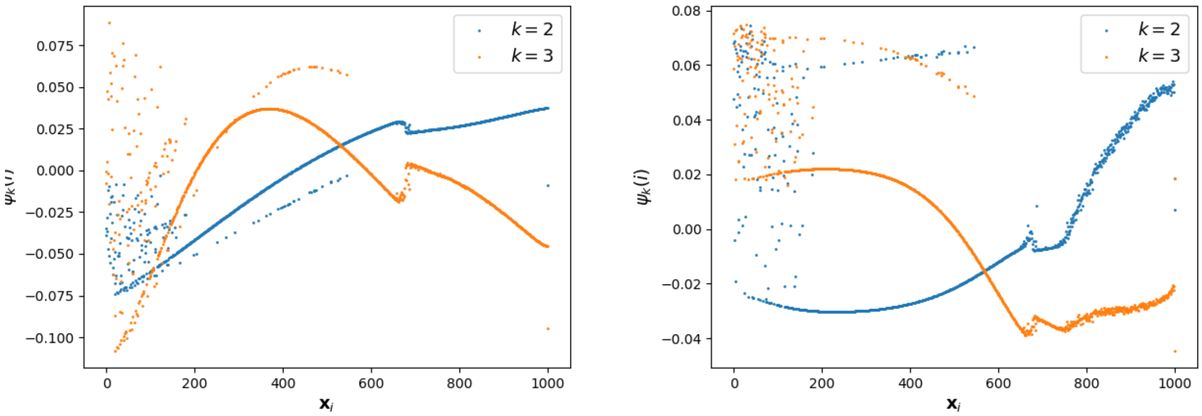
A.1.3 dMaps



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded cubes.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 26: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded cubes.



(a) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded cubes.

(b) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded cubes..

Figure 27: Eigenvectors found by dMaps plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 27a) and $\bar{w}_l$ (figure 27b).

A.2 Octahedra

A.2.1 Simulation output

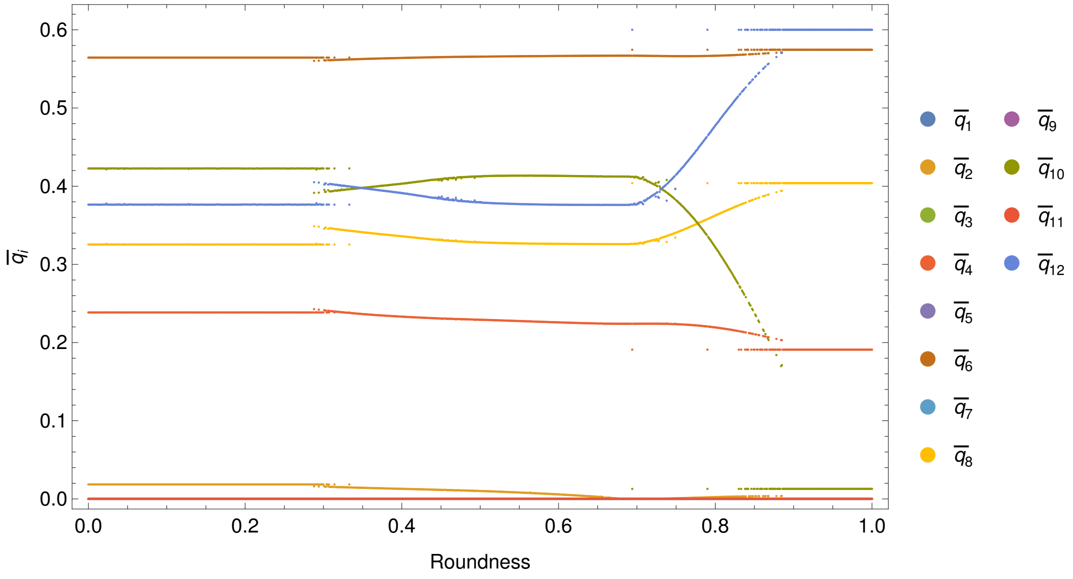


Figure 28: Bond order parameters $\bar{q}_1$ to $\bar{q}_{12}$ as a function of roundedness parameter r for increasingly rounded octahedra.

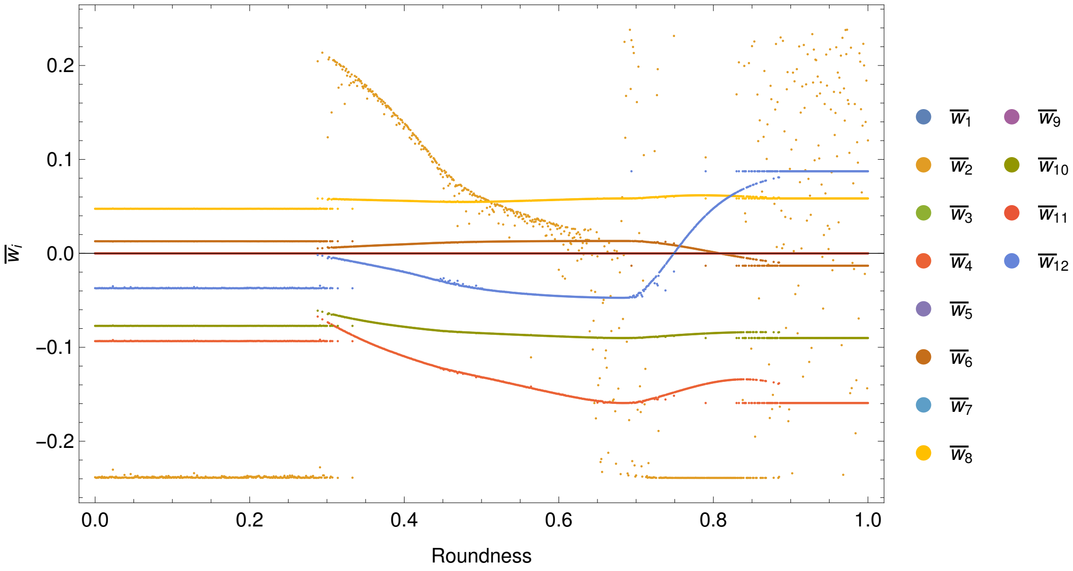


Figure 29: Bond order parameters $\bar{w}_1$ to $\bar{w}_{12}$ as a function of roundedness parameter r for increasingly rounded octahedra.

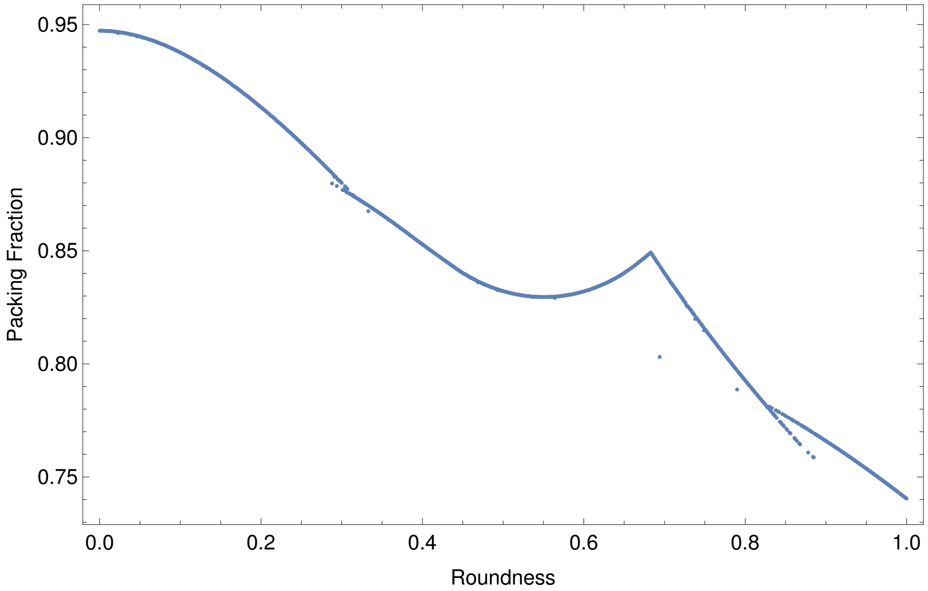


Figure 30: Packing fraction as a function of roundedness parameter r for increasingly rounded octahedra.

A.2.2 PCA

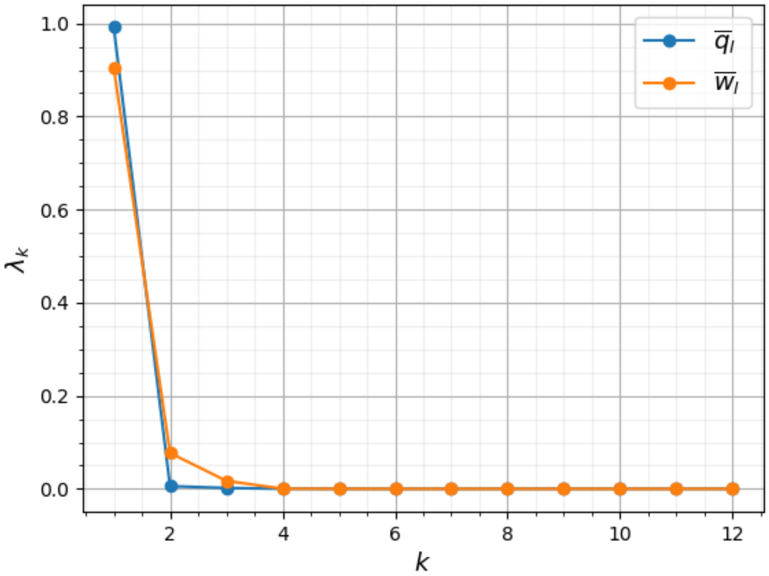
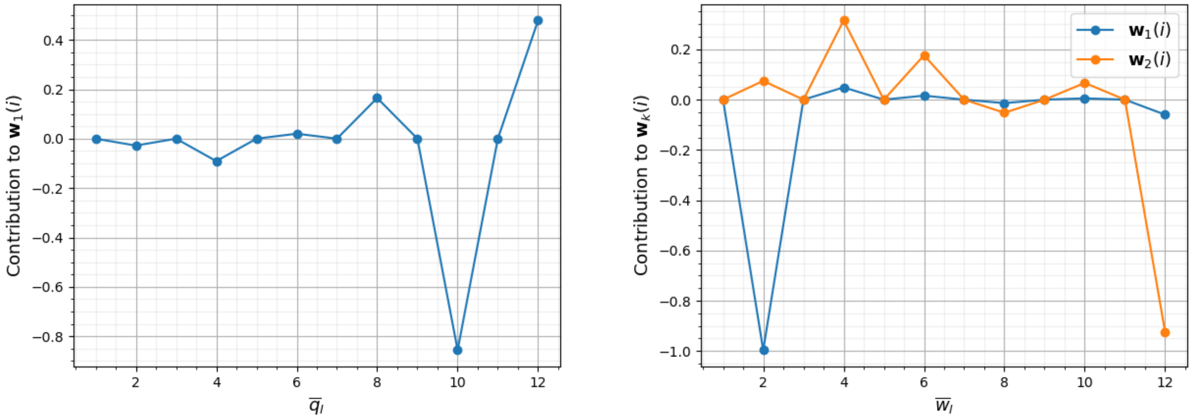


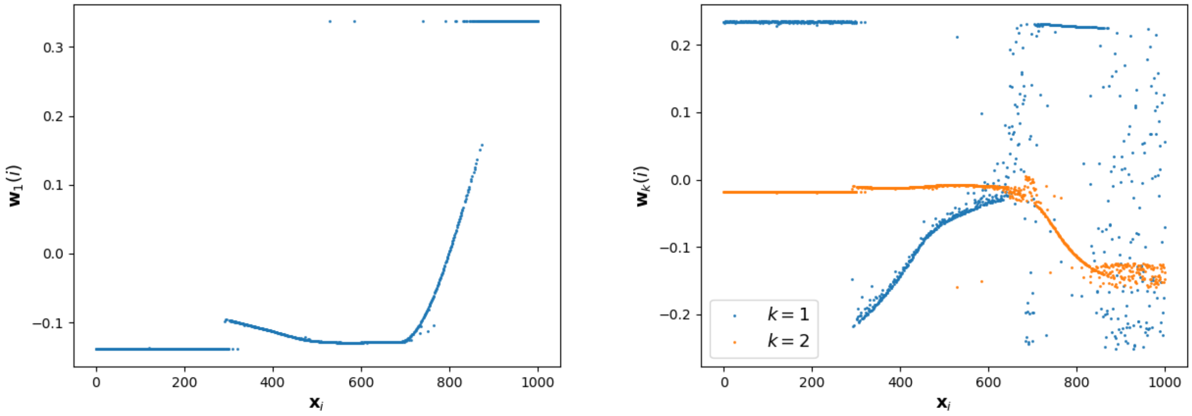
Figure 31: Normalized eigenvalues λ_k found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameter for rounded octahedra.



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded octahedra.

(b) Weight of bond order parameters $\bar{w}_l$ contributing to principal components for rounded octahedra.

Figure 32: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded octahedra.

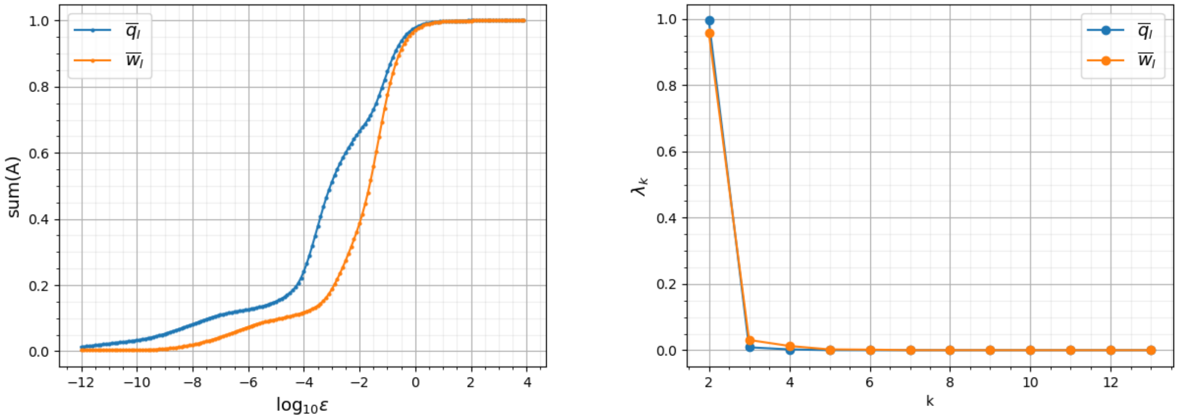


(a) Eigenvector $\mathbf{w}_1$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded octahedra.

(b) Eigenvector $\mathbf{w}_1$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded cubes.

Figure 33: Eigenvectors found with PCA plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 33a) and $\bar{w}_l$ (figure 33b) for rounded octahedra.

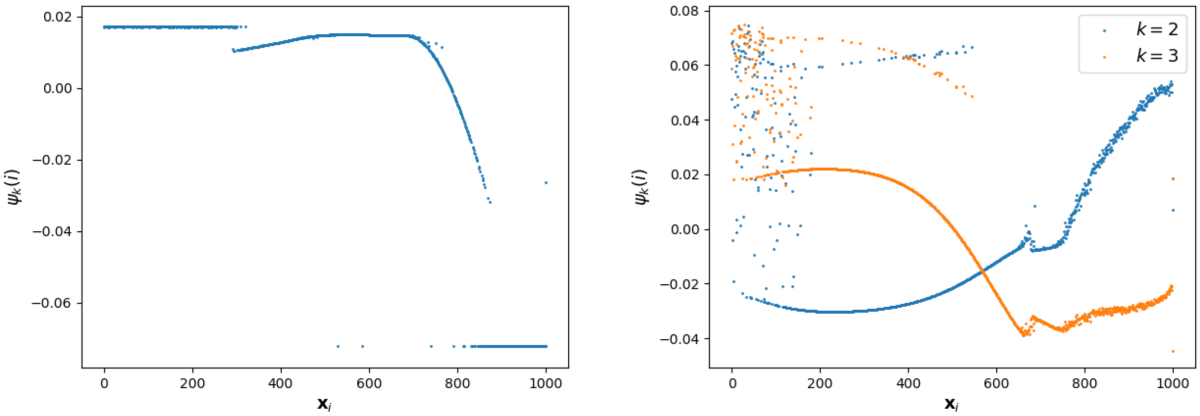
A.2.3 dMaps



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded octahedra.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 34: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded octahedra.



(a) Eigenvector ψ_2 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded octahedra.

(b) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded octahedra.

Figure 35: Eigenvectors found by dMaps plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 35a) and $\bar{w}_l$ (figure 35b).

A.3 Tetrahedra ($N = 2$)

A.3.1 Simulation Output

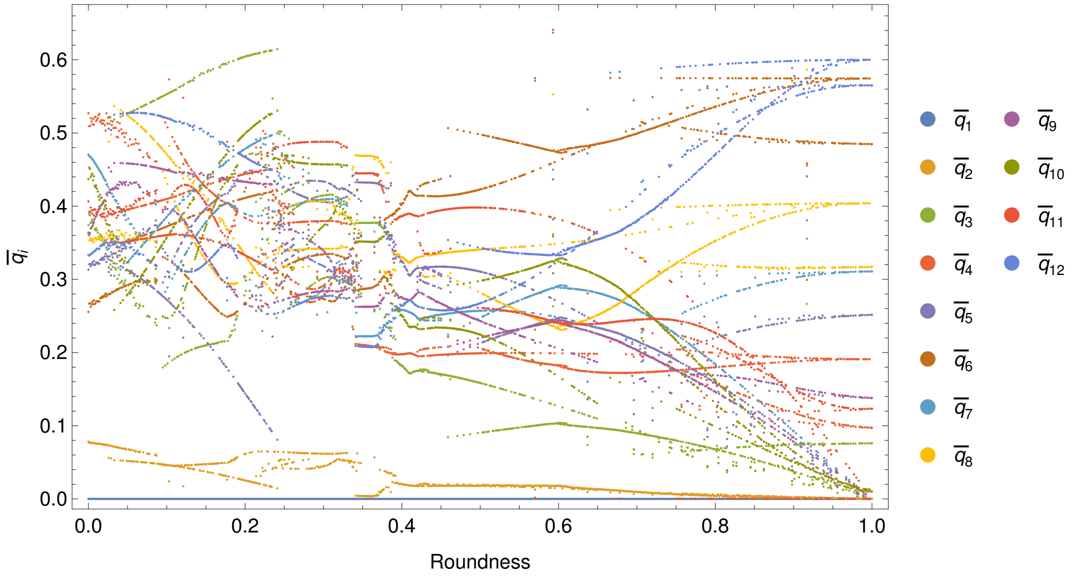


Figure 36: Bond order parameters $\bar{q}_1$ to $\bar{q}_{12}$ as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 2$.

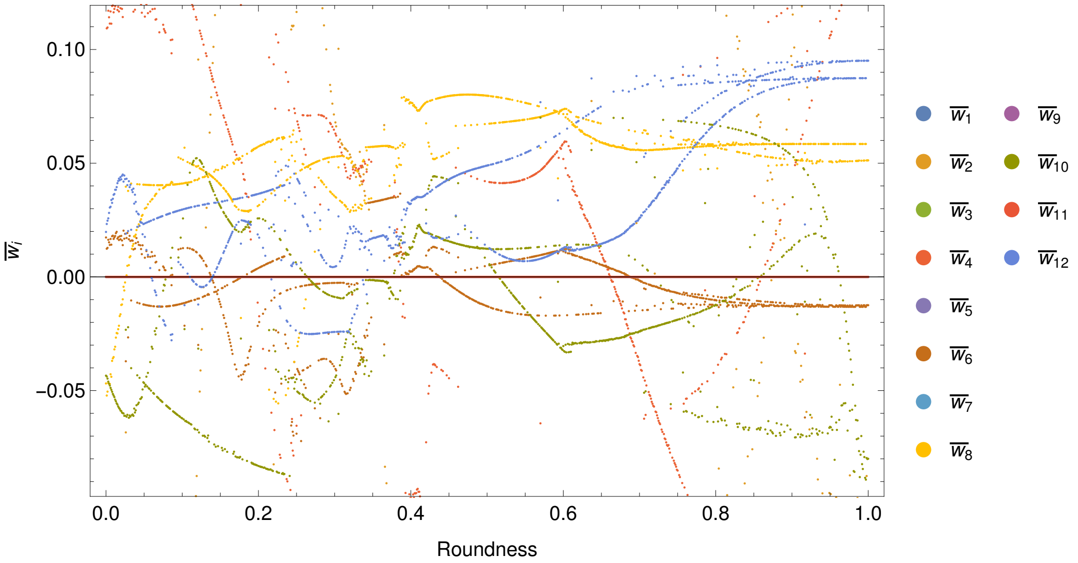


Figure 37: Bond order parameters $\bar{w}_1$ to $\bar{w}_{12}$ as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 2$.

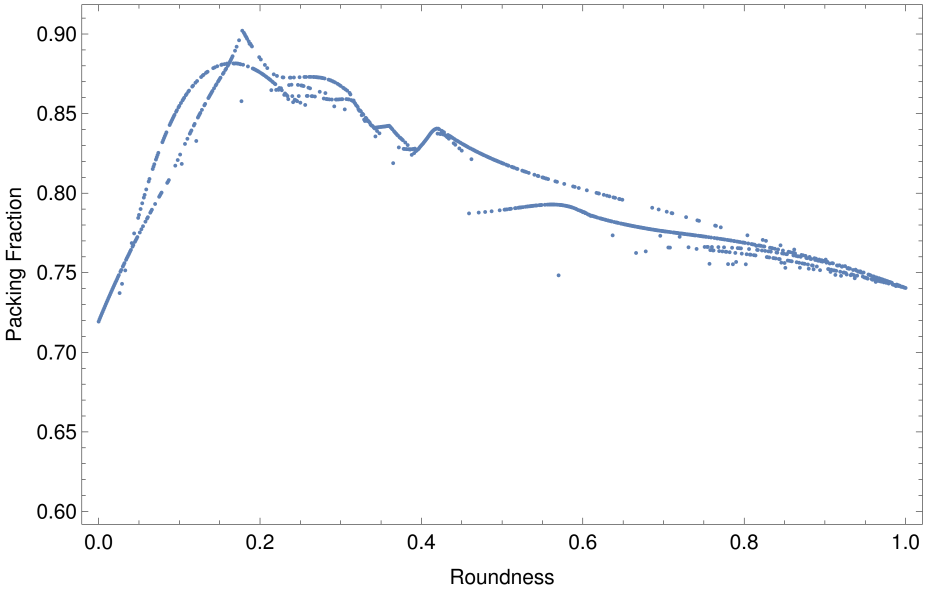


Figure 38: Packing fraction as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 2$.

A.3.2 PCA

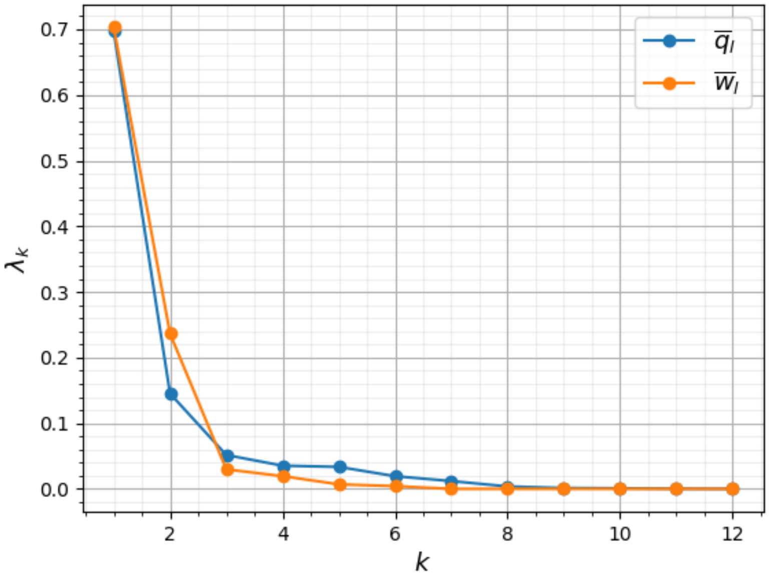
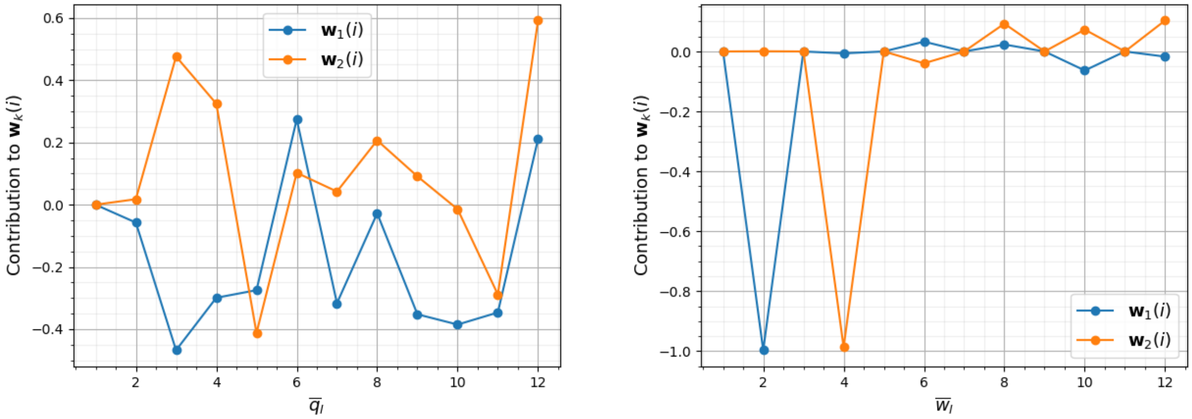


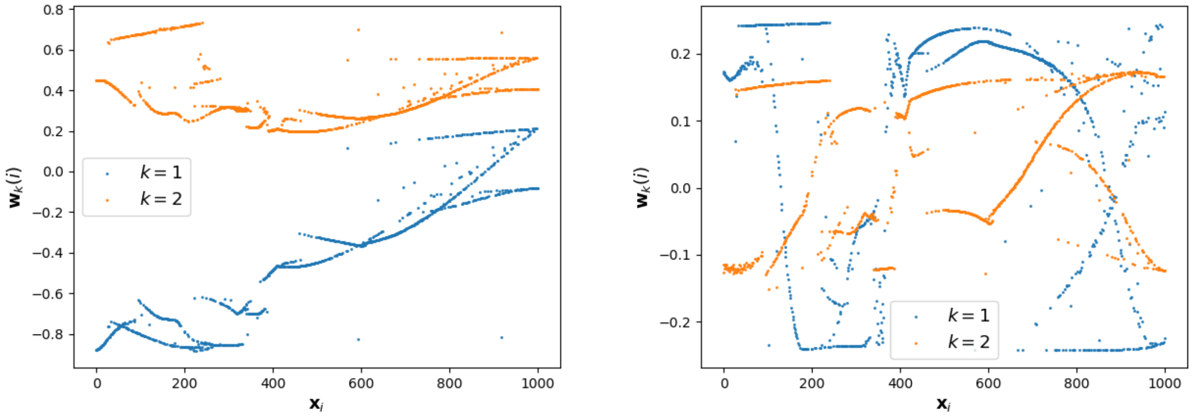
Figure 39: Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters for rounded tetrahedra with unit cell $N = 1$.



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded tetrahedra with unit cell $N = 2$.

(b) Weight of bond order parameters $\bar{w}_l$ contributing to principal component for rounded tetrahedra with unit cell $N = 2$.

Figure 40: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded tetrahedra with unit cell $N = 2$.

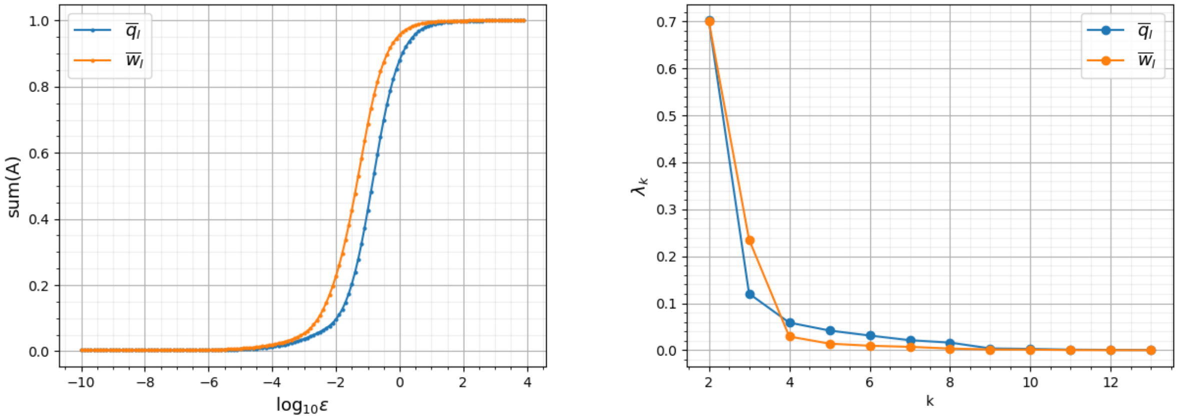


(a) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded tetrahedra with unit cell $N = 2$.

(b) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded tetrahedra with unit cell $N = 2$.

Figure 41: Eigenvectors found with PCA plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 41a) and $\bar{w}_l$ (figure 41b) for rounded tetrahedra with unit cell $N = 2$.

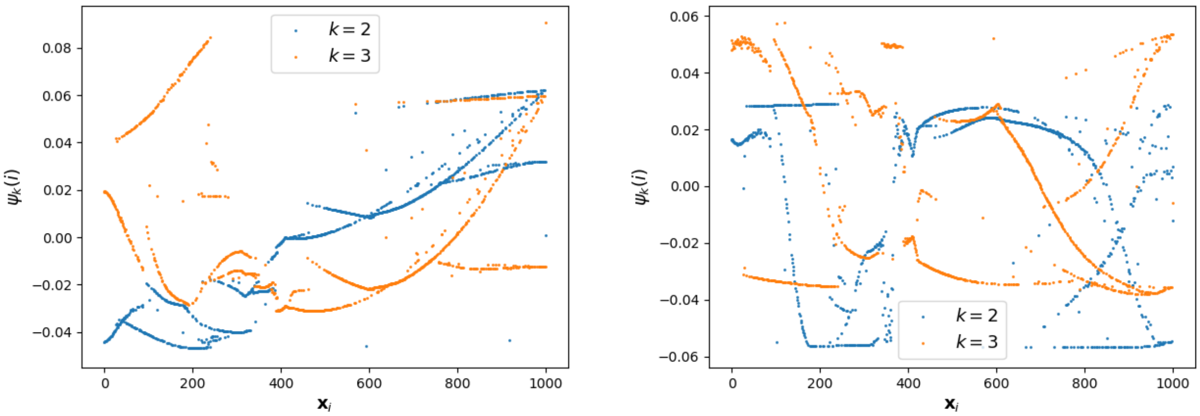
A.3.3 dMaps



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded tetrahedra with unit cell $N = 2$.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 42: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded tetrahedra with unit cell $N = 2$.



(a) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded tetrahedra with unit cell $N = 2$.

(b) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded tetrahedra with unit cell $N = 2$.

Figure 43: Eigenvectors found by dMaps plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 43a) and $\bar{w}_l$ (figure 43b).

A.4 Tetrahedra ($N = 4$)

A.4.1 Simulation Output

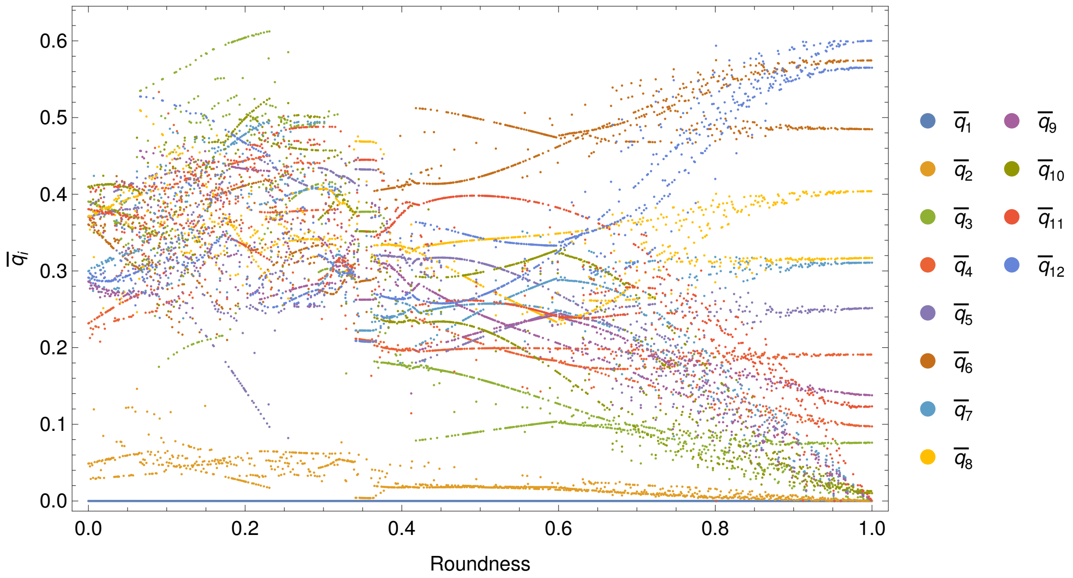


Figure 44: Bond order parameters $\bar{q}_1$ to $\bar{q}_{12}$ as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 4$.

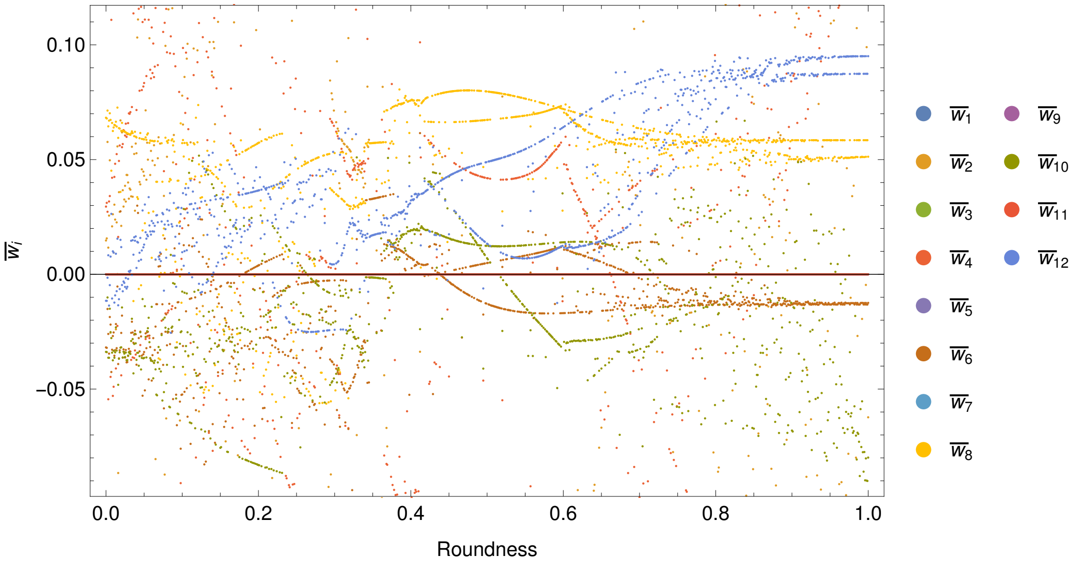


Figure 45: Bond order parameters $\bar{w}_1$ to $\bar{w}_{12}$ as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 4$.

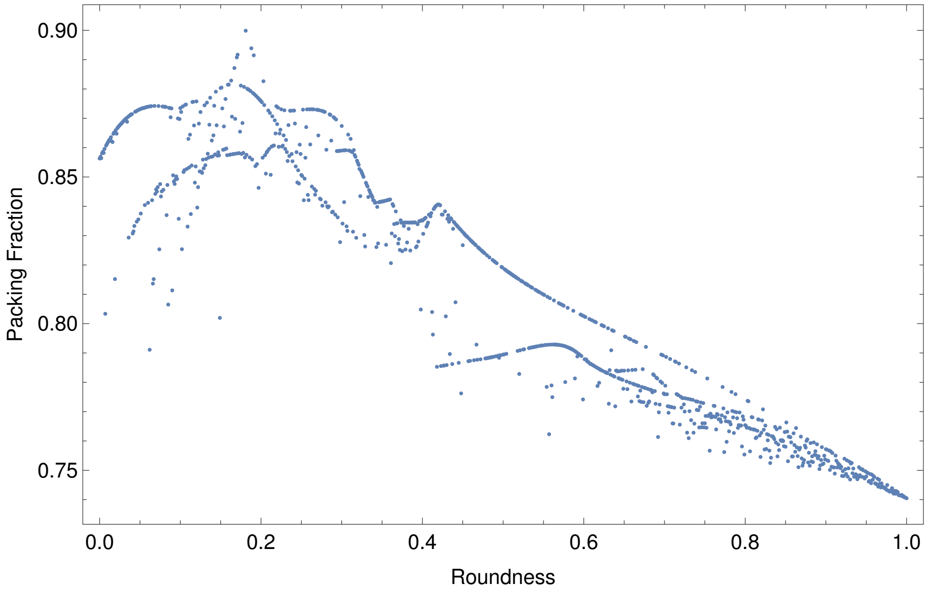


Figure 46: Packing fraction as a function of roundedness parameter r for increasingly rounded tetrahedra with $N = 4$.

A.4.2 PCA

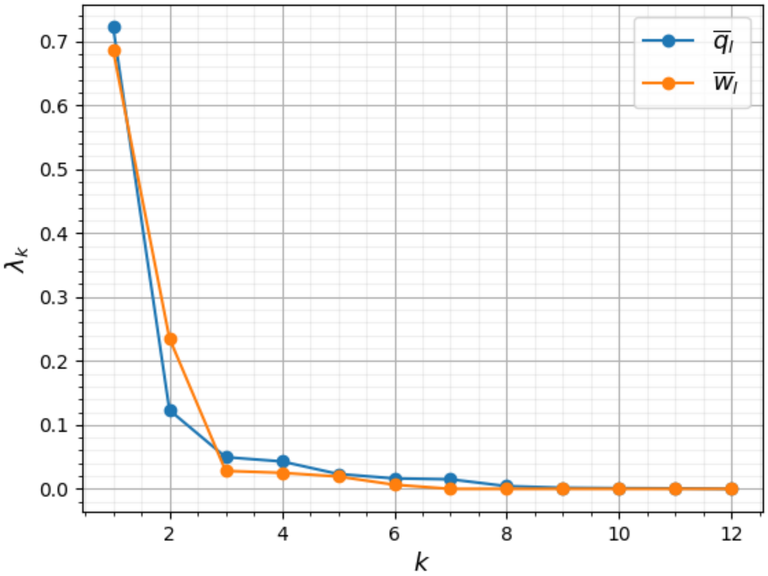
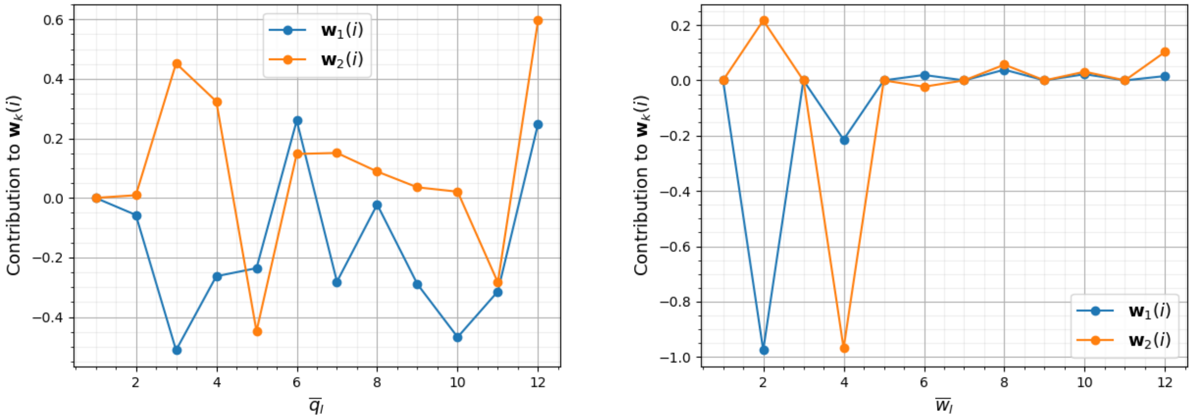


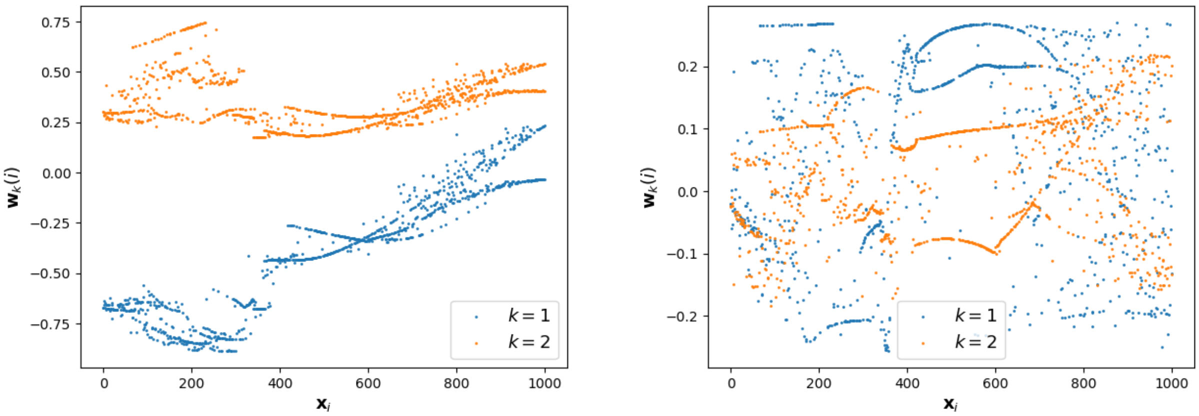
Figure 47: Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters for rounded tetrahedra with unit cell $N = 4$.



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded tetrahedra with unit cell $N = 4$.

(b) Weight of bond order parameters $\bar{w}_l$ contributing to principal component for rounded tetrahedra with unit cell $N = 4$.

Figure 48: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded tetrahedra with unit cell $N = 4$.

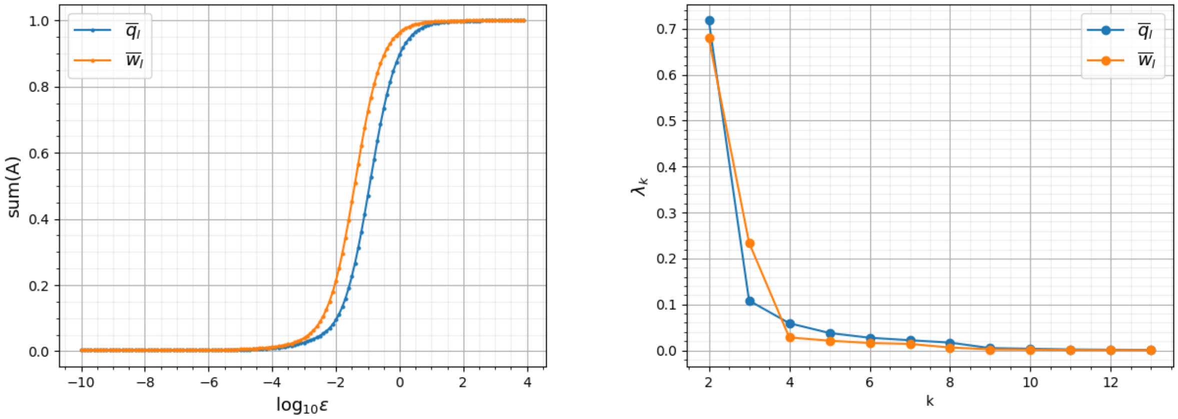


(a) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded tetrahedra with unit cell $N = 4$.

(b) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded tetrahedra with unit cell $N = 4$.

Figure 49: Eigenvectors found with PCA plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 49a) and $\bar{w}_l$ (figure 49b) for rounded tetrahedra with unit cell $N = 4$.

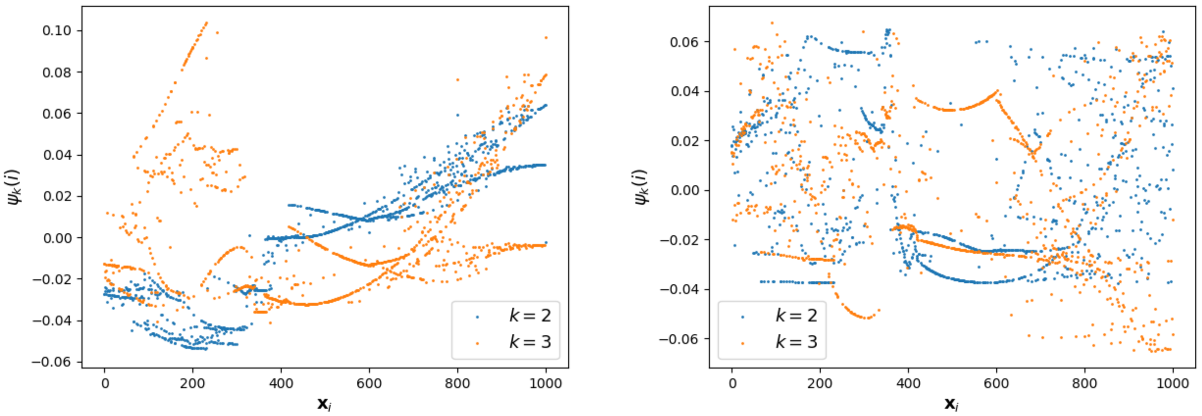
A.4.3 dMaps



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded tetrahedra with unit cell $N = 4$.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 50: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded tetrahedra with unit cell $N = 4$.



(a) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded tetrahedra with unit cell $N = 4$.

(b) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded tetrahedra with unit cell $N = 4$.

Figure 51: Eigenvectors found by dMaps plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 51a) and $\bar{w}_l$ (figure 51b) for increasingly rounded tetrahedra with unit cell $N = 4$.

A.5 Truncated Tetrahedra

A.5.1 Simulation Output

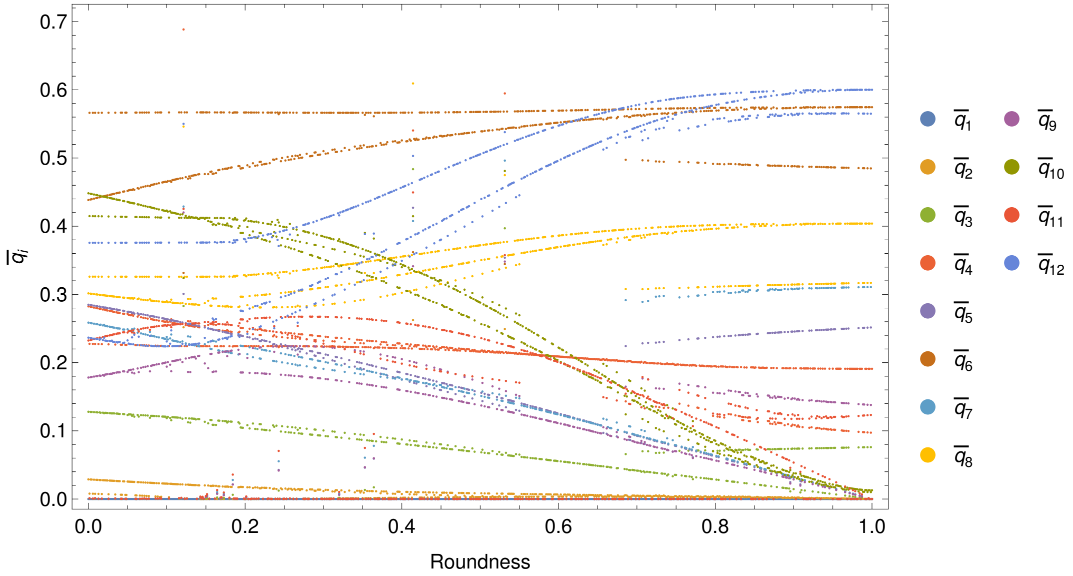


Figure 52: Bond order parameters $\bar{q}_1$ to $\bar{q}_{12}$ as a function of roundedness parameter r for increasingly rounded truncated tetrahedra with $N = 2$.

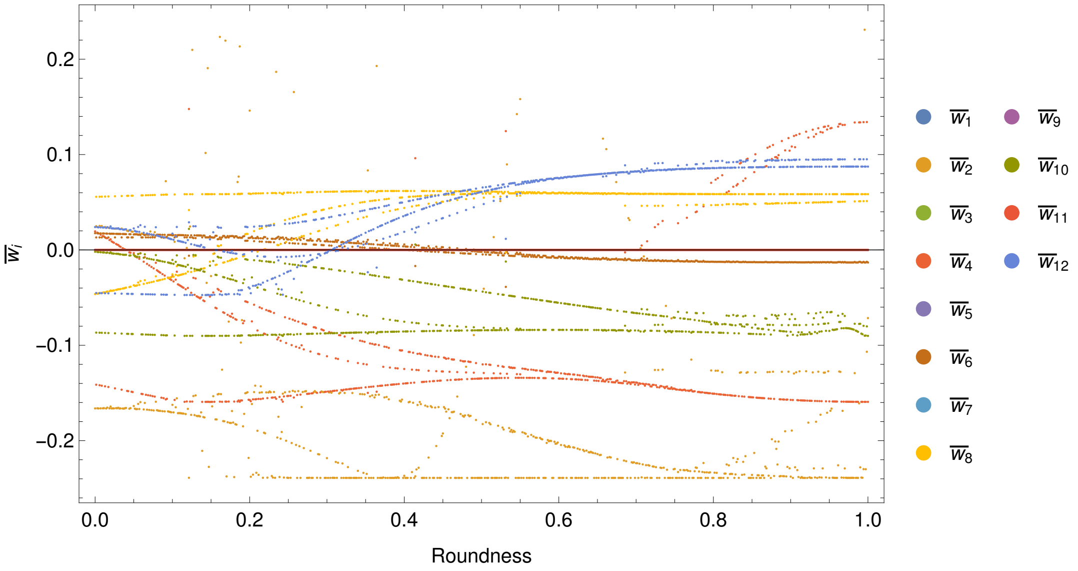


Figure 53: Bond order parameters $\bar{w}_1$ to $\bar{w}_{12}$ as a function of roundedness parameter r for increasingly rounded truncated tetrahedra with $N = 2$.

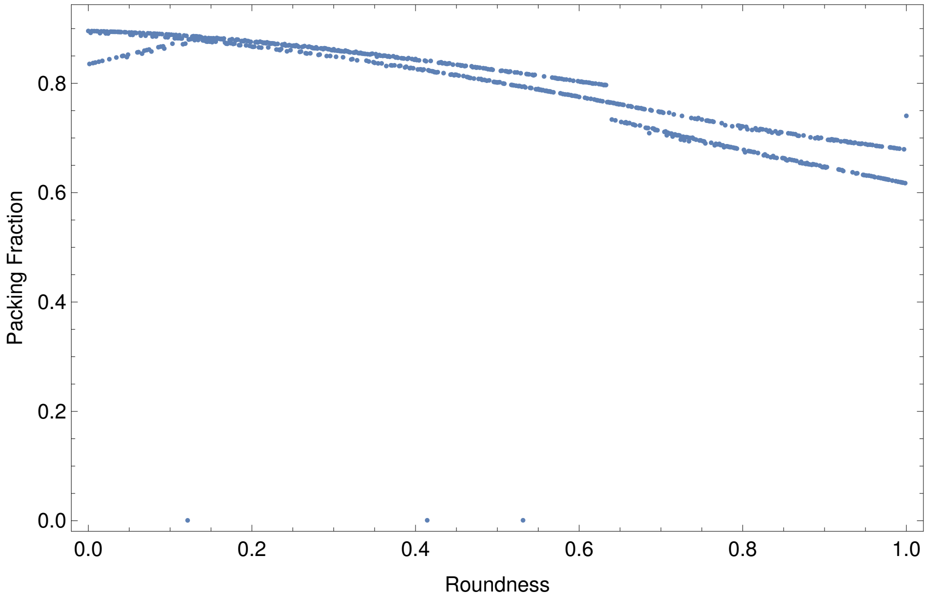


Figure 54: Packing fraction as a function of roundedness parameter r for increasingly rounded truncated tetrahedra with $N = 2$.

A.5.2 PCA

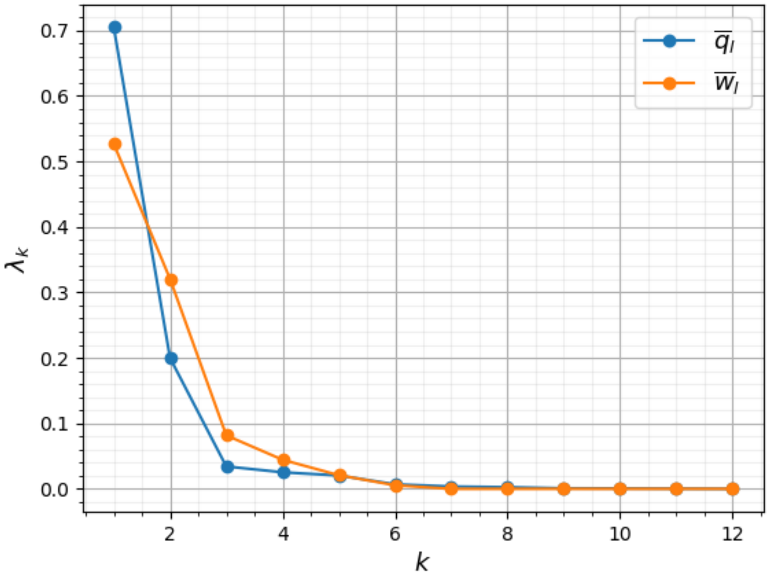
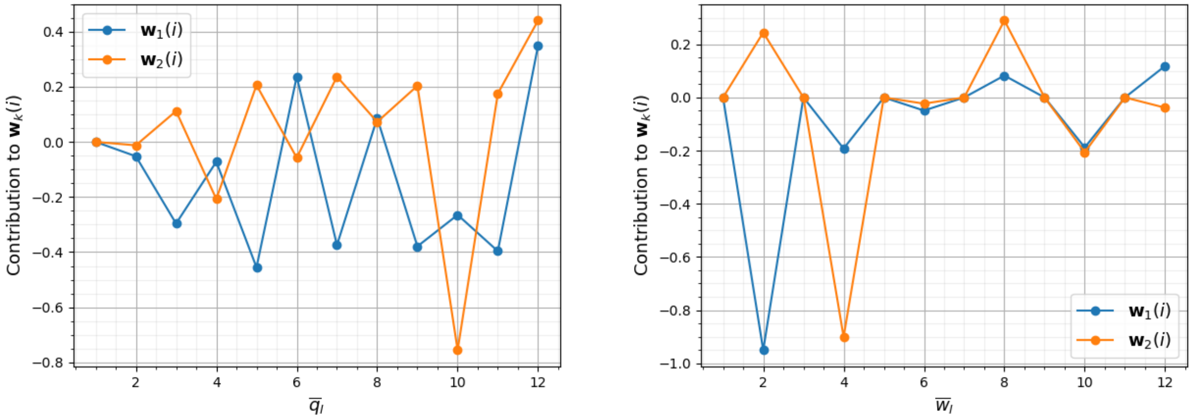


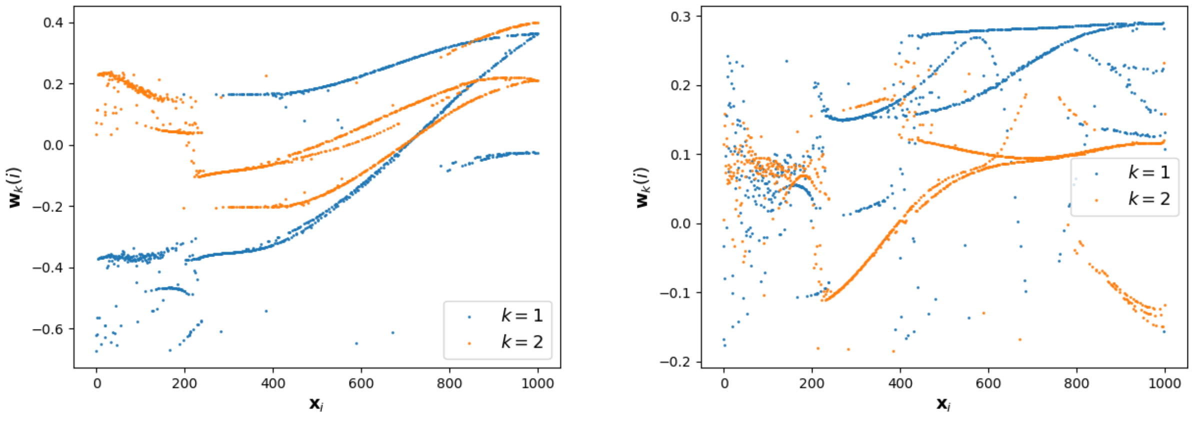
Figure 55: Normalized eigenvalues found with PCA sorted in non-ascending order for the data sets with the $\bar{q}_l$ and $\bar{w}_l$ bond order parameters for rounded truncated tetrahedra.



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded truncated tetrahedra.

(b) Weight of bond order parameters $\bar{w}_l$ contributing to principal component for rounded truncated tetrahedra.

Figure 56: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded truncated tetrahedra.

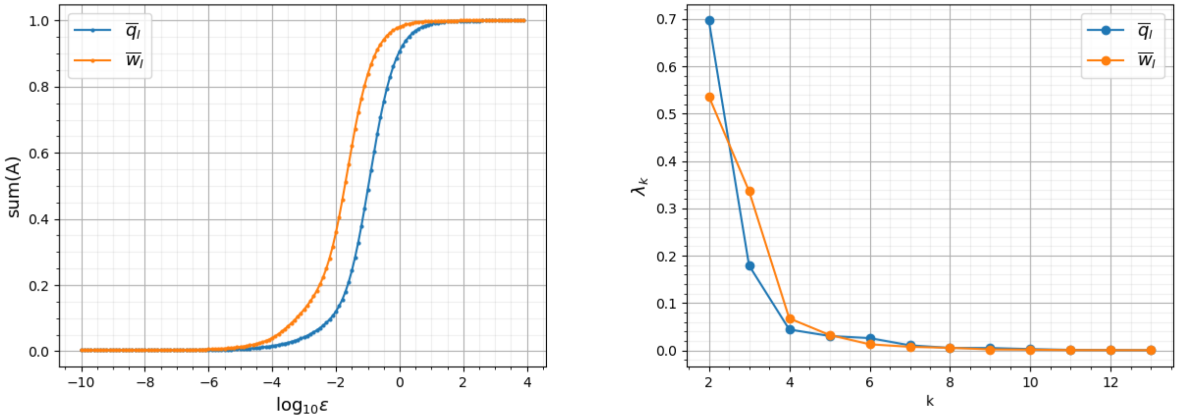


(a) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded truncated tetrahedra.

(b) Eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ found by PCA as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded truncated tetrahedra.

Figure 57: Eigenvectors found with PCA plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 57a) and $\bar{w}_l$ (figure 57b) for rounded truncated tetrahedra.

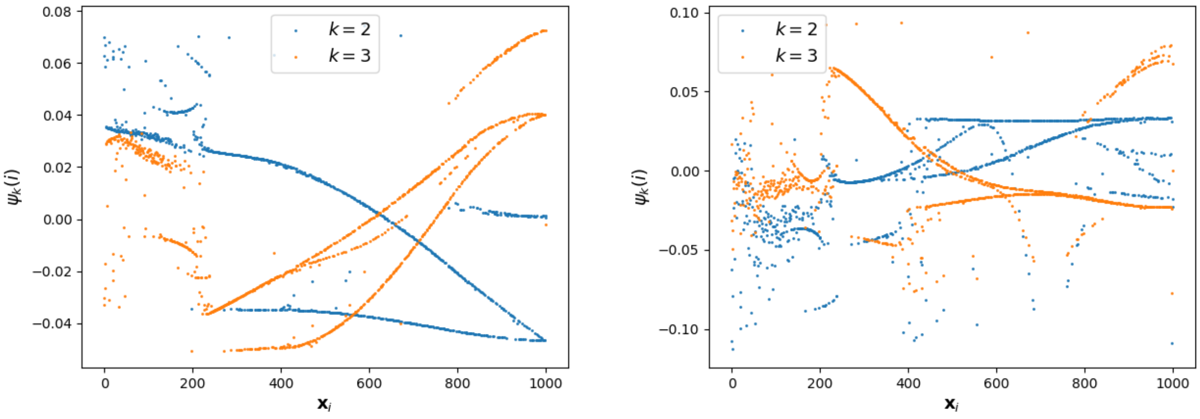
A.5.3 dMaps



(a) Weight of bond order parameters $\bar{q}_l$ contributing to principal components for rounded tetrahedra.

(b) Normalized eigenvalues found with dMaps with $\varepsilon = 0.5$ for bond order parameters $\bar{q}_l$ and $\varepsilon = 0.4$ for BOP $\bar{w}_l$, sorted in non-ascending order.

Figure 58: Contribution of bond order parameters $\bar{q}_l$ and $\bar{w}_l$ to principal components $\mathbf{w}_k(i)$ found by PCA for rounded tetrahedra.



(a) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{q}_l$ for rounded tetrahedra.

(b) Eigenvectors ψ_2 and ψ_3 found by dMaps as a function of roundedness $\mathbf{x}_i$ for the bond order parameters $\bar{w}_l$ for rounded tetrahedra.

Figure 59: Eigenvectors found by dMaps plotted as a function of roundedness $\mathbf{x}_i$ for rounded cubes for bond order parameters $\bar{q}_l$ (figure 59a) and $\bar{w}_l$ (figure 59b).