ENHANCED SELF-ORGANISING MAP MODEL FOR SURFACE RECONSTRUCTION OF UNSTRUCTURED DATA

YOU CHENG CHUN

MASTER OF SCIENCE (COMPUTER SCIENCE)

FACULTY OF INFORMATION AND COMMUNICATION
TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN
NOVEMBER 2023

ENHANCED SELF-ORGANISING MAP MODEL FOR SURFACE RECONSTRUCTION OF UNSTRUCTURED DATA

By

YOU CHENG CHUN

A dissertation submitted to the Department of Computer Science,
Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Science (Computer Science)
November 2023

ABSTRACT

ENHANCED SELF-ORGANISING MAP MODEL FOR SURFACE RECONSTRUCTION OF UNSTRUCTURED DATA

You Cheng Chun

Surface reconstruction (SR) is a process of recovering the digital representation of an object in reverse engineering. When the unstructured data are applied in the SR process, incorrect surface is produced because the data do not have any connectivity information. Self-Organising Map (SOM) models were proposed to organise the unstructured data to regain the connectivity information, but incorrect surface with holes, internal neurons and different grid sizes problems were appeared. Although the SOM model can generate the correct surface, its output is not in the standard format of Computer Aided Geometric Design. So, Non-Uniform Rational B-Spline (NURBS) surface approximation approach was applied to the output using parameterisation methods. However, the surfaces generated still contain gaps and were not optimal. Hence, the surfaces can be optimised using optimisation techniques. Therefore, the objectives of this research are to propose a SOM model for organising the unstructured data and to present and optimise the NURBS surface approximation approach with Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimisation (PSO). The data set used includes four primitive objects and a medical image data. The codes were developed using Microsoft Visual Studio 2022 with C++ programming and GNUPlot was used to visualise the results. The results shown that the Double Net SOM (DNSOM) model performed faster than 3-D SOM and Cube Kohonen SOM (CKSOM), achieved the lowest Topographic Error and generated the correct surface with fewer neurons compared to CKSOM. Additionally, the improved NURBS approach with Chord Length method was able to generate the correct surface with no gaps and the least surface error. DE can optimise the improved NURBS surface better compared to GA and PSO by achieving 243 out of 280 least optimised surface errors. The research outcomes can be utilised in reverse engineering to recover the surface of an object.

ACKNOWLEDGEMENT

First and foremost, I would like to acknowledge the support of Ministry of Higher Education (MOHE) for funding this project under the Fundamental Research Grant Scheme (FRGS) with the grant number FRGS/1/2019/ICT02/UTAR/02/3. In addition, I would also like to acknowledge the support of Universiti Tunku Abdul Rahman (UTAR) in funding my research under UTAR Research Fund (UTARRF) Topup Scheme with the vote number 6235/L18.

Next, I would like to convey my deepest appreciation to my supervisor, Ts. Dr. Lim Seng Poh, for his professional advice and guidance throughout my journey as a Masters student. The completion of this project would not have been possible without him as someone who has always inspired and motivated me to do better in the work I pursued.

Special thanks go to my co-supervisor, Ts. Dr. Lee Cheng Kang, who was very attentive and patient in every meeting we conducted. The brilliant comments and suggestions he gave have been really useful to my project.

I would also like to express my heartfelt gratitude to my parents, You Ming Hai and Siew Swee Lian, for their unconditional support and encouragement. I am truly grateful to be born into this loving family and to have them as my parents.

Last but not least, I wish to extend my sincere thanks to my close friends whom I can share my ups and downs with throughout this research journey. Their accompaniment has made me mentally stronger to fight against challenges and keep helplessness and loneliness at bay.

APPROVAL SHEET

This dissertation entitled "**ENHANCED SELF-ORGANISING MAP MODEL FOR SURFACE RECONSTRUCTION OF UNSTRUCTURED DATA**" was prepared by YOU CHENG CHUN and submitted as partial fulfillment of the requirements for the degree of Master of Science (Computer Science) at Universiti Tunku Abdul Rahman.

Approved by:

(Ts. Dr. Lim Seng Poh) Date: 15th November 2023
Supervisor
Department of Computer Science,
Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman.

(Ts. Dr. Lee Chen Kang) Date: 15th November 2023
Co-supervisor
Department of Internet Engineering and Computer Science,
Lee Kong Chien Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 15th November 2023

SUBMISSION OF DISSERTATION

It is hereby certified that ***You Cheng Chun*** (ID No: ***20ACM02161***) has completed this dissertation entitled "*Enhanced Self-Organising Map Model for Surface Reconstruction of Unstructured Data*" under the supervision of Ts. Dr. Lim Seng Poh (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology, and Ts. Dr. Lee Chen Kang (Co-Supervisor) from the Department of Internet Engineering and Computer Science, Lee Kong Chien Faculty of Engineering and Science.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*YOU CHENG CHUN*)

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

(You Cheng Chun)

Date: 15th November 2023

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CAD	Computer-Aided Design
CAGD	Computer-Aided Geometric Design
CKSOM	Cube Kohonen Self-Organising Map
CN	Control Net
DE	Differential Evolution
DLSOM	Deep Learning Self-Organising Map
DNSOM	Double Net Self-Organising Map
GA	Genetic Algorithm
LiDAR	Light Detection and Ranging
NON	Number of Neurons
NOV	Number of Output Neurons
NOR	Number of Redundancies
NURBS	Non-Uniform Rational B-Spline
PSO	Particle Swarm Optimisation
SOM	Self-Organising Map

CHAPTER 1

INTRODUCTION

1.1 Overview

Surface reconstruction is defined as a process of rebuilding a surface based on different input data, such as triangular mesh, set of structured or unstructured point clouds and intersection lines, whereby the surface generated must represent the original object's surface [1]. It is undeniably a very challenging task arising in various areas such as Computer-Aided Design (CAD) [2], medical imaging [3], [4], [5], geology [6], [7], [8], [9], [10], urban reconstruction [11] and reverse engineering [12], [13], [14], [15]. Thus, surface reconstruction in these areas is an ongoing and popular research that can be explored.

The surface reconstruction pipeline begins with the acquisition of the point clouds representing the geometry of scanned objects and surroundings [16]. The point clouds are acquired from depth cameras [17] or Light Detection and Ranging (LiDAR) sensors [18]. The data acquired in surface reconstruction can either be structured or unstructured [16]. It is important to identify them before choosing which surface reconstruction techniques to be used to reconstruct the surface of the collected data [19]. Structured data contain connectivity information while the unstructured data do not have any connectivity information [20], [21], [22]. Since unstructured data do not have any connectivity information, it is challenging to reconstruct the surface of the unstructured data because the data should be organised correctly to regain the correct connectivity information among the data. The surface generated would be incorrect if the correct connectivity information among the data is not regained. In addition, a correct and smooth surface with minimum surface error should also be generated when reconstructing the surface of the unstructured data.

Various Self-Organising Map (SOM) models were utilised to organise the unstructured data. Structured data were produced when the SOM model organised the unstructured data correctly. However, the structured data are not in the standard format for Computer-Aided Geometric Design (CAGD) [23]. Parametric surface such as Non-Uniform Rational B-Spline (NURBS) is often used in CAGD due to its stability, flexibility and local modification properties. According to Knopf, Sangole and Archana [24], parametric surface fitting requires prior knowledge about the connectivity between the data. Since the connectivity information between the data is obtained after the data is organised, the parametric surface fitting can be applied on the organised data. Lim and Haron [25] proved that the NURBS surface approximation approach can be applied after the unstructured data was organised successfully.

Although the NURBS surface approximation approach is proven to be applicable to structured data, the NURBS surface generated is not the most accurate. Previous works [26], [27], [28], [29], [30], [31], [32] have shown that parametric curves and surfaces such as Bézier, B-Spline and NURBS can be optimised by adjusting their parameters to generate better curves and surfaces of the data with various optimisation techniques [33], [34]. Optimisation techniques that were used to optimise the parametric curve and surface include Genetic Algorithm (GA) [26], [27], [35], [36], [37], Differential Evolution (DE) [28], [29] and Particle Swarm Optimisation (PSO) [30], [31], [32].

1.2 Problem Background and Statements

Previous works have shown that Self-Organising Map (SOM) models were used to regain the connectivity information of the unstructured data. The 2-D SOM model was utilised to organise the unstructured open surface data. Although the model can organise the unstructured open surface data, it fails to organise and generate the correct surface given the unstructured closed surface data. The surface generated contain holes [25], [42], [43]. To tackle this limitation, a SOM model was initialised as an icosahedron and undergone the subdivision process, is utilised to organise the unstructured closed surface data in [41]. Furthermore, a Deep Learning SOM (DLSOM) [42] model was introduced to organise the unstructured closed surface data. Apart from the 2-D SOM model, the 3-D SOM model was not able to organise the unstructured data and generate the correct surface due to the existence of the internal neurons as highlighted in [25]. Hence, Cube Kohonen SOM (CKSOM) [25] was introduced to organise the unstructured closed surface data and to overcome the limitation of 2-D SOM and 3-D SOM models. Despite the strengths of the model, the length and width of its grid cannot be tuned with different values.

After the unstructured data was organised successfully by the SOM model, the data would be structured data. Several previous works [23], [28], [29], [32] represented the structured data with parametric surfaces via surface approximation approach. In [32], B-Spline surface approximation was employed as the surface representation for the output of the growing grid SOM. Furthermore, NURBS surface approximation approach was employed in [28] on the output of the growing grid SOM. The NURBS surface approximation approach was also applied on the CKSOM model to represent the output of the model with the standard representation format for the CAGD [23]. However, the NURBS surfaces generated contain gaps at the boundary of the surface. Hence, the NURBS surface approximation approach should be improved to overcome the problem.

Although the surface approximation is applicable on the output of the SOM models as the surface representation, the surface generated may not represent the original data accurately. Hence, various optimisation techniques from the area of soft computing were utilised to optimise the parameters of the parametric curve or surface so that a curve or surface with minimum error can be generated. The parametric curve or surface can be utilised to optimise the parameters of the parametric curve or surface. GA was utilised in [26] to perform parameter optimisation for B-Spline curve fitting. Hierarchical GA was introduced in [27] to perform B-Spline surface approximation. Furthermore, parallel hierarchical GA [36] was constructed to approximate the B-Spline curve given the unstructured data by locating the optimal number and locations of the knots, and the control points of B-Spline simultaneously. Besides, DE was used to find the optimal control points for the Bézier curve [29]. DE was also applied on the growing grid SOM to optimise the NURBS surfaces [28]. In addition, PSO was employed in [32] to optimise the B-Spline surface representing the organised data of the growing grid SOM by finding the optimal control points. In [30], PSO was utilised to obtain the optimal values for all the coefficients of NURBS given different 3-D data points.

Although various SOM models such as the 2-D SOM and 3-D SOM were proposed to organise the unstructured data, they are still suffering from limitations as stated in [25]. The limitations include holes problem in 2-D SOM [43] and connectivity problems in 3-D SOM [25]. Besides, it was identified that the length and width of the grid in CKSOM cannot be tuned with different value. So, to generate a surface that is similar to the original object, the unstructured data should be organised appropriately. Additionally, representing the structured data with suitable surface approximation is important in surface reconstruction because the surface produced by the SOM models is not the standard format in CAGD. However, gaps appeared when it was applied on the SOM model as shown in [23] which might cause higher surface error. So, the surface error can be optimised with the optimisation techniques as shown in [28], [32]. Therefore, the surface represented should achieve minimum surface error and similar to the original object.

1.3 Research Objectives

The objectives of this research are:

i. To propose a SOM model for organising unstructured data.

ii. To propose a surface approximation approach based on the proposed SOM model in representing the surface.

iii. To optimise the surface approximation approach through the implementation of GA, DE and PSO.

1.4 Research Scopes

The scopes of the research are stated as follows:

i) Data sets

- Four primitive shape data such as cube, sphere, spindle and oiltank [44], and a medical data which is talus bone [45] are the data sets used in this research. The four primitive shape data are used to show that the proposed SOM model, known as Double Net SOM (DNSOM) model can organise unstructured data, regain their connectivity information and apply on the medical data.
- The data sets are unstructured data in the form of coordinates (x, y, z).
- Additional data set (Stanford bunny data [46]) is used in this research as an additional testing to further evaluate the model.

ii) Parameterisation methods

- The Uniform, Chord Length, Centripetal and Exponential are the only parameterisation methods considered to be applied in the improved NURBS surface approximation approach.
- Various sizes of control net (CN) are utilised to examine the performance of the parameterisation method.

iii) Optimisation

- GA with Tournament Selection, Uniform Crossover, Random Mutation and Weak Parent Replacement, DE [33] and PSO with velocity clamping and constriction factor [34] are the optimisation methods applied on the control points of the improved NURBS surface approximation approach to minimise the surface error.

iv) Performance Measurements

- Minimum and maximum errors, Quantisation Error (QE), Topographic Error (TE) and CPU time are used in the first objective to evaluate the performance of the 2-D SOM, 3-D SOM, CKSOM and DNSOM models. There is no ground truth available because SOM model is an unsupervised learning model [146].
- Surface error based on Euclidean distance [23] and the concept of the DNSOM model in second objective was utilised to measure the performance between the surface approximation approaches and parameterisation methods.
- Optimised surface error based on Euclidean distance and concept of the DNSOM model was used in third objective to evaluate the performance between the optimisation techniques.

v) Visualisation

- GNUPlot is used to visualise the data sets and the results of the first, second and third objective.

1.5 Dissertation Organisation

This dissertation comprised of seven chapters. The organisation of this dissertation is as follows. Chapter 1 provides a general overview and the problem background and statements of this research. This chapter also highlights the objectives and the scope of this research. Chapter 2 reviews the related works. Various Self-Organising Map (SOM) models, parametric curves and surfaces such as B-Spline and Non-Uniform Rational B-Spline (NURBS) are reviewed in Chapter 2. Apart from that, optimisation techniques such as GA, DE and PSO are also reviewed in Chapter 2. Chapter 3 demonstrates the research methodology of this research. This chapter also includes the hardware and software used in this research.

Meanwhile, Chapter 4 describes the system flow of DNSOM model, analyses and discusses the performance of the DNSOM model. Chapter 5 explains the steps of the improved NURBS surface approximation approach. Besides, Chapter 6 describes the system flow of the optimisation techniques applied to optimise the improved NURBS surface approximation approach and this Chapter also analyses and discusses the performance of the optimisation techniques. Chapter 7 provides the conclusion of this research. The limitations, future works and contributions of this research are also presented in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter discusses the process and data in surface reconstruction. The existing surface reconstruction techniques, surface representation and surface approximation in surface reconstruction are discussed thoroughly in this chapter. This chapter also reviewed the existing optimisation techniques.

2.2 Surface Reconstruction

Surface reconstruction is the process of retrieving the data through the scanning of objects and reconstructing the surface of the retrieved data [19]. The process and data involved in surface reconstruction are discussed in this section.

2.2.1 Process

The process in surface reconstruction consists of parameterisation and surface approximation [30], [32], [47]. Parameterisation is a process of generating the parameters that define the relationship among the surface data and surface approximation is the process of fitting the surfaces according to the parameter values of the parameterisation [32]. In [23], [32], data are organised initially and the data are presented with the free-form parametric surface through parameterisation and surface approximation after the data are organised. Besides, the free-form parametric surface in [32] was optimised with optimisation technique to produce a more accurate surface. In [30], optimisation technique is used to perform both parameterisation and surface approximation on the data which are unstructured to generate a correct surface with high accuracy. Therefore, these previous works show that the data determine how the parameterisation and surface approximation is performed because structured data with rectangular topology can be applied with parametric surfaces by performing parameterisation and surface approximation as a linear problem but unstructured data would have to apply the parametric surfaces as a complex high-dimensional non-linear optimisation problem.

2.2.2 Data

Data in surface reconstruction are collected through the scanning of the existing objects with depth cameras [17] or Light Detection and Ranging (LiDAR) sensors [18]. The data collected can be in the form of structured or unstructured [48]. Data with connectivity information are known as structured data. Meanwhile, data without any connectivity information are known as unstructured data [49]. When both of the data are used in surface reconstruction, a correct surface would be produced for the structured data and an incorrect surface would be produced for the unstructured data because structured data have the connectivity information among the data but the unstructured data do not have any connectivity information among the data. Thus, the unstructured data need to be organised correctly so that the correct connectivity information among the data can be obtained and a correct surface can be produced. It is important to obtain the correct connectivity information for the unstructured data because the reconstructed surface relies on it [16]. The structured data can be presented with the parametric surface through parameterisation and surface approximation. Besides, other properties of the data such as noise, sampling density, misalignment, outliers and missing data have the impact in surface reconstruction [50]. These data make the task of computing a surface representation that resembles the original object and retaining their critical surface features a very challenging task [51].

2.3 Surface Representation and Approximation

Explicit and implicit are the two types of surface representation [52]. Triangulated and parametric surfaces are the two types of explicit surfaces [19]. Triangulated surface was generated with Voronoi diagram (VD) and its dual, Delaunay triangulation (DT) techniques such as Crust [68], Cocone [69], Power Crust [70], SuperCocone [71], Tight Cocone [72] and Localised Cocone [73]. However, most of the techniques are not robust towards noisy and non-uniform data. In contrast, implicit surface is generated by initially define an implicit function and extract the zero-level iso-surface. Then, the zero-level iso-surface can be visualised through ray casting and marching cube [53]. The implicit function can be a signed distance function [54], [55], [56], [57], [58], radial basis function (RBFs) [59], [60], [61], piecewise polynomial functions [62], indicator functions [63], [64], [65] or wavelets [66], [67]. Besides, parametric surface can be generated through surface interpolation or approximation. Surface approximation generates the surface better than the surface interpolation because surface approximation generates the surface that approximate the data points, minimising the surface error, whereas the surface interpolation generates the surface that passes through all the data points [74], [75]. This makes the surface approximation to be more robust towards data with noise and the surface interpolation to be sensitive to data with noise [76]. Surface approximation can minimise the influence of the noise in the data [76].

Surface interpolation would generate a surface that interpolates the incorrect data point when the data has noise [76]. Surface approximation is performed after the parameter values of the data is computed through parameterisation. A good parameterisation is essential to determine the connectivity among the data in the surface parametric domain, and the topology and boundaries of the surface [30]. Data with rectangular topology is required for the parametric surface because parametric surface use such data as input [77]. Hence, the method used to organise the unstructured data must have the rectangular topology. The parametric surface is used primarily in the computer-aided geometric design (CAGD) [78] because they have a great flexibility and can represent any smooth shape well [74].

2.3.1 Parametric Representation on B-Spline and Non-Uniform Rational B-Spline

The parametric representation is referred to the representation of the data with free-form parametric curves and surfaces such as B-Spline and Non-Uniform Rational B-Spline (NURBS). This section discusses the mathematical formulation of the B-Spline and NURBS curves and surfaces. B-Spline curves and surfaces were introduced due to the global influence of the control points in Bézier curve and surface [76], [79]. The entire curve or surface will be affected when a single control point was adjusted. The B-spline curve with n + 1 control points P_i (i = 0, …, n) and degree p is defined as follows [80]:

$$C(t) = \sum_{i=0}^{n} N_{i,p}(t)P_i \tag{2.1}$$

where $N_{i,p}(t)$ are the normalised B-Spline basis function defined on a knot vector $T = \{t_0 = \dots = t_p = 0, t_{p+1}, \dots, t_n, t_{n+1} = \dots = t_{n+p+1} = 1\}$. The knot vector

T consists of non-decreasing real numbers knots on the interval [0, 1]. The first and last knots of T are usually repeated with multiplicity equal to degree p + 1. $N_{i,p}(t)$ can be defined as follows [81], [82]:

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{2.3}$$

The B-spline surface of degrees (p, q) with (m + 1) × (n + 1) control points is defined as follows:

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{2.4}$$

where $P_{i,j}$ is the control points, p and q are the degree of the surface in u- and v-direction, $N_{i,p}(u)$ and $N_{j,q}(v)$ are the normalised B-spline basis functions given the knot vectors $U = \{u_0 = \ldots = u_p = 0, u_{p+1}, \ldots, u_n, u_{n+1} = \ldots = u_{n+p+1} = 1\}$ and $V = \{v_0 = \ldots = v_q = 0, v_{q+1}, \ldots, v_m, v_{m+1} = \ldots = v_{m+q+1} = 1\}$ in u- and v-direction respectively.

NURBS is also one of the most common surface representations in real world applications [30]. A NURBS curve of degree p is defined as follows [83], [84]:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{2.5}$$

where $C(u)$ represents a point on the NURBS curve at u, P_i are the control points, w_i are the corresponding weights of the control points P_i, $N_{i,p}(u)$ are the normalised B-Spline basis function and n + 1 is the number of control points.

A NURBS surface of degree (p, q) with (n + 1) × (m + 1) control points is defined as follows [83], [84]:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{2.6}$$

where, u and v are the parameter, $P_{i,j}$ is the (i, j)-th control points in 3-D space, $w_{i,j}$ is the corresponding weight of $P_{i,j}$, $N_{i,p}(u)$ and $N_{j,q}(v)$ are the i-th and j-th B-Spline basis functions of degree p and degree q defined on knot vectors U and V in the u- and v-direction respectively which is computed recursively with Equation 2.3.

One of the important properties of the NURBS surface is that the four corners of the NURBS surface data interpolates the four corner control points [147]. The difference between the NURBS and the B-Spline is the existence of weight associated to each control points. By setting the weight of all the control points in Equations 2.5 and 2.6 to 1.0, the NURBS curve and surface becomes a B-Spline curve and surface respectively [85]. The weight associated to each control point influences the NURBS curve or surface [86]. When the weight of a control point increases, the point on the NURBS curve or surface will move towards the corresponding control point or vice versa [81].

However, it is challenging to perform curve or surface approximation with NURBS because it is a non-linear problem as there are more than one unknown variable such as control points and their weight needed to be computed. Other than the control points and their weight, the parameters and knot vectors are also considered as unknown variables in [30]. Various previous works were proposed to perform curve and surface approximation with NURBS. A two-step linear approach [85] was proposed in which the weights were initially identified with symmetric eigenvalue decomposition to solve the non-linear problem. After the weights were identified, the weights were used to compute the control points. Furthermore, a weights iterative optimisation technique for NURBS curve fitting was introduced in [81] to obtain the optimal weights with the least square method. In [23], [29], [37] SOM models with rectangular topology were used to organise the unstructured data and surface approximation was performed on the output of the models. The NURBS surfaces can be applied on the output of the SOM models with rectangular topology because the NURBS surfaces use such data as input [77]. Other than the influence of weight on the NURBS curve and surface, the parameterisation of the data points also influences the accuracy of NURBS curve and surface obtained [26].

2.3.2 Parameterisation and Surface Approximation

Parameterisation is a process of defining the relationship among the data and surface approximation is the process of fitting the surfaces according to the parameter values of the parameterisation [29]. Parameterisation is an important process because it influences the performance of the NURBS curve and surface approximation as shown in [87], [23] as different parameterisation methods would produce different parameter values and the parameter values reflect the distribution of the data. Appropriate parameterisation is required to obtain a good surface approximation [47]. The most commonly used parameterisation methods are Uniform, Chord Length, Centripetal and Exponential parameterisation methods [84], [23], [87], [88]. Uniform method is the simplest method [87] because the parameter values are computed without the data points. Parameter values are computed from the Uniform method by utilising the definition in 2.7 [89]:

$$t_i = \frac{i}{n}, 1 \leq i \leq n-1 \tag{2.7}$$

where $t_0 = 0$ and $t_n = 1$ are the first and last parameters, t_k is the middle parameters, n is the index of the last point. The Chord Length method produces better results compared to the Uniform method if the data point is distributed non-linearly [95]. Meanwhile, the Centripetal method is an extension of the Chord Length method. By replacing the power factor, α in Equation 2.8 from 1.0 to 0.5, the Chord Length method is extended to the Centripetal method. According to Iglesias et al. [148], the Centripetal method commonly yields better results than the Chord Length method for shapes with sharp turns. Exponential method was proposed in [88] by setting the power factor, α in Equation 2.8 to 0.8. Parameter values are computed from the Centripetal, Exponential and Chord Length methods by applying α in Equation 2.8 with 0.5, 0.8 and 1.0 respectively [79], [89]:

$$t_k = \frac{\sum_{i=1}^{k} |Q_i - Q_{i-1}|^{\alpha}}{\sum_{i=1}^{n} |Q_i - Q_{i-1}|^{\alpha}}, k = 1, \cdots, n-1 \tag{2.8}$$

where $n + 1$ is the total number of data points or total number of parameters, $t_0 = 0$ and $t_n = 1$ are the first and last parameters, t_k is the middle parameters, $\left|Q_i - Q_{i-1}\right|$ is the distance between adjacent data points Q_i and Q_{i-1}, α is the power factor and L is the length of the data polygon.

The knot vector is generated after the parameter values are computed. Knot values can be generated with averaging knot vector method and it is defined as follows [79]:

$$T_0 = T_1 = \cdots = T_p = 0 \tag{2.9}$$

$$T_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} t_j \, , j = 1, 2, \cdots, n - p \tag{2.10}$$

$$T_{m-p} = T_{m-p+1} = \cdots = T_m = 1 \tag{2.11}$$

where t is the knot vector. There would be $m + 1$ knots, where $m = n + p + 1$ for $n + 1$ parameters with t_0, t_1, …, t_n with the degree p. This method was used in this research because it generates the knots according to parameter values.

Different parameterisation methods were used to compute the parameter values of various curve data for the B-Spline curve in [87] and the results show that the parameterisation methods do affect the results. Hence, the selection of an appropriate parameterisation methods is important for surface approximation [34]. In 2008, Forkan and Shamsuddin [32] represented the 3-D structured data of the growing grid SOM with B-Spline surface using the Centripetal method. Additionally, Zhang, Feng and Cui [90] conducted NURBS surface approximation with Chord Length method. Lim and Haron [23] proposed the use of NURBS surface approximation approach with different number of control points and parameterisation methods (Uniform, Chord Length, Centripetal) to represent the closed surface data of the Cube Kohonen Self-Organising Map (CKSOM) model. This approach would be applied in this research since it was proposed to be applied on multiple SOMs. Besides, NURBS was applied in [91] to approximate the surface of the data after the corner, boundary and interior points of the data are identified and classified using a deep neural network.

Based on the previous works [23], [32], parametric surface was applied on the unstructured data after the connectivity information among the data were regained and the model used to organise the unstructured data in these previous works is Self-Organising Map (SOM) model. So, various SOM models were discussed in the next section.

2.4 Self-Organising Map Model

Self-Organising Map (SOM) [92] is an unsupervised learning neural network that performs dimensionality reduction by representing high-dimensional data with much lower dimensional space [93]. It was introduced by a Finnish professor Teuvo Kohonen [94]. The SOM contain neurons, arranged in triangular, rectangular or hexagonal topology [95], [96]. Each neuron is associated with a weight vector, which represents its location in the input space and has the same dimension as the input vectors [41]. Neurons with the smallest Euclidean distance between the weight and the input vectors is selected as the winning neuron [97]. Three main phases involved in the learning process of SOM are competition, cooperation and adaptation [98]. These phases will eventually update the weight vector of each neuron towards the input vectors. The common termination criterion employed in SOM is maximum number of iterations [99], [100], [101].

As shown in the previous works [41], [102], [103], [24], [42], [104], [105], [106], [22], [40], various SOM models were applied in surface reconstruction and the organisation of the unstructured data. Different SOM models were proposed to solve the problem of existing SOM models. In addition, improvements were made on the existing SOM models to increase their performance as shown in [103], [25]. In 1999, Hoffmann [103] made a modification on the SOM neighbourhood function by replacing the data type of the return value from natural number to positive real number. The new function prevented the radius to decrease by 1 unexpectedly and gradually. The improvement helped to move the neurons closer to the input points. Consequently, a smoother surface was obtained. Furthermore, the 2-D SOM with rectangular structure, and a spherical SOM with the shape of an icosahedron and triangular mesh were used by Yu [41] to reconstruct the open and closed surface data respectively. Multiresolution learning and edge swap were suggested to learn the concave structures of an object. Although the suggested model can deal with the concave structures of the object, the use of edge swap is not user adaptive as it requires the user to shift the vertices of their incident triangles that are far away from the input data close to the input data. This action has to be repeated several times until the vertices reach a nearby input data.

Conformal self-organising map (CSM) [105] was proposed to provide conformal mapping to achieve the conformality requirement in the mapping of geometrical surface. The CSM and SOM were used to perform geometrical surface mapping on several data sets such as the 2-D uniformly-distributed square and 3-D half sphere, and their performances were evaluated with distortion error. Distortion error was calculated from the dimensionality reduction and quantisation errors. The CSM achieved lower distortion error when the total number of iterations increased indicating that the neurons in the CSM were updated nearer to the input data. However, the SOM generates the results faster than the CSM. Moreover, a conformal spherical self-organising map (CSSM), an extension of CSM was proposed to perform surface reconstruction of closed surface [104]. However, the surface produced was not smooth due to the use of flat triangles to represent the CSM. A method was proposed [106] to construct a piecewise smooth seamless surface with special correspondence based on the derived surface of the CSM to produce a smooth surface.

The Kohonen learning rule was presented in [102] to perform the adaptation process. New neurons were added to increase the adaptation process at particular area. Besides, local subdivision task was on the identified problematic triangles and local adaptation process was applied on the subdivides area. With the approach, the issue of SOM in reconstructing concave regions is solved. Also, spherical SOM model was proposed in [24] to organise the unstructured data. The neurons in the spherical SOM were decorated with uniform triangles on a tessellated unit sphere. The Region-of-Influence (ROI) procedure was adapted in the learning process of the model to generate a correct closed surface for object with concave areas and objects with holes. The procedure eliminates the need to refine the generated surface with mesh refinement transformations. However, this model was very vulnerable towards the density of the data [24]. Also, it might wrongly label the area of sparse data as intended holes and cause the model fit around it [24].

Furthermore, a multiresolution strategy for surface reconstruction of unstructured data was introduced in [98]. This method suggested the use of batch SOM, a set of mesh operators such as vertex removal, edge swap, triangle subdivision and vertex split, and simple constraints for selective mesh refinement to tackle the issue of SOM in reconstructing the surface at the concave regions of an object. This method can represent the surface of an object with smaller number of triangles for the same resolution level compared to the surface reconstruction method proposed in [41]. It allows the user to set the threshold which controls the number of triangles in the meshes unlike the method proposed in [41] which subdivides every face in the mesh into four smaller faces at a certain resolution to get a higher resolution surface. In addition, a growing grid SOM with rectangular map was introduced in [32] to organise the unstructured data of open surfaces. The growing grid SOM can generate the open surfaces correctly. However, the map is not suitable to reconstruct the closed surface data and has the same problem as the 2-D SOM because the neurons at the edges of the maps are not connected to one another. The difference between them is that the grid size of the growing grid SOM increases from time to time but the grid size of the 2-D SOM remains the same throughout the learning process after initialisation.

Due to the fixed topology of SOM, the models based on SOM may generate a reconstructed surface with vertices or triangles dangling around the regions of dense data that does not belong to the original object [40]. Nevertheless, methods in [41], [102], [24], [40], [22] can be used to solve this problem. However, growing self-organising surface map [22] cannot generate the surface correctly if the point cloud is not uniformly-sampled and it requires some post-processing steps. In addition, a method in which the map can grow incrementally to produce meshes with various resolutions was introduced in [40] to solve this problem. The method can produce models that fit the shape of an object, including its concave regions and holes. However, this method may not preserve the topology of the map. The method will keep on changing, inserting and removing the structural information of the map during the learning process.

Cube Kohonen SOM (CKSOM) [25] was introduced to address the holes problem of 2-D SOM in the reconstruction of closed surface and the connectivity problem of the 3-D SOM. The model was created through the merging of six 2-D SOMs. As shown in [25], the CKSOM model performs better than the 2-D SOM and 3-D SOM models and the model was also tested with different types of data sets. Despite its ability to deal with the problems of both 2-D and 3-D SOMs, it restricts the user to set different width and length of the grid. Hence, objects with longer widths or lengths may fail to reconstruct efficiently [25]. The model would use a larger grid size to reconstruct the objects. Furthermore, DL SOM [42] is also introduced to reconstruct the closed surface data. Although it could deal with closed surface, it failed to produce the correct connectivity for the concave area of the object. Therefore, DL SOM is not considered in this research.

Based on the previous works, SOM models were able to organise the unstructured data and generate the correct surface. However, previous works show that the SOM models are still suffering from limitation that needs to be addressed and the surface generated is not the standard representation in CAGD. Therefore, NURBS surface was often used to represent the output of the SOM models. Previous works [23], [28] have shown that the output of the SOM model with rectangular topology can be applied with NURBS surfaces. In addition. the NURBS surfaces can be optimised using optimisation techniques to generate surface with higher accuracy. Hence, the optimisation techniques related to surface reconstruction are discussed in the next section.

2.5 Optimisation Techniques

Optimisation techniques such as Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimisation (PSO) are often implemented to optimise the free-form parametric curve and surface such as B-Spline and NURBS in surface reconstruction.

GA was proposed by John Holland based on the survival of the fittest and it is a population-based optimisation technique. GA has four main operators and selection, crossover, mutation and replacement. Each individual or chromosome is a single possible solution in GA and each individual and chromosome is subdivided into genes. Each gene is represented according to an encoding scheme and the encoding scheme is determined according to the optimisation problem.

Encoding is a process of representing the chromosomes [107]. Value encoding scheme is mainly used in neural networks to find their optimal weights as the chromosome is represented with string of values and the values can be real, integer number or character. Selection is performed after the chromosomes are encoded. Selection is a process of choosing two or more individuals to perform crossover and mutation. Tournament Selection is the most popular selection scheme in GA because it is easy to implement [108], [109]. However, the tournament size set for the Tournament Selection cannot be too large. When the tournament size is large, the probability of loss diversity is also greater [110]. The individual selected to perform crossover is known as parent. Crossover operator is performed after the parents are selected with the selection operator. Crossover allows genes in the selected individuals to be exchanged to produce new solutions [111]. One of the common crossover schemes is Uniform Crossover. In Uniform Crossover, every gene is exchanged between the pair of randomly selected chromosomes with the swapping probability, p_e and it is typically set to 0.5 [113]. The crossover operator is used to prevent the duplication of the parents from the old population in the offspring. After the offspring are generated by crossover, the mutation is performed on the offspring. The mutation operator helps to maintain the diversity of the population by introducing new elements into the chromosomes [114] and prevents the algorithm from being trapped in a local minimum [107]. One of the mutation operators is Uniform Mutation [115]. In Uniform Mutation, a predefined number of genes are selected randomly and each of the gene is assigned with a value that is randomly generated within a certain range. It is used in value encoded GA. The replacement operator is executed after the mutation operator. The replacement operator includes Weak-Parent Replacement [107]. In Weak-Parent Replacement, the parents with lower fitness value than the offspring are replaced with their offspring in the next generation. The GA continues with selection, crossover, mutation and replacement operators until a stopping criterion is achieved which can be the achievement of certain number of generation or certain fitness value in which the fitness value is calculated with a fitness function.

Two important parameters used in GA are crossover probability, P_c and mutation probability, P_m. P_c and P_m are used to control the crossover and mutation respectively. P_c is in the range of [0, 1] in which 0 indicates that the completely new generation of individuals would be the same as the older population except those resulted from the mutation operator and 1 indicates that all the offspring are generated by crossover [112]. P_m is set in the range of [0, 1]. If the mutation probability is 1, the whole chromosome is altered, otherwise nothing is altered. The mutation probability should not be set too high as it will change the GA into a random search [107].

Differential Evolution (DE) was a population-based optimisation technique proposed by Storn and Price [33] in 1997 and its main operations are initialisation, mutation, crossover and selection. During initialisation, a population of N individuals are encoded as D-dimensional vectors of real numbers is generated and D elements of each vector are randomly initialised within [$x_{\min}$, $x_{\max}$] where $x_{\max}$ is the maximum value of the search space, $x_{\min}$ is the minimum value of the search space. The population with N individuals can be expressed as $X = (x_1, x_2, \ldots, x_N)$. Each individual can be expressed as $x_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t))$ where $x_i(t)$ is the i-th individual, $x_{iD}(t)$ is the D elements of the i-th individual at t and t is the current generation. Each individual is the possible solution to the optimisation problem.

Mutation operation is performed after the initialisation of the individuals. In mutation operation, a mutant vector is generated with a mutation strategy for each target vector. A mutant vector in the original DE is generated by adding the weighted difference between two vectors to a third vector and it is represented as follows [29]:

$$v_{i,G+1} = x_{r1,G} + F\left(x_{r2,G} - x_{r3,G}\right) \tag{2.12}$$

where r_1, r_2, r_3 are randomly chosen integers, mutually exclusive from one another and they are also different from the index i. Since r_1, r_2, r_3, and i are distinct, N must be greater or equal to four to abide this condition. F is a scaling factor which controls the amplification of the differential variations $(x_{r2,G} - x_{r3,G})$ and its value is within [0, 2].

The crossover was performed after the mutant vector was generated through the mutation strategy to increase the diversity of the population [116]. During the crossover, a trial vector $u_{ji,G+1}$ is developed from the elements of the target vector, $x_{i,G}$, and the elements of the mutant vector, $v_{i,G+1}$. Binomial crossover is the commonly-used crossover and it is defined as follow [117]:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } rand_{ji} \leq CR \text{ or } j = rnbr_i \\ x_{ji,G} & \text{if } rand_{ji} > CR \text{ and } j \neq rnbr_i \end{cases}, i = 1,2,\ldots,N; j = 1,2,\ldots,D \tag{2.13}$$

where $rand_{ji}$ is the uniform random number generator in the range [0, 1] for j-th dimension of the i-th individual, CR is the predefined crossover probability, chosen from the range [0, 1], $rnbr_i$ is a randomly selected integer is an element of 1, 2, …, D and it is to ensure that $u_{i,G+1}$ has at least one parameter from $v_{i,G+1}$.

The fitness value of the trial vector is evaluated with a fitness function and the selection operation is performed after the crossover operation. In selection operation, the fitness value of the trial vector is compared with that of the target vector in the current population. If the fitness value of the trial vector is less than or equal to that of the target vector, the target vector is replaced by the trial vector in the next generation. Otherwise, the target vector is remained for the next generation. The selection operation described above can be expressed as follows [118]:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } f(u_{i,G+1}) \le f(x_{i,G}) \\ x_{i,G} & \text{otherwise} \end{cases} \quad i = 1,2,\dots,N \tag{2.14}$$

where $x_{i,G}$ is the target vector of i-th individual, $u_{i,G+1}$ is the trial vector, $f(u_{i,G+1})$ is the fitness value of the trial vector, $f(x_{i,G})$ is the fitness value of the target vector, N is the total number of individuals.

Particle Swarm Optimisation (PSO) was developed by Eberhart and Kennedy [119] in 1995 and it was inspired by the social behaviour of bird flocking and fish schooling in food searching [119]. Various variants of PSO were introduced to solve different kinds of optimisation problems. The original PSO was defined in Equation 2.18. Each particle has velocity and position. The position of each particle corresponds to a possible solution of the problem. The position of each particle is initialised randomly within the lower and upper bound of the optimisation problem. The Equations 2.15 and 2.16 are used to update the particles positions.

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \tag{2.15}$$

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 r_1 \left(x_{ij}^{p(t)} - x_{ij}^{(t)} \right) + c_2 r_2 \left(x_j^{g(t)} - x_{ij}^{(t)} \right) \tag{2.16}$$

where i is the number of particle, t is the current generation, x_{ij} and v_{ij} are the i-th position and velocity of the particle in the j-th dimension, c_1 and c_2 are the cognitive and social acceleration constants respectively, r_1 and r_2 are the random number which are uniformly distributed between 0 and 1, $x_{ij}^{p(t)}$ is the best previous position of particle i in j-th dimension, $x_j^{g(t)}$ is the global best position in j-th dimension.

Other than the original PSO, various variants of PSO were introduced through the modification of Equation 2.16. One of the variants of PSO is the PSO with inertia weight and it is defined as follows:

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1 r_1 \left(x_{ij}^{p(t)} - x_{ij}^{(t)}\right) + c_2 r_2 \left(x_j^{g(t)} - x_{ij}^{(t)}\right) \tag{2.17}$$

The inertia weight, *w* is one of the PSO parameters originally proposed in [120] to balance the exploration and exploitation characteristics of PSO by controlling the velocity of the particles. Constant inertia weight within the range [0.9, 1.2] was preferred in [120]. Besides, PSO with constriction factor, *K* is also another variant of PSO and it was introduced by [121] to analyse the convergence behaviour [122]. It can be incorporated by modifying the Equation 2.16 to Equation 2.18 [123].

$$v_{ij}^{(t+1)} = K\left[v_{ij}^{(t)} + c_1 r_1 \left(x_{ij}^{p(t)} - x_{ij}^{(t)}\right) + c_2 r_2 \left(x_j^{g(t)} - x_{ij}^{(t)}\right)\right] \tag{2.18}$$

where

$$K = \frac{2}{\left|2 - \rho - \sqrt{\rho^2 - 4\rho}\right|}, \text{ where } \rho = c_1 + c_2, \rho > 4 \tag{2.19}$$

The constriction factor, *K* is set to 0.729 based on Equation 2.20.

$$\rho = c_1 + c_2 = 4.1 \tag{2.20}$$

where c_1 and c_2 are both set to 2.05.

Eberhart and Shi [123] compared the performance between the PSO with inertia weights and constriction factors and it was found that the performance of the PSO with constriction factor and velocity clamping is on par with the PSO with inertia weights. Velocity clamping was introduced to control the global exploration of the particle by keeping the particles within the search space [124].

GA, DE and PSO were applied in the optimisation of parametric curves and surfaces. In 2007, GA was applied in [78] to perform Bézier curve and surface parameterisation. Furthermore, GA is applied in [125] to fit the data points with B-Spline surface. GA was first used to find the parameter values of the data points. Then the GA was applied again to find the knot vectors. In addition, DE is also another optimisation method that are being applied widely to optimise the parametric curves and surfaces. DE was applied in [29] to find the optimised control points of the Bézier curve. DE was applied to optimise the control points of the NURBS surfaces in [28]. Besides, PSO was applied to obtain a suitable parameterisation of the data points for Bézier surface reconstruction in [31]. PSO was applied to determine the optimal location of knots in B-Spline curve [80]. PSO was used in the surface fitting of NURBS to obtain the control points and their weights, parametric values of the data points and knot vectors without any pre- or post-processing [30]. Hence, GA, DE and PSO can be applied in solving surface reconstruction case studies.

2.6 Summary

Based on the discussions in the previous sections, it is noticed that the process in surface reconstruction begins with the acquisition of the data in which the data can be in the form of structured or unstructured and ends with the generation of a surface that represent the data. When the data is in unstructured form, reorganisation of the data is required to regain the connectivity information among the data so that the original shape of the object can be recovered. Reorganisation of the data can be performed with self-organising map (SOM) models. However, the models have their own limitations.

It is also noticed that the explicit surface tends to represent the data incorrectly when the data has noise. In addition, the implicit surface tends to overfit the data and the deep learning methods are also suffering with limitation such as fails to generate the correct surface. Parametric surface is commonly-used in the real-world applications due to its flexibility and ability to represent the shape of an object well. It is discovered that the output of the SOM model can be represented with the parametric surface and this would make the SOM model to be useful in the field of computer aided geometric design (CAGD). Although the parametric surface can be used to represent the output of the SOM model using the mathematical formula, the accuracy of the parametric surface is not optimal because it is generated with control points, weights, parameters or knot vectors that are not optimal. To overcome the problem, the free-form parametric surfaces can be optimised with GA, DE and PSO either through the optimisation of control points, weights, parameters and knot vectors.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Overview

This chapter discusses the methodology of the research and provides a general flow of the research beginning with the literature review and problem definition, data collection and definition, organise the unstructured data using Double Net Self-Organising Map (DNSOM) model, integrate the improved NURBS surface approximation approach on the DNSOM model, optimise the improved NURBS surface approximation approach with optimisation techniques, and documentation.

3.2 Research Framework

This section discusses the research framework and provide a general information about the tasks involved in this research. Figure 3.1 illustrates the research framework of this research.

3.2.1 Literature Review and Problem Definition

Initially, a thorough literature review was conducted to gain a good understanding on the existing research and to identify the unexplored areas related to the research title. Literature review also helps to build the knowledge on the research title. The theories, concepts and previous works of SOM, free-form parametric curves and surfaces such as B-Spline and NURBS, and optimisation techniques such as Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimisation (PSO) related to surface reconstruction were studied. The problem statement of this research was defined based on the findings of the literature review. Upon the definition of the problem statement, three objectives were determined to overcome the defined problems.

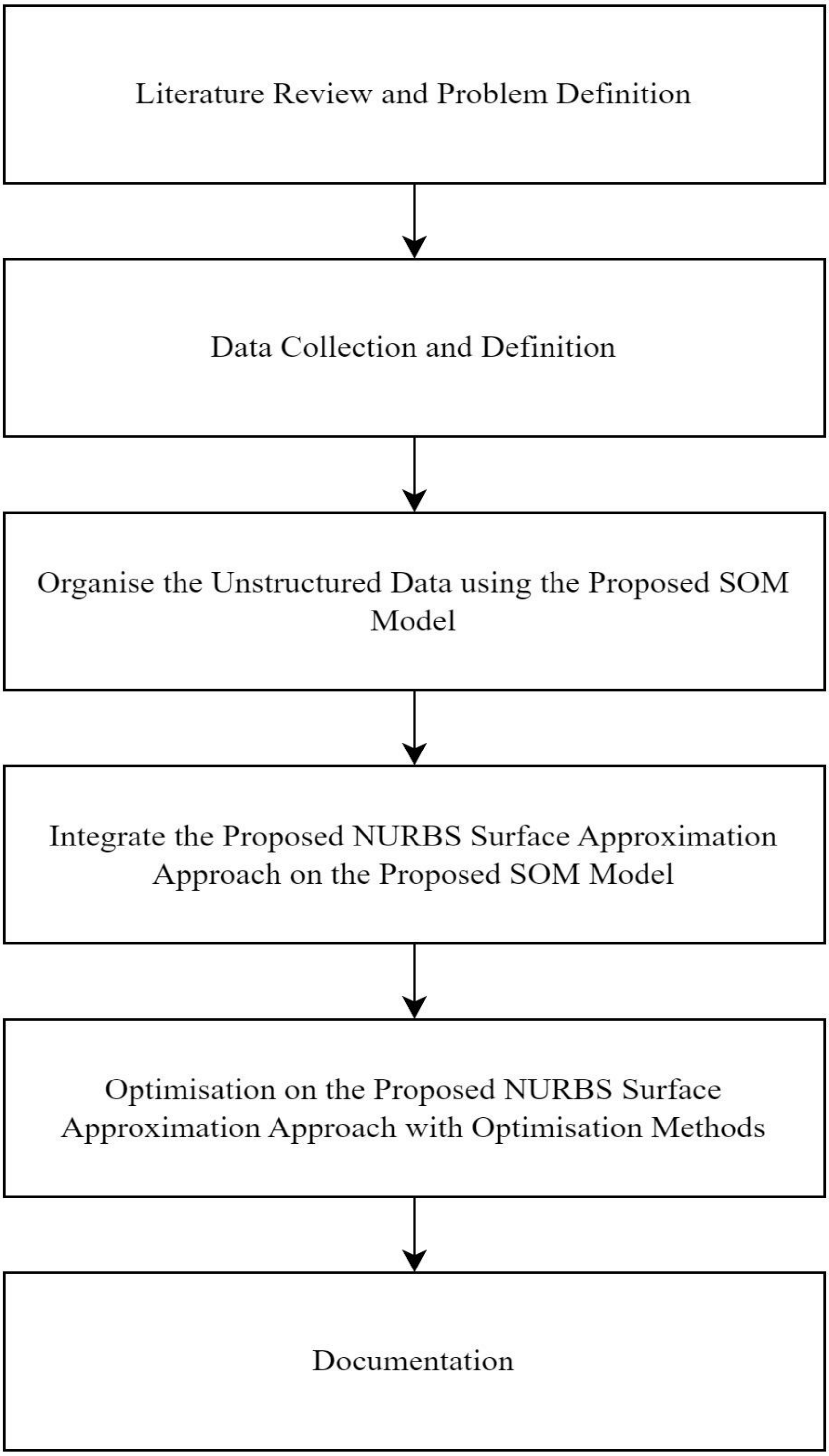


Figure 3.1: Research Framework

3.2.2 Data Collection and Definition

After performing a thorough literature review, identifying the problem and determining the objectives of the research, data collection was performed. The data required are collected and their properties were defined. For the data to be applicable in this research, the data collected must be in coordinate (x, y, z) and unstructured form. The data collected includes four primitive shapes [25] and a medical data [45]. The four primitive shapes were the cube, sphere, spindle and oiltank data, and the medical data was the talus bone data. The general information about the data and the visualisation of the data were provided in Table 3.1 and Table 3.2 respectively. The cube, sphere, spindle, oiltank and talus bone have 7352, 7082, 7552, 5942 and 5253 data points respectively. The data were visualised with GNUPlot and they were demonstrated in Table 3.1. The data show that they are in unstructured form. The data were normalised and the first five coordinates of each data were included in Table 3.1. Figure 3.2 is the superior view of the talus bone and it was adapted from [126]. The figure was included to demonstrate the structure of the talus bone. Additional data which is the Stanford bunny data with 35947 data points were used to test the performance of the DNSOM model and the visualisation of the data was shown in Table 3.2.

3.2.3 Organise the Unstructured Data using the DNSOM Model

Organise the unstructured data using the DNSOM model was the first objective of this research. The motivation of this objective was to develop a model using two 2-D Self-Organising Map (SOM) to form the DNSOM model in which the model was able to overcome the limitation of the 2-D SOM and 3-D SOM in organising the unstructured data. At the same time, the model was able to organise the unstructured data with fewer number of neurons compared to the Cube Kohonen SOM (CKSOM) model proposed in [25]. This was achievable because only two 2-D SOMs were used to develop the DNSOM model and the model allows the setting of different grid size. Data reduction is one of the important properties of the SOM.

Table 3.1: Data information

Data	No. of data points	*x*	*y*	*z*
Cube	7352	0.000000	0.000000	1.000000
		0.000000	0.000000	0.971430
		0.028580	0.000000	0.971430
		0.028580	0.000000	1.000000
		0.057150	0.000000	0.971430
Sphere	7082	0.500000	1.000000	0.500000
		0.500000	0.999315	0.473830
		0.498630	0.999315	0.473865
		0.497265	0.999315	0.473975
		0.495905	0.999315	0.474155
Spindle	7552	0.500000	1.000000	0.500000
		0.500167	1.000000	0.500000
		0.500165	1.000000	0.499978
		0.500162	1.000000	0.499958
		0.500155	1.000000	0.499938
Oiltank	5942	0.500000	1.000000	0.500000
		0.528970	0.999462	0.500000
		0.528810	0.999462	0.496973
		0.528335	0.999462	0.493978
		0.527550	0.999462	0.491050
Talus bone	5253	0.868386	0.673464	0.000000
		0.830361	0.637315	0.000600
		0.883370	0.678238	0.000817
		0.862615	0.658639	0.001184
		0.830930	0.653490	0.001922
Stanford bunny	35947	0.365193	0.615243	0.549817
		0.320558	0.621379	0.528523
		0.171359	0.766240	0.820968
		0.593469	0.629562	0.705158
		0.462973	0.607046	0.572036

Table 3.2: Y-X projection view of all data collected

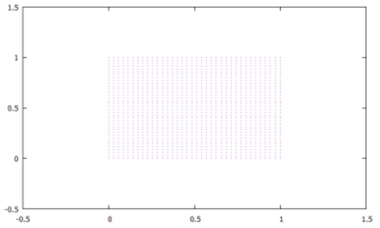
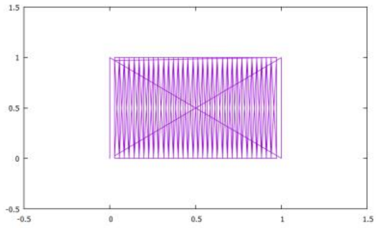
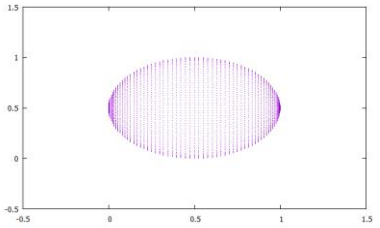
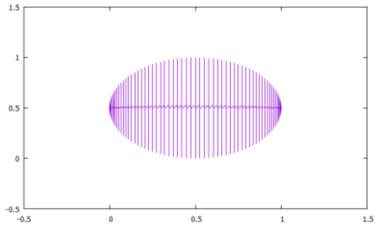
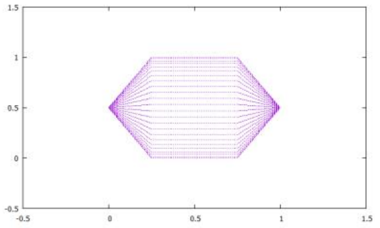
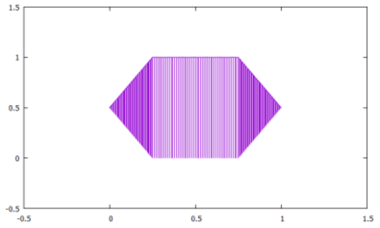
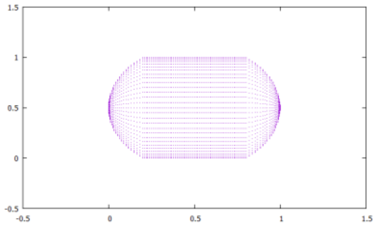
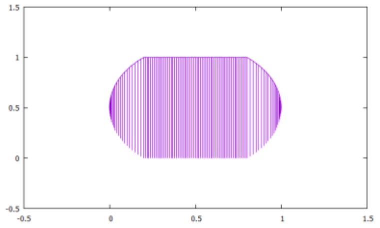
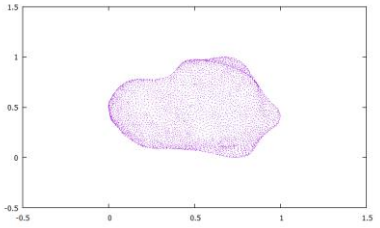
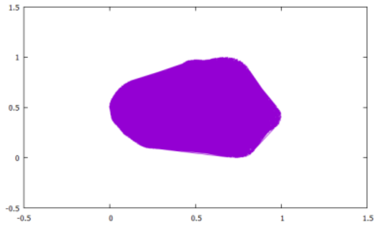
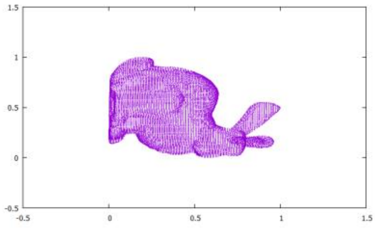
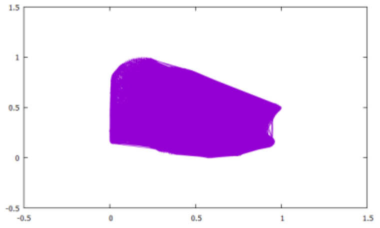
Data	Points	Lines
Cube		
Sphere		
Spindle		
Oiltank		
Talus Bone		
Stanford bunny		

Figure 3.2: Superior view of talus bone. Adapted from [126]

Five performance measurements were used to compare the performance among the models, which are minimum and maximum error, Quantisation Error (QE), Topographic Error (TE) and CPU time. Minimum error, maximum error and CPU time were recommended in [25]. Meanwhile, the QE was applied in [25], [127]. The TE was used in [127]. The output of the DNSOM model was structured and it is known as the DNSOM surfaces data. The output of the model was used as the input of the second objective.

3.2.4 Integrate the Improved NURBS Surface Approximation Approach on the DNSOM Model

Integrate the DNSOM model with improved NURBS surface approximation approach is the second objective of this research. The output of the DNSOM model was used as the input of this objective and they include the output of the model for the cube, sphere, talus bone, oiltank and spindle data. All outputs were structured 3-D data points. Although the data are structured, it is not the standard representation in Computer-Aided Geometry Design (CAGD) industries [23]. Therefore, the model cannot be applied in CAGD directly. To overcome this challenge, free-form parametric surfaces such as NURBS are used to represent the output of a model and they are the common standard representation in CAGD. Hence, NURBS surfaces were used to represent the DNSOM surface data. The conventional NURBS surface approximation approach from [23] was applied on each 2-D SOM separately with different parameterisation methods and sizes of control net (CN). Consequently, the NURBS surfaces were generated. Besides, improvements were made to the conventional NURBS surface approximation approach and it is known as improved NURBS surface approximation approach. The improved approach was applied on all the output of the model with different parameterisation methods and sizes of CN. The performance measurement involved in this objective is known as surface error and it was derived based on the DNSOM model and Euclidean distance. The outputs of both approaches for each of the data include the basis function, NURBS surfaces data and surface error.

3.2.5 Optimisation on the Improved NURBS Surface Approximation Approach

Optimisation techniques were proposed to optimise the improved NURBS surface approximation approach aiming to achieve smaller surface error between the improved NURBS surface data and the DNSOM surface data, which is the third objective of this research. It can be achieved by optimising the control points of the improved approach. The quantitative and qualitative performance measurement used in this objective were optimised surface error between the optimised improved NURBS surfaces data and the DNSOM surface data, and visualisation respectively. The surface error was utilised to evaluate the performance between the optimisation techniques and it was computed based on Euclidean distance and the DNSOM surface data. Besides, visualisation was used to evaluate and to compare the optimised improved

NURBS surfaces with the improved NURBS surfaces from the second objective, aiming to identify the differences between the improved NURBS surfaces before and after optimisation. The inputs for this objective include the DNSOM surface data, the basis functions, control points and NURBS surface data of the improved NURBS surface approximation approach. Each of the optimisation technique was applied on various parameterisation methods, sizes of CN and data. The outputs of this objective were the optimised control points, optimised surface error, CPU time and optimised improved NURBS surfaces data.

3.2.6 Documentation

Documentation is the final and important step in this research. The system flow of each objective was documented accordingly alongside with their outcomes and findings.

3.3 Hardware and Software Requirements

The hardware used in this research is a desktop with Intel ® Core ™ i7-7700K CPU @ 4.20GHz, NVIDIA GeForce GTX 1050 2GB and 32 GB RAM. Meanwhile, the software used in this research were Microsoft Visual Studio 2022 with C++ programming and GNUPlot. Microsoft Visual Studio 2022 with C++ programming was used to code the models in first objective, the surface approximation approaches in second objective and the optimisation techniques in the third objective, and to perform every experiment in this research. Besides, visualisation was conducted with GNUPlot.

3.4 Summary

The theories and concepts of SOM models, B-Spline and NURBS curves and surfaces as well as GA, DE and PSO were studied when conducting the literature review. After conducting the literature review, it was noticed that the 2-D SOM, 3-D SOM and CKSOM models are still suffering from limitation and the output of the models are not a standard representation in CAGD. The output of the model can be represented with NURBS surfaces but gaps were identified when the output of the model were represented with the NURBS surfaces separately. Additionally, the NURBS surfaces generated may not have a high accuracy. After defining the problems, data were collected and defined. The data collected were in coordinates (x, y, z) and unstructured form. The unstructured data were organised with the DNSOM model. The improved NURBS surface approximation approach was applied on the output of the DNSOM model to generate the NURBS surfaces without gaps and optimisation techniques were used to optimise the improved NURBS surfaces. Lastly, the system flow, findings and discussions involved were documented.

CHAPTER 4

THE DOUBLE NET SELF-ORGANISING MAP (DNSOM) MODEL

4.1 Overview

As mentioned before, 2-D SOM model has holes problem, 3-D SOM model has connectivity problem and CKSOM model has problem settings its grid size with different length and width. Hence, this chapter proposes a model to organise the unstructured data and addresses the limitations of the models. The model was formed through the merging of two 2-D SOMs and it was inspired by Lim and Haron [25]. The proposed model is known as Double Net Self-Organising Map (DNSOM). Acquiring Data, Initialising Parameters, Merging Neurons, Detecting Neighbours, Generating Weights, Learning Process and Producing Output are the processes involved in organising the unstructured data with the model. Class Number was used to group the neurons on the border of two views in DNSOM model during Merging Neurons and this would solve the issues of holes because neurons grouped in the same class number were assigned with the same weight vector to perform the learning process. The neuron with the same class number will be updated if it was selected as the winning or neighbouring neurons during the learning process. This chapter also presents the performance of the DNSOM model.

4.2 System Flow of the DNSOM Model

Figure 4.1 shows the steps included in the DNSOM model to overcome the holes problem in 2-D SOM, the connectivity problem in 3-D SOM and the inability to set different grid size in Cube Kohonen SOM (CKSOM). The model was inspired based on the work of Lim and Haron [25] and it was proposed to organise the unstructured data. Three new equations were derived to construct the model.

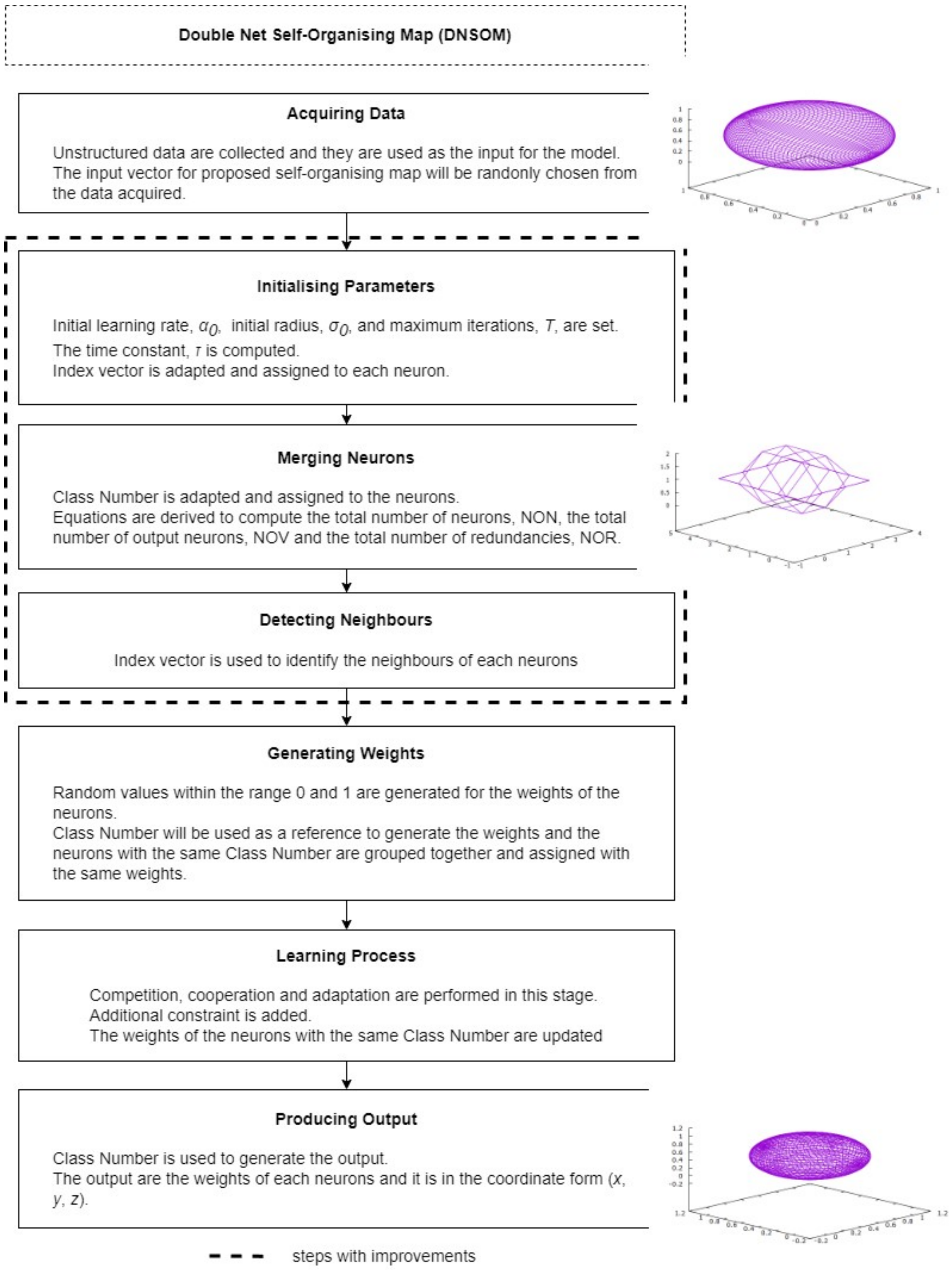


Figure 4.1: Flowchart of the DNSOM model

4.2.1 Acquiring Data

The data applied in this research were described in Research Methodology (Chapter 3) and they include four sets of primitive data (cube, sphere, spindle and oiltank), one set of medical image data (talus bone) and a complex data (Stanford bunny). Generally, all the data are in 3-D coordinates (x, y, z) and unstructured form. The data were inserted into the DNSOM model and they were randomly chosen as the input vector of the DNSOM model in Learning Process (Section 4.1.6).

4.2.2 Initialising Parameters

Table 4.1 shows the parameters initialised for the DNSOM model and they were referred from [25], [98]. The same values of parameters in Table 4.1 were initialised for the 2-D SOM, 3-D SOM and CKSOM models for comparison purposes. Rectangular topology was used for the DNSOM model because the NURBS surfaces uses such topology as input data [77]. Every neuron in the model has a weight vector in which its dimension is identical to that of the input vector or data. Since the data are in 3-D, the dimension of the weight vector was set to 3. Furthermore, this step includes the initialisation of grid size, n, initial learning rate, α_0, initial radius, σ_0 and maximum iterations, T for the model. The time constant, τ was also calculated in this step. The learning rate was used to control the weight of the neurons during the Learning Process (Section 4.2.6) [128] and the initial learning rate was referred from [25]. The radius was used to determine the neighbourhood distance for each winning neuron. The initial radius was set to half of the n and it was referred from [25]. The learning rate and radius would eventually reduce to 0.01 and 1 during the learning process [25]. Maximum iteration, T

was the total number of iterations for the model in learning towards the input vector [25]. The time constant, τ was applied to handle the decay rate of the learning rate and the radius. Its value was derived with the maximum iterations and the logarithm of initial radius, $\log(\sigma_0)$, and it was recommended by [98]. The value of n was assigned to the width, n_x and length, n_y of the grid for the DNSOM model. When different values of n_x and n_y were used to set the grid size, the initial radius was set to the half of the minimum value between n_x and n_y and it was proposed in [129].

Table 4.1: Parameters and their respective values

Parameter	Value
Dimension of weight vector	3
Grid size, n	10, 20, 30
Initial Learning Rate, α_0	0.9
Initial Radius, σ_0	Half of the grid size
Maximum iteration, T	30000
Time constant, τ	Derived with Maximum iteration and the Logarithm of the Initial Radius

As mentioned in [25], Index Vector was assigned to each neuron to form the structure of the CKSOM model and to identify the neighbours for each neuron. It was adapted in this research to form the structure of the DNSOM model. The Index Vector consists of three index values (i, j and k). The index values, i and j, start with 0 and increase by 1 until $n_x - 1$ and $n_y - 1$ respectively. Similar to CKSOM model, Index Vector was set from bottom to top and left to right. k for neurons with $i = 0$, $i = n_x - 1$, $j = 0$ and $j = n_y - 1$ was set to 1 as $k = 1$ refers to neurons in both maps that are merged to create a connection between the two maps. The k of the remaining bottom and top neurons was set to 0 and 2 respectively and $k = 0$ refers to the neurons at the bottom view while k = 2 refers to the neurons at the top view. This is to indicate the neurons that are not merged while forming the structure of the DNSOM model and to create a space between the two views for organising the closed surface data. Figure 4.2 shows an example the Index Vectors allocated for the bottom view with the grid size of n_x = 4 and n_y = 5. The bottom and top views were referred to the two 2-D SOMs.

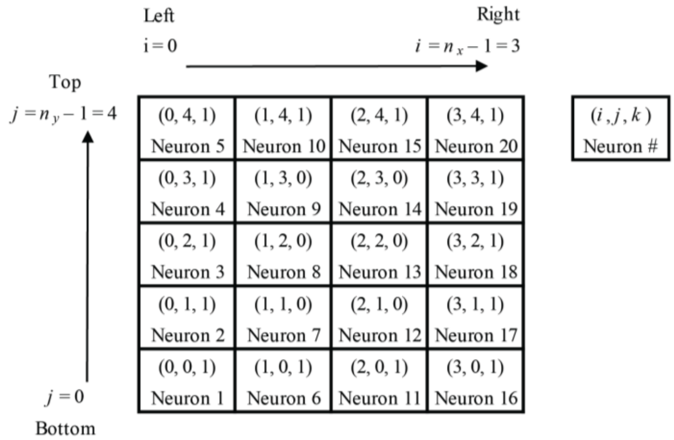


Figure 4.2: Allocation of Index Vectors for the bottom view with grid size, n_x = 4 and n_y = 5

4.2.3 Merging Neurons

Two 2-D SOMs were merged in Merging Neurons (Section 4.2.3) to overcome the issues of 2-D SOM and 3-D SOM models. However, neurons with the same Index Vector would cause redundancy problem during learning process after SOMs were merged. Hence, Class Number was used to group the same neurons and update the weight of the neurons to overcome the redundancy problem. The grid size of the DNSOM model is determined with two parameters which are the width, n_x and length, n_y. Due to different grid sizes, three new equations were derived in this research. Equation 4.1 was formed to compute the total number of neurons (NON) of the DNSOM model.

$$\text{NON} = 2n_x n_y \tag{4.1}$$

Equation 4.2 was formed to compute the total number of output neurons (NOV) involved in the learning process.

$$\text{NOV}=2\left[n_x n_y - n_x - n_y + 2\right] \tag{4.2}$$

The NOV represents the total number of Class Number allocated to group the neurons with similar Index Vector together. It also refers to the number of vertices used to represent the surface of the data. Equation 4.3 was formed to compute the total number of redundancies (NOR) for any grid size.

$$\text{NOR} = 2n_x + 2\left(n_y - 2\right)$$

or

$$\text{NOR} = 2n_y + 2(n_x - 2) \tag{4.3}$$

or

$$\text{NOR} = \text{NON} - \text{NOV}$$

Neurons from the bottom and top views with the same Index Vector were grouped with the same Class Number. The allocation of Class Number based on the Index Vector of each neuron for the DNSOM model with the grid size, n_x = 4 and n_y = 5 and NON = 40 was tabulated in Table 4.2.

Figure 4.3 shows the bottom and top view of the DNSOM model using the Class Number to represent the position of the neurons and the allocation of Class Number in the DNSOM model starts with the bottom view, followed by the top view. Each of the Class Number was allocated sequentially from the bottom to top and left to right. Figure 4.4 shows the bottom and top views of the model in Figure 4.3. Figure 4.5 shows the DNSOM model with the grid size, n_x = 4 and n_y = 5.

Table 4.2: Class Number allocation based on Index Vector for the DNSOM model with grid size, n_x = 4 and n_y = 5, and NON = 40

NON	Index Vector			Class Number	View	NON	Index Vector			Class Number	View
	i	*j*	*k*				*i*	*j*	*k*		
1	0	0	1	1	Bottom	21	0	0	1	1	Top
2	0	1	1	2		22	0	1	1	2	
3	0	2	1	3		23	0	2	1	3	
4	0	3	1	4		24	0	3	1	4	
5	0	4	1	5		25	0	4	1	5	
6	1	0	1	6		26	1	0	1	6	
7	1	1	0	7		27	1	1	2	21	
8	1	2	0	8		28	1	2	2	22	
9	1	3	0	9		29	1	3	2	23	
10	1	4	1	10		30	1	4	1	10	
11	2	0	1	11		31	2	0	1	11	
12	2	1	0	12		32	2	1	2	24	
13	2	2	0	13		33	2	2	2	25	
14	2	3	0	14		34	2	3	2	26	
15	2	4	1	15		35	2	4	1	15	
16	3	0	1	16		36	3	0	1	16	
17	3	1	1	17		37	3	1	1	17	
18	3	2	1	18		38	3	2	1	18	
19	3	3	1	19		39	3	3	1	19	
20	3	4	1	20		40	3	4	1	20	

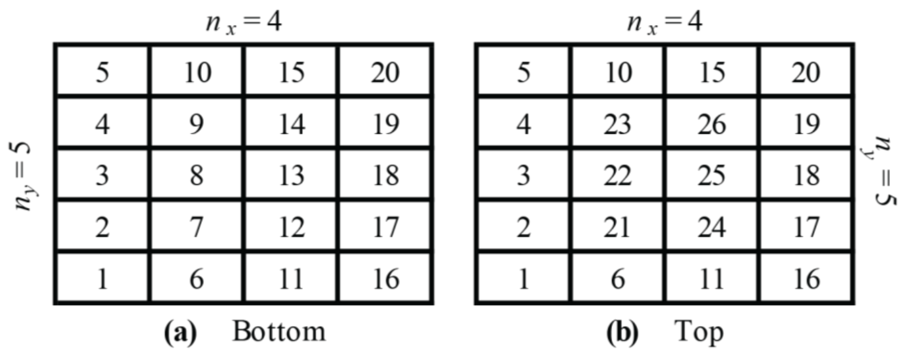
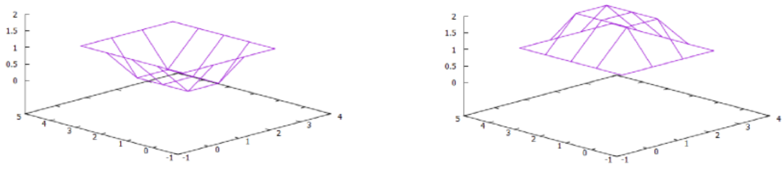


Figure 4.3: Bottom and top views of the DNSOM model with Class Number for grid size, n_x = 4 and n_y = 5

After the process was completed, the Class Numbers and their respective Index Vector were extracted as shown in Table 4.3. Based on the information in Table 4.3, 26 neurons were used in the Learning Process (Section 4.2.6) and they represent the output layer of the model. Nevertheless, all the 40 neurons were used for the Producing Output (Section 4.2.7).



(a) Bottom **(b)** Top

Figure 4.4: Bottom and top views of the DNSOM model with grid size, n_x = 4 and n_y = 5

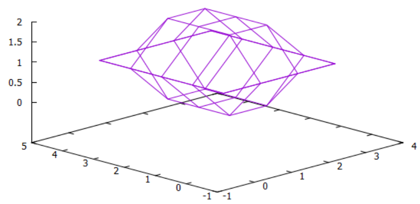


Figure 4.5: The DNSOM model with grid size, n_x = 4 and n_y = 5 after merging both views from Figure 4.4

4.2.4 Detecting Neighbours

Detecting Neighbours determines the neighbours for each Class Number. Redundant neurons were grouped and represented with a Class Number after Merging Neurons (Section 4.2.3). Due to the use of Class Number to represent the structure of the DNSOM model, the model would have different neighbourhood from the 2-D SOM model. Index Vector was used to identify the neighbouring neurons of each Class Number.

Table 4.3: Extracted Class Number with Index Vector

Class Number	Index Vector			Class Number	Index Vector		
	i	j	k		i	j	k
1	0	0	1	14	2	3	0
2	0	1	1	15	2	4	1
3	0	2	1	16	3	0	1
4	0	3	1	17	3	1	1
5	0	4	1	18	3	2	1
6	1	0	1	19	3	3	1
7	1	1	0	20	3	4	1
8	1	2	0	21	1	1	2
9	1	3	0	22	1	2	2
10	1	4	1	23	1	3	2
11	2	0	1	24	2	1	2
12	2	1	0	25	2	2	2
13	2	2	0	26	2	3	2

Equation 4.4 – 14 were used to detect the neighbouring neurons and they are referred from [25], [93], [98].

$$I = |i_{\mathrm{WCN}} - i_{\mathrm{NCN}}| \tag{4.4}$$

$$J = |j_{\mathrm{WCN}} - j_{\mathrm{NCN}}| \tag{4.5}$$

$$K = |k_{\mathrm{WCN}} - k_{\mathrm{NCN}}| \tag{4.6}$$

$$\mathrm{dist}^2 = (\sqrt{I^2 + J^2 + K^2})^2 \tag{4.7}$$

$$\mathrm{dist} = \sqrt{I^2 + J^2 + K^2} \tag{4.8}$$

where i_{WCN}, j_{WCN} and k_{WCN} are the Index Vector of the winning neuron, i_{NCN}, j_{NCN} and k_{NCN} are the Index Vector of the neighbouring neuron, I, J and K are the distance of index i, j and k between the winning neuron and the neighbouring neuron, dist^2 is the distance used in the learning process of SOM, dist is the Euclidean distance between the winning neuron and neighbouring neuron computed using their respective Index Vector.

Additional condition was added to determine the neighbours of each Class Number. After the winning neuron was identified, the distance between the winning neuron and a neighbouring neuron was computed with Equation 4.8. The weight of the neighbouring neuron was updated if the dist for the neuron is smaller than the radius decay and its value of K is not equal to 2 and the value of the dist^2 was substituted into Equation 4.11, Section 4.2.6 for that specific neuron. Neighbouring neuron with dist smaller than the radius decay indicates that the neurons is a valid neighbour of the winning neuron since it is inside the neighbourhood radius of the winning neuron [93]. Meanwhile, the valid neighbouring neuron of the winning neuron must have the value of K not equal to 2 in order to update its weight. This is to ensure that the neighbouring neuron is not located at the opposite view. If the weight of the neurons located at opposite view of the winning neuron was updated, the surface generated will contain gaps. Table 4.4 shows an example of a winning neuron, Class Number 7 was used to calculate its distances to each Class Number using Equations 4.4 – 4.7. When the Class Number 7 is selected as the winning neuron, the weight of the neighbouring neurons with Class Number 21 – 26 will not be updated although the dist for the neurons is smaller than the radius decay because the value of K is equal to 2. This condition is applied to avoid the weight of the neurons from the top view to be updated when the winning neuron is from the bottom view. Conversely, when the winning neuron is from the top view, the weights of the neurons from the bottom view will not be updated. With this condition, the correct surface can be generated.

Table 4.4: Distances between Class Number 7 and each Class Number

Class Number	Class Number 7 $i = 1, j = 1, k = 0$				Class Number	Class Number 7 $i = 1, j = 1, k = 0$			
	I	J	K	dist^2		I	J	K	dist^2
1	1	1	1	3	14	1	2	0	5
2	1	0	1	2	15	1	3	1	11
3	1	1	1	3	16	2	1	1	6
4	1	2	1	6	17	2	0	1	5
5	1	3	1	11	18	2	1	1	6
6	0	1	1	2	19	2	2	1	9
7	0	0	0	0	20	2	3	1	14
8	0	1	0	1	21	0	0	2	4
9	0	2	0	4	22	0	1	2	5
10	0	3	1	10	23	0	2	2	8
11	1	1	1	3	24	1	0	2	5
12	1	0	0	1	25	1	1	2	6
13	1	1	0	2	26	1	2	2	9

4.2.5 Generating Weights

Weights for each neuron, W, were generated with the random values ranging from 0 to 1. Same weights were assigned to the neurons with the same Class Number. The neurons represent the vertices of the model and the Index Vector represents the position of each the neurons. Meanwhile, the weights were depicted as 3-D coordinates (x, y, z). The weights were utilised to fit the input vector and produce the final output after the Learning Process (Section 4.2.6).

4.2.6 Learning Process

Learning process begins after the generation of weights and completes with the production of output. Figure 4.6 shows a flowchart summarising the learning process.

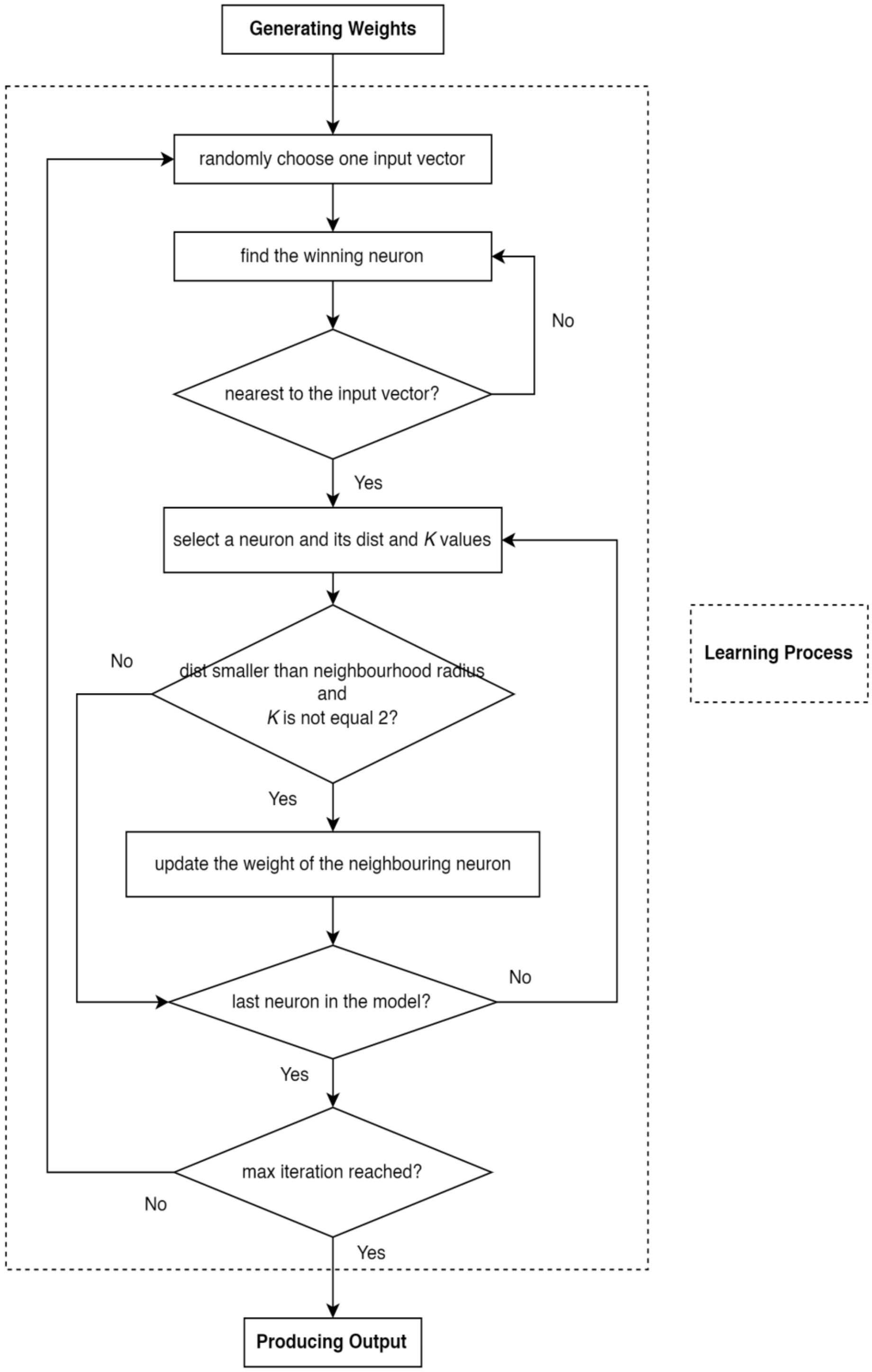


Figure 4.6: Learning Process flowchart

The phases included in the learning process are competition, cooperation and adaptation. One input vector X was chosen randomly from the data at the competition phase. The neuron with the least Euclidean distance to

X was chosen as the winning neuron and the Euclidean distance was calculated with Equation 4.9.

$$d_j = \sqrt{\sum_{i=0}^{i=2} \left(X_i(t) - W_{ij}(t)\right)^2} \tag{4.9}$$

where W_{ij} is the weight connecting the ith element in the input vector and jth neuron, X_i is the input vector, d_j is the Euclidean distance and t is the iteration. The weights of the winning neuron and its neighbouring neurons were updated with the Gaussian function (Equations 4.10 – 4.14) at the cooperation and adaptation phases.

$$W(t+1) = W(t) + G_f(X(t) - W(t)) \tag{4.10}$$

$$G_f = \alpha(t)\exp\left(-\frac{\mathrm{dist}^2}{2\sigma^2(t)}\right), t = 1,2,3,\cdots,T \tag{4.11}$$

$$\alpha(t) = \alpha_0\exp\left(-\frac{t}{\tau}\right), t = 1, 2, 3, \cdots, T \tag{4.12}$$

$$\sigma(t) = \sigma_0\exp\left(-\frac{t}{\tau}\right), t = 1, 2, 3, \cdots, T \tag{4.13}$$

$$\tau = \frac{T}{\log(\sigma_0)} \tag{4.14}$$

where W is the neuron weights, X is the input vector, G_f is the Gaussian function, dist^2 is the distance between winning neuron and the neighbouring neuron defined in Equation 4.7, α is the learning rate at t iteration, α_0 is the initial radius, T is the total number of iterations, t is the current iteration, and τ is the time constant.

When the weight of a neuron~~s~~ was updated, the weight of the neurons with the same Class Number was assigned with the same weight. However, the dist for the neuron must be smaller than the radius decay or the neighbourhood radius of the winning neurons at t iteration, and the value of K is not equal to 2 in order to update its weight. The first condition was applied in the learning process of SOM to filter the invalid neighbouring neurons and to find the neighbouring neurons of the winning neuron [93]. Invalid neighbouring neurons are neurons having larger dist than the radius decay. The weight of the invalid neighbouring neurons would not be updated since they were outside the radius of the winning neuron. Generally, the weight of the valid neighbouring neurons was updated once the condition was fulfilled since they fall inside the neighbourhood radius of the winning neuron [93]. But the condition was insufficient to update the weight of the valid neighbouring neurons in the DNSOM model. The second condition was applied uniquely in the DNSOM model to prevent the weight of the neurons from the top view to be updated when a neuron from the bottom view was chosen as the winning neurons or vice versa. This is to prevent the neurons from the top view from learning towards the winning neuron of the bottom view or vice versa. The surface generated would be incorrect if the second condition is not applied. The learning process terminated when the maximum iteration was achieved. Then the process was continued with Producing Output (Section 4.2.7).

4.2.7 Producing Output

The final output was generated after the Learning Process (Section 4.2.6) was completed. The final output of the DNSOM model is 3-D structured data and it was the weight of every neuron (x, y, z) which was outputted based on the Class Number.

4.3 Analysis and Discussion

This section analyses and discusses the performance between the 2-D SOM, 3-D SOM, CKSOM and the DNSOM models based on the results obtained. Table 4.5 shows the minimum and maximum Euclidean distance for the spindle data and SOM models. Table 4.6 shows the Quantisation Error (QE) and Topographic Error (TE) for the spindle data and SOM models. Meanwhile, Table 4.7 shows the CPU time for spindle data and SOM models. Table 4.8 shows the visualisation for spindle data and SOM models. The evaluation metrics and visualisation for the remaining data and SOM models were included in Appendix A and Appendix B respectively.

Table 4.5: Minimum and maximum Euclidean distance for spindle data and SOM models

Grid Size, *n*	Min Error				Max Error			
	2-D SOM	3-D SOM	CKSOM	DNSOM	2-D SOM	3-D SOM	CKSOM	DNSOM
10	0.004991	0.001225	0.001017	0.002142	0.750138	0.459577	0.453888	0.612911
20	0.001050	0.000085	0.000030	0.001004	0.651950	0.436344	0.425152	0.528028
30	0.000675	0.000025	0.000024	0.000059	0.636605	0.402486	0.401306	0.505220

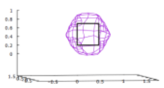
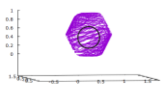
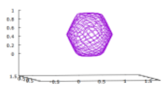
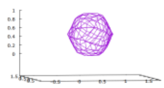
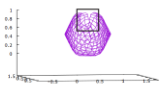
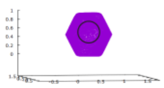
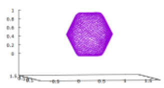
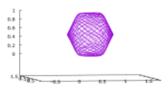
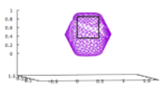
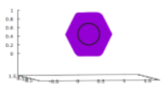
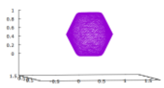
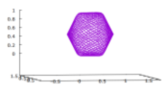
Table 4.6: QE and TE for spindle data and SOM models

Grid Size, *n*	QE				TE			
	2-D SOM	3-D SOM	CKSOM	DNSOM	2-D SOM	3-D SOM	CKSOM	DNSOM
10	0.142370	0.057148	0.061301	0.104913	0.239433	0.598300	0.257233	0.232033
20	0.096793	0.032491	0.037149	0.065414	0.239167	0.623033	0.244967	0.218333
30	0.079069	0.025129	0.029380	0.052135	0.225433	0.674800	0.240133	0.214100

Table 4.7: CPU time for spindle data and SOM models

Grid Size, *n*	CPU Time (s)			
	2-D SOM	3-D SOM	CKSOM	DNSOM
10	0.062082	2.822090	1.747945	0.505617
20	1.329022	26.724297	7.936086	2.241000
30	3.365511	189.238892	19.378843	5.309147

Table 4.8: Visualisation for spindle data and SOM models

Grid Size, *n*	2-D SOM	3-D SOM	CKSOM	DNSOM
10				
20				
30				

Five evaluation metrics such as minimum and maximum errors, QE, TE and CPU time were used to measure the performance of the SOM models. The QE is an evaluation metric utilised to evaluate the accuracy of the SOM models and it was applied in [25], [130]. QE measures the average distance between the winning neuron and the input vector [131]. As for the TE, it is used to measure how good the structure of the inputs is modelled by the model [95]. This metric was used in [95], [131]. QE and TE are derived in Equation 4.15 and Equation 4.16:

$$QE = \frac{1}{T}\sum_{t=1}^{T} \|X(t) - W_c(t)\| \tag{4.15}$$

where $X(t)$ is the input data at the iteration t, $W_c(t)$ is the winning neuron's weight vector of input data $X(t)$ and T is the maximum number of iterations.

$$TE = \frac{1}{T}\sum_{t=1}^{T} d(X(t)) \tag{4.16}$$

where $X(t)$ is the input data at iteration t, if the first winning neuron and the second winning neuron of $x(t)$ is not adjacent, $d(X(t)) = 1$ and vice versa, $d(X(t)) = 0$, T is maximum number of iterations.

Besides, CPU time is used to demonstrate the speed of the models in generating the output and it was suggested in [25]. Meanwhile, visualisation was used to visualise the surface of the models and it was also utilised in [25]. Table 4.9 shows the number of neurons involved in every SOM model and the total number of neurons or vertices used to represent the surface for the respective models and the grid sizes are highlighted in bold.

Table 4.9: Total output neurons representing the surface

Grid Size, *n*	2-D SOM	3-D SOM	CKSOM			DNSOM		
	$n \times n$	$n \times n \times n$	NON	NOR	NOV	NON	NOR	NOV
10	**100**	**1000**	600	112	**488**	200	36	**164**
20	**400**	**8000**	2400	232	**2168**	800	76	**724**
30	**900**	**27000**	5400	352	**5048**	1800	116	**1684**

The minimum errors in Table 4.5 decrease when the grid size increases shows that the winning neurons can move closer towards the input data. As shown from the results in Table 4.6 and Table 4.7, the QE was reduced and the CPU time increased for every data set when the grid size was increased. The findings are aligned with the findings in [25]. Furthermore, the TE of the model decreases as the grid size increases. The higher the TE, the weaker the model in preserving the topology of the data. Besides, more vertices were used to represent the surface when the grid size increased. When the number of vertices representing the surface increases, the surface becomes smoother.

According to Table 4.6 and Table 4.7, 2-D SOM model had the highest QE and shortest CPU time. The high QE value shows that it had a low accuracy as the winning neuron is less fitted towards the input vector. The 2-D SOM model obtained the shortest CPU time because it used the least number of vertices to represent the surface as shown in Table 4.7 and Table 4.9. The model used the least number of vertices to represent the surface because it is a single map unlike the 3-D SOM, CKSOM models and the DNSOM model which are made up of *n*, six and two 2-D SOMs respectively and each of the map has the identical grid size of $n \times n$. Despite the ability of the 2-D SOM model to generate the outputs faster, it failed to reconstruct the surface of the closed surface data due to the absence of connectivity information between the neurons at the boundary [25], [42], [43] and the surface generated still contains holes as marked by the square in Figure 4.7. Thus, the 2-D SOM is the most underperforming SOM model when compared to others.

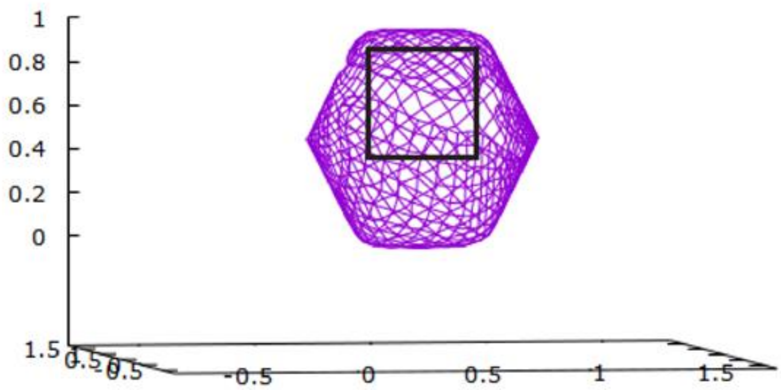


Figure 4.7: Visualisation for the 2-D SOM model with grid size, *n* = 30 and spindle data retrieved from Appendix B. The surface generated contains holes as marked by the square.

As for the 3-D SOM model, it had the lowest QE, highest TE and CPU time. As shown in Table 4.9, the 3-D SOM model used the highest number of vertices to represent the surface. This is because it used $n \times n \times n$ neurons to represent the surface unlike the CKSOM model and the DNSOM model. The CKSOM model and the DNSOM model used six and two $n \times n$ 2-D SOMs for their structure respectively and not all the neurons were used to represent the surface for both models. Meanwhile, they used the Class Number where only the distinct neurons the Class Number where only the distinct neurons were used to represent the surface. Therefore, the 3-D SOM model represented the surface with the greatest number of vertices. In addition, the model took the longest time to generate the output because there were more neurons involved in the training process. As marked by the circle in Figure 4.8, the output of the model remained in unstructured form and incorrect surface was produced because the weights of both the internal neurons were updated during the learning process. Thus, the 3-D SOM model is not suitable for the surface reconstruction of closed surface data.

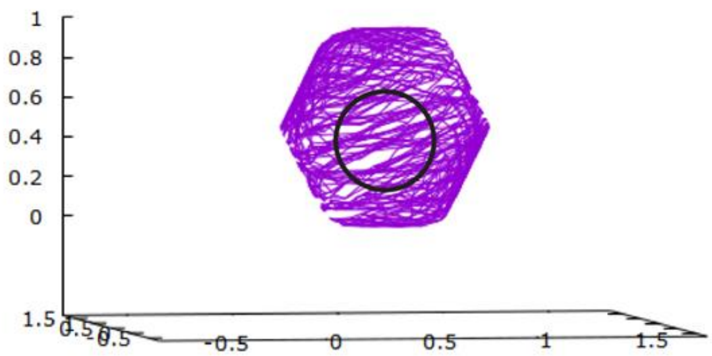


Figure 4.8: Visualisation for the 3-D SOM model with grid size, *n* = 10 and spindle data retrieved from Appendix B. The surface generated contains internal neurons as marked by the circle.

As for the CKSOM model, it had a lower QE compared to the 2-D SOM model, a slightly higher QE compared to the 3-D SOM model and a moderate CPU time. The CKSOM model had a higher accuracy than the 2-D SOM model as it had a lower QE compared to the 2-D SOM model. But it had a lower accuracy than the 3-D SOM model because it had a slightly higher QE compared to the 3-D SOM model. Although the CKSOM model achieved a lower accuracy than the model, it can generate the correct surface as shown in Table 4.8. Hence, the CKSOM model can reconstruct the surface of the closed surface data without holes and without connectivity problem between its neurons which eventually solves the problem of 2-D SOM and 3-D SOM models respectively. However, the model fails to organise the unstructured data and generate the correct surface with different grid sizes because the length and width of its grid are set with the same fixed value, *n*. The discussion can be referred to the additional experiment for CKSOM model with different grid size using the same data set. Furthermore, the results tabulated in Table 4.9 show that the CKSOM model used fewer number of vertices than 3-D SOM model to represent the surface.

As for the DNSOM model, it achieved the lowest TE among the models for all the data and grid sizes. This shows that the DNSOM model can preserve the topology of the data better than the other models. The results presented in Table 4.9 show that the DNSOM model used fewer number of vertices to represent the surface compared to the 3-D SOM and CKSOM models because it uses only two 2-D SOMs as compared to others. Additionally, it could generate output faster than 3-D SOM and CKSOM models since the number of neurons involved in the learning process was fewer than the models. Similar to the CKSOM model, the DNSOM model can solve the problem of 2-D SOM and 3-D SOM models. It can represent the surface without holes and without any connectivity problem among the neurons as shown in Table 4.8. Besides, the DNSOM model was able to organise the unstructured data and generate the correct surface with different grid sizes.

Additional experiment was performed on the CKSOM and DNSOM models to examine their performance. Table 4.10 and Table 4.11 show the metric evaluation of the CKSOM and DNSOM models when different grid sizes were used given the spindle data. The metric evaluation for CKSOM and DNSOM models for the remaining data sets can be found in Appendix C. In contrast, Table 4.12 shows the visualisation of the CKSOM and DNSOM models when different grid sizes were used given the spindle data. The visualisation of CKSOM and DNSOM models for the remaining data sets can be found in Appendix D.

Table 4.10: Minimum and maximum Euclidean distance of CKSOM and DNSOM models with various sizes of width and length of grid given the spindle data

n_x	n_y	Min Error		Max Error	
		CKSOM	DNSOM	CKSOM	DNSOM
10	8	0.001051	0.003025	0.575847	0.556720
20	12	0.000602	0.001107	0.493366	0.534296
18	30	0.000078	0.000270	0.450791	0.463108

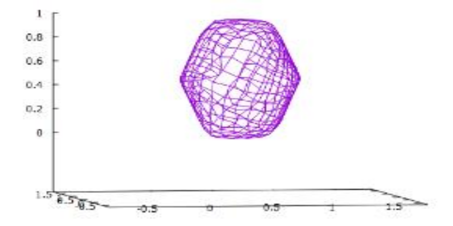
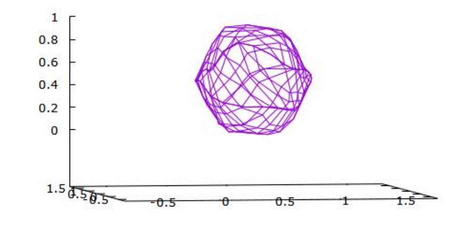
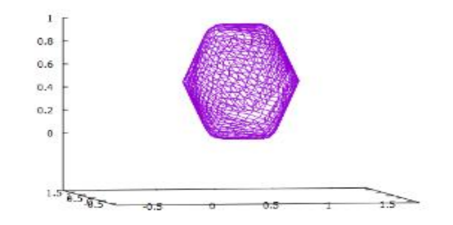
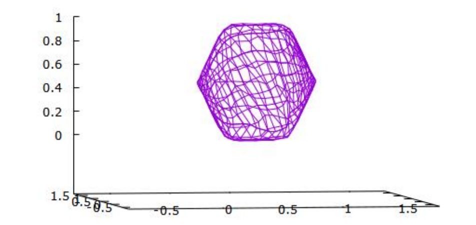
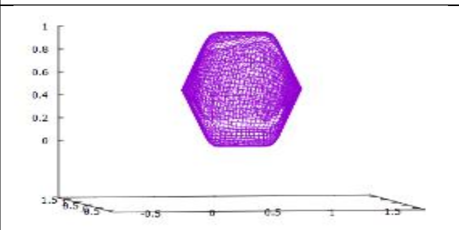
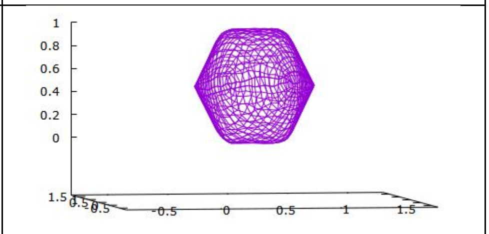
Table 4.13 shows the NON, NOV and NOR of the CKSOM and DNSOM models respectively with different width, n_x and length, n_y of the grid.

The bold font shown in Table 4.13 is the total number of vertices used by the CKSOM and DNSOM models to represent the surface for various n_x and n_y.

Table 4.11: QE, TE and CPU time of CKSOM and DNSOM models with various sizes of width and length of grid given the spindle data

n_x	n_y	QE		TE		CPU Time	
		CKSOM	DNSOM	CKSOM	DNSOM	CKSOM	DNSOM
10	8	0.068216	0.103635	0.354700	0.239133	1.807986	0.428606
20	12	0.043618	0.063878	0.350600	0.227667	5.579232	1.391338
18	30	0.034528	0.048394	0.350400	0.222067	13.621258	3.096171

Table 4.12: Visualisation of CKSOM and DNSOM model for spindle data with different width and length of grid

Data	CKSOM	DNSOM
10 × 8		
20 × 12		
18 × 30		

The DNSOM model was applied on the primitive and medical image data using 10 × 8, 20 × 12 and 18 × 30 grid sizes respectively. Besides, Table 4.13 shows the total number of output neurons used to represent the surface for both CKSOM and DNSOM models given different n_x and n_y. According to the results in Table 4.10 and Table 4.11, the findings of the CKSOM and DNSOM models are aligned to the findings of the models with the same grid size. Therefore, both models contain the same performance although different grid size was applied.

Table 4.13: Total output neurons of the CKSOM and DNSOM model representing the surface for various n_x and n_y

n_x	n_y	CKSOM			DNSOM		
		NON	**NOV**	**NOR**	**NON**	**NOV**	**NOR**
10	8	480	**400**	80	160	**128**	32
20	12	1440	**1308**	132	480	**420**	60
18	30	3240	**3040**	200	1080	**988**	92

Based on the visualisation in Table 4.12, the DNSOM model can represent the surface with different width, n_x and length, n_y of the grid. Meanwhile, the CKSOM model fails to generate the correct surface when different grid sizes were used as shown in Table 4.12 although it achieved lower QE compared to DNSOM model. Figure 4.9 illustrates the surface generated by the CKSOM model contains holes and the holes are highlighted with the rectangles. The holes appeared because the structure of the CKSOM model contains holes at the boundary as highlighted by the rectangle in Figure 4.10. Holes appeared on the CKSOM model because the length and width of the bottom, left and back views were assigned with different values.

In contrast, Figure 4.11 illustrates the surface generated by the DNSOM model without holes.

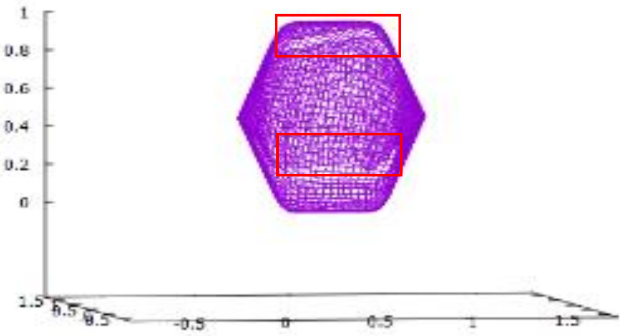


Figure 4.9: Visualisation for the CKSOM model with grid size, n_x = 18 and n_y = 30 given the spindle data

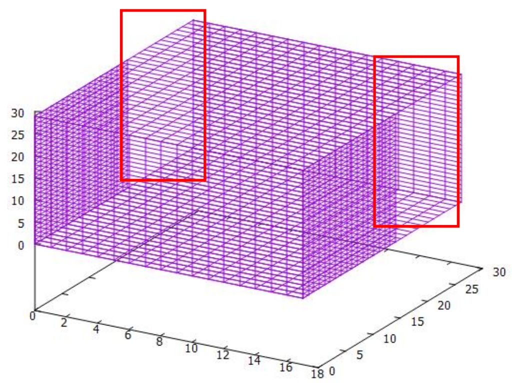


Figure 4.10: The CKSOM model with grid size, n_x = 18 and n_y = 30

This is because the boundary of the 2-D SOMs used to build the structure of the DNSOM model are connected as shown in Figure 4.12. Thus, the DNSOM model has overcome the limitation of the CKSOM model. Besides, smoother surface was generated when the number of vertices used to represent the surface increases.

Apart from testing and validating the DNSOM model in organising the data with different n_x and n_y on primitive shapes, additional experiment was performed to further explore the capability of the DNSOM model in organising complex data such as the Stanford bunny data [46]. The DNSOM model with the grid size, n = 30 was used to organise the Stanford bunny data.

The same initial learning rate, initial radius and maximum iteration from Table 4.1 were applied. The initial radius was set to the half of the minimum value between n_x and n_y and it was proposed in [129].

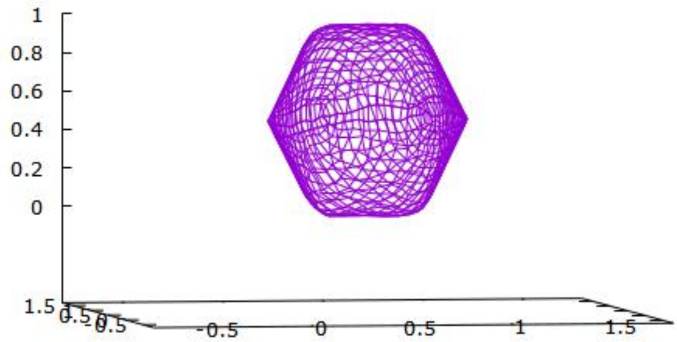


Figure 4.11: Visualisation for the DNSOM model with grid size, n_x = 18 and n_y = 30 given the spindle data

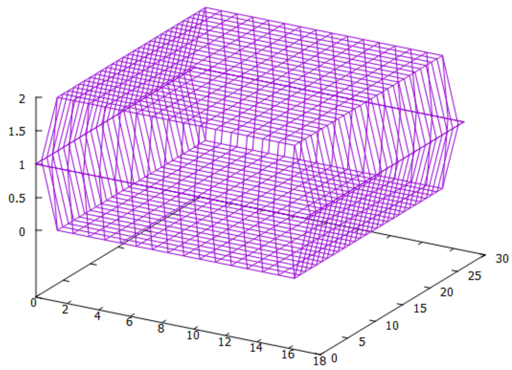


Figure 4.12: The DNSOM model with grid size, n_x = 18 and n_y = 30

Figure 4.13 shows that the DNSOM model failed to generate the ear of the Stanford bunny data correctly as marked by the circle because the winner neurons and their neighbouring neurons were not updated during the learning process. The QE and TE of the experiment were 0.049694 and 0.225400 respectively. The CPU time of the experiment was 5.386289 seconds. It is noticed that the Deep Learning (DL) SOM in [42] also faced the same problem while reconstructing the Stanford bunny data. Nevertheless, both models can reconstruct the overall shape of the Stanford bunny.

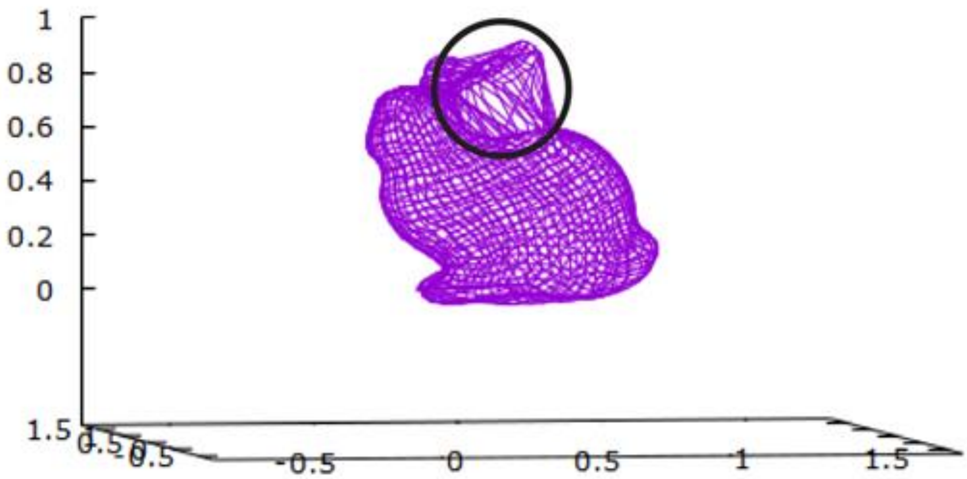


Figure 4.13: Visualisation of Stanford bunny data for DNSOM model with grid size, *n* = 30 showing the incorrect representation of the ear as marked by the circle

Besides, equations were derived to compute the NON, NOV and NOR of the DNSOM model. They were used to generate the model. Different grid sizes were used to verify and validate the equations. As suggested in [25], the area formula (Area = $a \times b$) can be used to calculate the total number of neurons for each map in which a is the length and b is the width of the grid. In this experiment, a was represented as n_y and b was represented as n_x. Thus, when a = 5 and b = 4, then the area of a map would be Area = $a \times b = 5 \times 4 = 20$. Since, two maps were used to create the model, the total number of neurons used to create the model would be $2 \times 5 \times 4 = 40$.

Furthermore, manual calculation can be performed on Figure 4.3 to calculate the NON, NOV and NOR. Each box in Figure 4.3 indicates a neuron and the number in each box indicates the Class Number of each neuron. By counting all the boxes without considering the Class Number, the total number of neurons, NON was obtained. Meanwhile, the total number of output neurons, NOV or the neurons used to represent the surface can be obtained by counting the boxes with distinct Class Number. By counting the redundant Class Number once, the total number of redundancies, NOR was obtained. The redundant neurons would not be used to represent the surface. With the use of the manual calculation, the model in Figure 4.3 obtained the results of NON = 40, NOV = 26 and NOR = 14. The same way was used to calculate the NON, NOV and NOR manually for various n_x and n_y, and the results were tabulated in Table 4.14. Additionally, the NON, NOV and NOR equations were derived using Arithmetic Progression and the derivation is shown in Table 4.15. The equation of NON, NOV and NOR can be proved and validated by comparing the results from Table 4.14 and Table 4.15. The results show that the equations derived are valid and can be used to calculate the NON, NOV and NOR of the DNSOM model when different width, n_x and length, n_y of the grid are set for the model.

Table 4.14: NON, NOV and NOR for various n_x and n_y based on manual calculation

n_x	n_y	**NON**	**NOV**	**NOR**
2	3	12	6	6
3	4	24	14	10
4	5	40	26	14
5	6	60	42	18
6	7	84	62	22
7	8	112	86	26

Table 4.15: Generation of equation of NON, NOV and NOR with Arithmetic Progression

n_x	n_y	A	B	C	$D = B - C$	$E = B + D$
2	3	6 + 6	6	6 + 0	0	6
3	4	12 + 12	12	6 + 4	2	14
4	5	20 + 20	20	8 + 6	6	26
5	6	30 + 30	30	10 + 8	12	42
6	7	42 + 42	42	12 + 10	20	62
7	8	56 + 56	56	14 + 12	30	86
⋮	⋮	⋮	⋮	⋮	⋮	⋮
		$2n_xn_y$	n_xn_y	$2n_x + 2(n_y - 2)$ or $2n_y + 2(n_x - 2)$	$n_xn_y - 2(n_x + n_y - 2)$	$2[n_xn_y - n_x - n_y + 2]$

where,

n_x - Width of the grid

n_y - Length of the grid

A - NON

D - Total distinct neurons at top map

B - Total distinct neurons at bottom map

E - NOV

C - NOR

4.4 Summary

A new SOM model was proposed through the merging of two 2-D SOMs. The DNSOM model can organise unstructured data and generate closed surface without holes. It overcomes the holes problem in 2-D SOM, the connectivity problem in 3-D SOM and the grid size problem in CKSOM. As mentioned previously, the output of the DNSOM model cannot be used directly as the standard representation in the field of computer-aided geometric design (CAGD). Previous works have applied the NURBS surface approximation approach on the output of the SOM models, so that the SOM models can be used in the field of CAGD. Previous works have also shown that gaps would appear when the approach was applied on multiple SOM. Hence, Chapter 5 focuses in overcoming the limitation of the NURBS surface approximation approach.

CHAPTER 5

THE IMPROVED NURBS SURFACE APPROXIMATION APPROACH

5.1 Overview

This chapter proposes a NURBS surface approximation approach to overcome the limitations of the NURBS surface approximation approach on the Double Net Self-Organising Map (DNSOM) model. In [23], the NURBS surface approximation approach was applied on the output of the Cube Kohonen Self-Organising Map (CKSOM), a model formed through the merging of six 2-D SOMs. The approach applied NURBS on each of the CKSOM surfaces data separately. It is named as the conventional approach here. Six surfaces data were involved because CKSOM is made up of six 2-D SOMs. Consequently, gaps were discovered when NURBS surface approximation approach was applied on the outputs of CKSOM because the surface data at the edges of both surfaces do not have the same value. When the conventional approach was applied on the DNSOM surfaces data, the same problem occurred. Therefore, an improved surface approximation approach was proposed to overcome the problem. This chapter also compares the performance between the conventional and improved approaches.

5.2 System Flow of the Improved Surface Approximation Approach

The system flow of the improved NURBS surfaces approximation approach is illustrated in Figure 5.1. The system flow of the improved surface approximation approach was adapted from [23] because it was applied on multiple SOMs. Improvements were made to the perform parameterisation, perform control points calculation and perform surfaces error calculation steps as shown in the dashed boxes in Figure 5.1 in order to generate the NURBS surfaces without gaps.

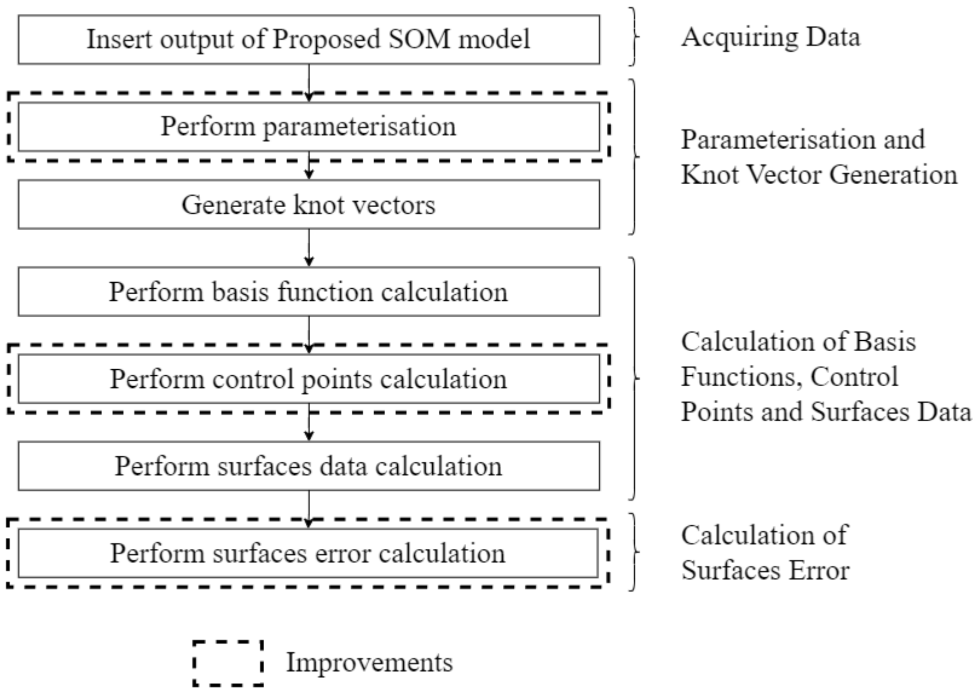


Figure 5.1: Flowchart for the improved NURBS surfaces approximation approach

5.2.1 Acquiring Data

The data used in this approach were the 3-D structured closed surface data of the cube, sphere, spindle, oiltank and talus bone generated by the DNSOM model with the grid size, $n_x = n_y = 20$ and $n_x = 18$, $n_y = 30$. The DNSOM surfaces data with these grid sizes were used because they were large enough to generate a set of control nets (CNs) and they achieved the lowest QE as shown in Table 4.6. The model was constructed using two 2-D Self-Organising Maps (SOMs) to organise the unstructured closed surface data. The data were used in the conventional and the improved NURBS surface approximation approaches. To avoid non-linear problem, the weights, $w_{i,j}$, were set to 1 when applying the conventional and the improved NURBS surface approximation approaches [133]. When $w_{i,j}$ are set to 1, the NURBS surface is reduced to B-Spline surface [134]. As suggested by Kumar, Kalra and Dhande [26], a curve or surface must have at least cubic degree (order 4) to represent generic 3-D entities. Iglesias, Gálvez and Collantes [135] also suggested the use of cubic degree (order 4) because low-degree curve or surface has limited flexibility in controlling its shape while the high-degree curve or surface can cause unwanted wiggles and require more computation. Therefore, cubic degree is used in this research based on these suggestions.

5.2.2 Parameterisation and Knot Vector Generation

Parameterisation and knot vector generation is required in NURBS surface approximation approach to generate the NURBS surface and it was performed on each of the surface data separately to obtain the control points used to generate the NURBS surfaces for each of the data. The steps included in this process are perform parameterisation and generate knot vectors.

5.2.2.1 Perform Parameterisation

Given the DNSOM surface data, parameterisation methods were used to obtain the parameters (*u* and *v*) for each surface. Let U_{si} and V_{sj} be the vectors containing the parameter u in horizontal direction with $n + 1$ columns of the DNSOM surface data and parameter v in vertical direction with $m + 1$ rows of the DNSOM surface data respectively, where n and m are the column and row indexes. Equation 5.1 and Equation 5.2 were adapted from [23], which were the modifications of the equations proposed by Shene [136] by adding the surface number. Two surface numbers were allocated because the DNSOM model is comprised of two 2-D SOMs. The bottom surface was represented with surface number, 1 and the top surface was represented with surface number, 2 because the SOMs are arranged from bottom to top. Parameterisation methods such as Uniform, Chord Length, Centripetal and Exponential methods were used to evaluate their performances. After the parameters (U_{si} and V_{sj}) for each surface were obtained, the average parameters of each row and column for each surface were used to compute the knot vectors.

$$s_{si} = \frac{u_{si,0} + u_{si,1} + u_{si,2} + \cdots + u_{si,n}}{n+1} \tag{5.1}$$

$$r_{sj} = \frac{v_{s0,j} + v_{s1,j} + v_{s2,j} + \cdots + v_{sm,j}}{m+1} \tag{5.2}$$

where s_{si} are the average parameters in the u direction, r_{sj} are average parameters in the v direction, s is the surface number, i is the row and j is the column.

However, the use of average parameters to generate the knot vectors would cause the generation of the NURBS surfaces with gaps at the edges of each surface. The parameters for both surfaces in the u and v directions were standardised with Equation 5.3 and Equation 5.4 so that the control points at the edges of the CNs has the same value. When the control points at the edges of the CNs has the same value, the surface data located at the edges of both surfaces would have the same value too.

$$S_i = \frac{s_{1i} + s_{2i}}{2} \quad (5.3)$$

$$R_j = \frac{r_{1j} + r_{2j}}{2} \quad (5.4)$$

where S_i is the standardised parameters for both surfaces in the u direction, R_j is the standardised parameters for both surfaces in the v direction. S_i were assigned to s_{1i} and s_{2i}, and R_i was assigned to r_{1j} and r_{2j}. The standardised parameters would be used to generate the knot vectors instead of the average parameter for the improved approach in this research.

5.2.2.2 Generate Knot Vectors

After the parameter values were obtained via the parameterisation methods, the averaging knot vector method was used to generate the knot values of each surface. The method was suggested by Jiang and Wang [134], Forkan and Shamsuddin [32], Lim and Haron [23], Makhlouf, Elloumi, Louhichi and Deneux [137] and adapted from Shene [136]. The method was used in this research because the knot vector can be generated with Equations 2.9 – 2.11 in Section 2.3.2. The equations were applied to the standardised parameters, S and R to generate the knot vectors for each surface. Since the standardised parameters were used in the generation of knot vectors for both surfaces, both surfaces would have same knot vectors.

5.2.3 Calculation of Basis Functions, Control Points and Surfaces Data

5.2.3.1 Perform Basis Function Calculation

The basis function (N_u and N_v) for each surface was calculated after the generation of knot vectors. The basis function is required to obtain the control points in the next step. The basis function of each surface was computed with Equation 2.3 from Chapter 2, Section 2.3.2.

5.2.3.2 Perform Control Points Calculation

The calculation of control points for each surface was performed according to their basis function (N_u and N_v). The equation is referred from Sarfraz and Riyazuddin [138], and Zhang, Feng and Cui [90], and adapted from Lim and Haron [23].

$$N_u P N_v = D \tag{5.5}$$

$$N_u^T N_u P N_v N_v^T = N_u^T D N_v^T$$

$$N_u' P N_v' = N_u^T D N_v^T$$

$$N_u'^{-1} N_u' P N_v' N_v'^{-1} = N_u'^{-1} N_u^T D N_v^T N_v'^{-1}$$

$$P = N_u'^{-1} N_u^T D N_v^T N_v'^{-1} \tag{5.6}$$

where N_u and N_v are the basis function for each surface, P is the control point, D is the DNSOM model closed surface data, $N_u{}^T$ and $N_v{}^T$ is the transpose basis function, N_u' is the product of $N_u{}^T$ and N_u, $N_v{}'$ is the product of N_v and $N_v{}^T$, $N_u{}'^{-1}$ is the inverse of $N_u{}'$ and $N_v{}'^{-1}$ is the inverse of $N_v{}'$.

Equation 5.5 was used in the improved approach to compute the control points of each surface. The computation of control points for each surface was conducted separately because two DNSOM surfaces data were involved. After the computation of the control points for both surface, different coordinates were produced. Hence, the concept of Index Vector and Class Number from Chapter 4 were applied to group the control points at the edges of the CNs before the standardisation of the control points. Index Vector are allocated to both CNs. Figure 5.2 shows the Index Vector and Class

Number of each control point for the bottom and top CNs given the $CN_x = 3$ and $CN_y = 4$ where CN_x and CN_y are the column and row of the CN respectively. The Index Vector of each control point is comprised of the index values, *i* and *j*. Class Number was allocated from bottom to top, left to right of each surface. Each Class Number followed the sequence from bottom to top surface. If $i = 0$ or $i = CN_x - 1$ or $j = 0$ or $j = CN_y - 1$ for the Index Vector of each control point in the top CN, the control point would be grouped into the same Class Number as shown in Figure 5.2. The index values, *i* and *j* of the control points that fulfilled the condition is bold in Figure 5.2.

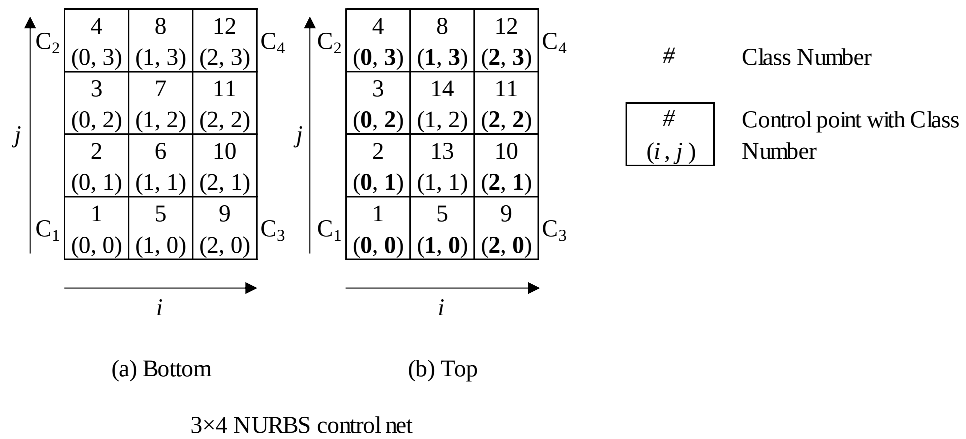


Figure 5.2: Index Vector and Class Number for the control points in the bottom and top CNs given the $CN_x = 3$ and $CN_y = 4$

After computing the control points for each of the surfaces and grouping the control points into their respective Class Number, the control points with the same Class Number are standardised by summing and averaging them accordingly. The standardised control points, P^e_{avg} were assigned and replaced the control points having the same Class Number to close the gaps after computed using Equation 5.7.

$$P_{avg}^{e} = \frac{P_1^e + P_2^e}{2} \tag{5.7}$$

where P_1^e and P_2^e are the control point at the edge of the bottom and top CNs respectively, P_{avg}^{e} is the standardised control point for the control point at the edge of the bottom and top CNs.

Similar to the conventional approach, the control points located at the four corners of bottom and top CNs of the improved approach must pass through the points positioned at the four corners of the bottom and top surfaces of the DNSOM model respectively. Class Number from the previous chapter was utilised to obtain the corners of the bottom and top CNs, and the corners of the bottom and top surfaces of the DNSOM surface data. Figure 5.2 shows the bottom and top surfaces of the DNSOM model, and the top and bottom CNs of the improved approach respectively. The red boxes in the figure are the corners of the bottom and top surfaces of the DNSOM surface data and CNs. Meanwhile, the grey and yellow boxes in the figure are the coordinates and control points at the edges of the DNSOM data and CNs respectively. The number in each box in the figure is the Class Number of the coordinates and control points. The D_1, D_2, D_3 and D_4 in Figure 5.3 are the points located at the four corners of the bottom and top surfaces data of the DNSOM model. Besides, C_1, C_2, C_3 and C_4 in Figure 5.3 are the control points located at the four corners of the bottom and top CNs.

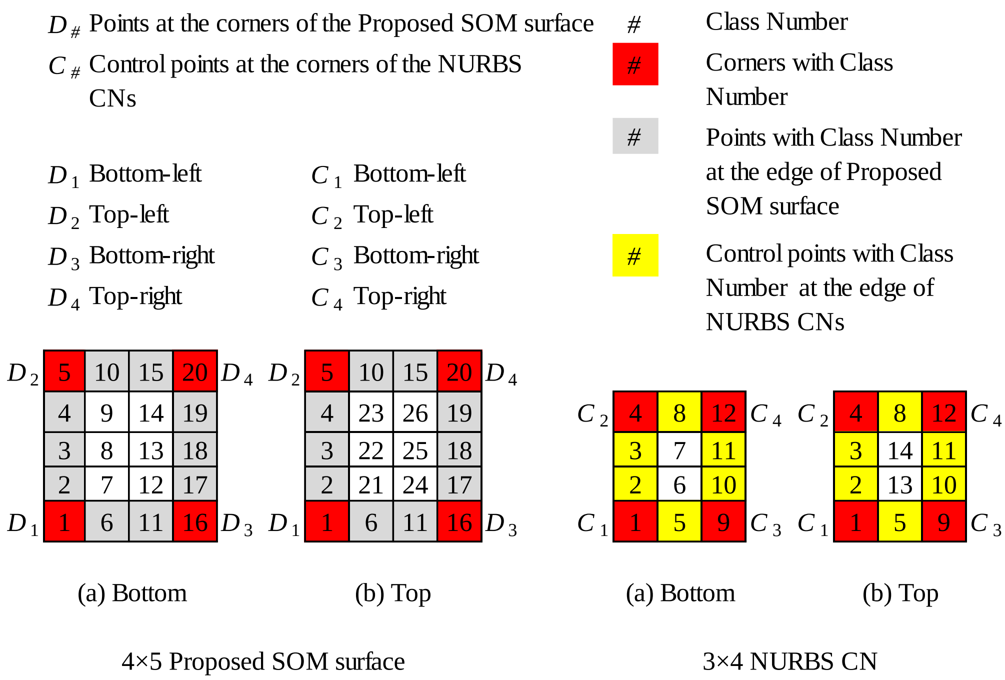


Figure 5.3: Bottom and top surfaces and CN of the DNSOM model and NURBS

Equations 5.8 – 5.11 were the general equations derived to compute the Class Number of each control points at the corner of the CN.

$$c_1 = (nx)^0 \tag{5.8}$$

$$c_2 = ny \tag{5.9}$$

$$c_3 = ny(nx-1) \tag{5.10}$$

$$c_4 = nxny \tag{5.11}$$

where c_i are the corners of the CN and DNSOM surfaces data, *nx* and *ny* are the width and length of the CN or the DNSOM surfaces data respectively.

Manual calculation was performed on the bottom and top CNs as demonstrated in Figure 5.4 to obtain the Class Number of each control points. Each number in the boxes represents the Class Number of the control points from the CNs of different width, CN_x and length, CN_y.

$CN_x = 3$, $CN_y = 4$

4	8	12
3	7	11
2	6	10
1	5	9

(a) Bottom

$CN_x = 3$, $CN_y = 4$

4	8	12
3	14	11
2	13	10
1	5	9

(b) Top

Figure 5.4: Manual calculation of Class Number performed on the bottom and top CNs with CN_x = 3 and CN_y = 4

The Class Number for the control points located at the corners of the CNs were tabulated in Table 5.1 and Arithmetic Progression was used to derive the equations for the improved approach to compute the Class Number of the control points located at the corners of the CNs.

Table 5.1: Derivation of equations used to compute the Class Number of the corners of CNs using Arithmetic Progression

CN_x	CN_y	C_1	C_2	C_3	C_4
3	4	1	4	9	12
4	5	1	5	16	20
5	6	1	6	25	30
6	7	1	7	35	42
7	8	1	8	49	56
…	…	…	…	…	…
		$(nx)^0$	ny	$ny(nx-1)+1$	$nxny$

5.2.3.3 Perform Surfaces Data Calculation

The surfaces data was calculated after the basis functions, control points and surfaces data were calculated. Equation 5.12 was used to compute the NURBS surfaces data.

$$N_u P N_v = D \tag{5.12}$$

where N_u and N_v are the basis functions in u and v directions respectively, P is the control points and D^G is the NURBS surfaces data.

5.2.4 Calculation of Surfaces Error

The surface error for the improved approach was calculated with Equation 5.13 and the equation was based on the Class Number and Euclidean distance formula. The use of Euclidean distance as the evaluation metric was suggested by Piegl and Tiller [139], Adi, Shamsuddin and Ali [140] and Lim and Haron [23]. The total Class Number is the total number of vertices used by the improved approach to represent the surface of the data. The total number of output neurons of the DNSOM model, NOV is used in Equation 5.13 because it is noticed that the equation used to compute the NOV of the DNSOM model can be used to calculate the total number of vertices used by the improved approach to represent the surface of the data. Hence, the total Class Number is equal to NOV.

$$E = \sum_{i=1}^{\mathrm{NOV}} \left| D_i - D_i^G \right| \tag{5.13}$$

where D_i and D_i^G are the DNSOM surface data and improved NURBS surface data for Class Number respectively and the NOV is calculated with Equation 4.2 in Chapter 4, Section 4.2.3.

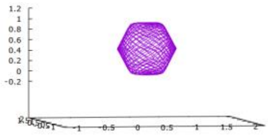
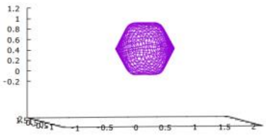
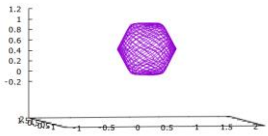
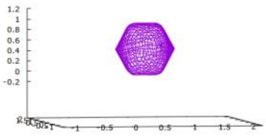
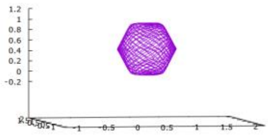
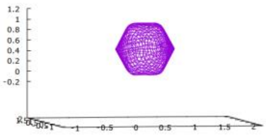
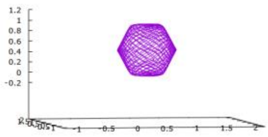
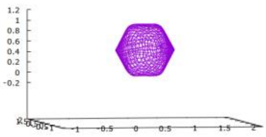
5.3 Analysis and Discussion

This section analyses and discusses the results of the research. Various sizes of control net (CN) and parameterisation methods were used to examine their performance on the conventional and improved NURBS surface approximation approach.

The visualisation of DNSOM model data using the conventional (*A*) and improved (*B*) NURBS surface approximation approaches with various sizes of CN and parameterisation methods for the $n_x = 20$ and $n_y = 20$, and for the $n_x = 18$ and $n_y = 30$ are shown in Appendix G and Appendix H respectively. Based on the results in Appendix G and Appendix H, better surface is generated when the size of CN increases. Table 5.2 and Table 5.3 show the visualisation of the spindle data for the conventional and improved NURBS surface approximation approach in Appendix G and Appendix H respectively given the same and different grid size.

As shown from the visualisation in Table 5.2 and Table 5.3, various parameterisation methods were successfully implemented in conventional and improved NURBS surface approximation approach. Based on the visualisation in Table 5.2 and Table 5.3, the surfaces generated are quite similar to the output of the DNSOM model although different grid size and different parameterisation were used. No gaps were noticed after applying the improved approach and the shape of the data was not affected. Therefore, parameterisation methods can be applied on the improved NURBS surface approximation approach. In addition, to further test the performance of the parameterisation methods, quantitative measurement was used.

Table 5.2: Visualisation of the spindle data for the conventional NURBS surface approximation approach given the same and different grid size

CN	18 × 18	16 × 28
Uniform		
Chord Length		
Centripetal		
Exponential		

Appendix E show the surface error of the conventional and improved approaches for 7 CN (6 × 6, 8 × 8, 10 × 10, 12 × 12, 14 × 14, 16 × 16, 18 × 18) and data when the n_x and n_y are equal to 20. Appendix F show the surface error of the conventional and improved approaches for 7 CN (4 × 16, 6 × 18, 8 × 20, 10 × 22, 12 × 24, 14 × 26, 16 × 28) and data when the n_x is equal to 18 and n_y is equal to 30 respectively.

Table 5.3: Visualisation of the spindle data for the improved NURBS surface approximation approach given the same and different grid size

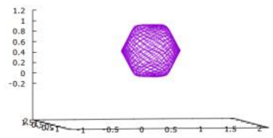
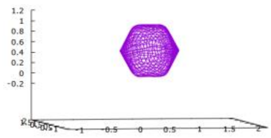
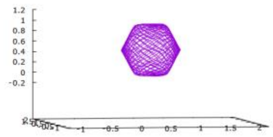
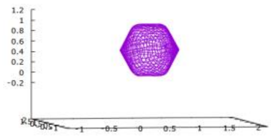
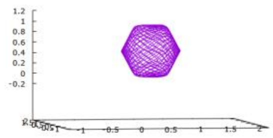
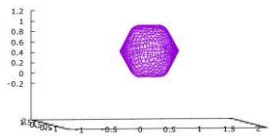
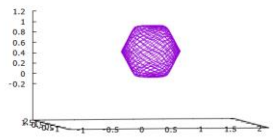
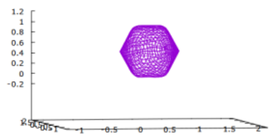
CN	18 × 18	16 × 28
Uniform		
Chord Length		
Centripetal		
Exponential		

Table 5.4 shows the summarised results of the methods in Appendix E that achieved the least surface error in conventional approach for each data when the n_x and n_y is 20. Since there were 7 CNs (6 × 6, 8 × 8, 10 × 10, 12 × 12, 14 × 14, 16 × 16, 18 × 18) for each data and parameterisation method when the n_x and n_y is 20, the method that achieved the least surface error for 6 CNs is recorded as 6/7. When the surface error is small, the NURBS surfaces data are approximated more towards the DNSOM surface data. The Exponential method performed better than the other methods given the cube,

sphere and oiltank data because it achieved the lowest surface error for 3/7, 4/7 and 4/7 CNs respectively. Apart from that, the Chord Length method performed better than the other methods because it obtained the least surface error for most of the CNs which is 4/7 for both spindle and talus bone data. Exponential method was the most suitable method for conventional approach because it achieved the highest number of minimum surface error which is 15/35.

Table 5.4: Summarised results of the methods that achieved the least surface error in conventional approach for each data given the n_x = 20 and n_y = 20

Data	Uniform	Chord Length	Centripetal	Exponential
Cube	2/7	0/7	2/7	3/7
Sphere	0/7	3/7	0/7	4/7
Spindle	0/7	4/7	0/7	3/7
Oiltank	0/7	3/7	0/7	4/7
Talus Bone	0/7	4/7	2/7	1/7
Total	2/35	14/35	4/35	15/35

Table 5.5 shows the summarised results of the methods in Appendix E that achieved the least surface error in improved approach for each CN and data when the n_x and n_y are 20. The Centripetal and Exponential methods performed better than the other methods because they achieved the least surface error for 3/7 CNs given the cube data. The Exponential method also performed better compared to other methods as it achieved the least surface error for 4/7 CNs given the oiltank data. Besides, the Chord Length method achieved better results than the other methods given the sphere, spindle and

talus bone data because they achieved the least surface error for 4/7, 5/7 and 5/7 CNs. Based on the results in Table 5.5, the Chord Length method outperformed the other methods for improved approach as it achieved the highest number of least surface error which is 18/35.

Table 5.5: Summarised results of the methods that achieved the least surface error in improved approach for each data given the n_x = 20 and n_y = 20

Data	Uniform	Chord Length	Centripetal	Exponential
Cube	0/7	1/7	3/7	3/7
Sphere	0/7	4/7	0/7	3/7
Spindle	0/7	5/7	0/7	2/7
Oiltank	0/7	3/7	0/7	4/7
Talus Bone	0/7	5/7	1/7	1/7
Total	0/35	18/35	4/35	13/35

Table 5.6 shows the summarised results for conventional and improved approaches when the n_x and n_y are 20 in Appendix E. Table 5.6 records the total number of least surface error achieved by the approaches for the 7 CNs of each data and parameterisation methods. Since there are 7 CNs for each parameterisation method, the approach that achieved the least surface error for 6 CNs is recorded as 6/7 in Table 5.6. Based on the results in Table 5.6, the improved approach performed better than the conventional approach as it achieved the highest number of minimum surface error which is 126/140.

Table 5.6: Summarised results for conventional (*A*) and improved (*B*) approaches given the n_x = 20 and n_y = 20

Data	Parameterisation Method	*A*	*B*
Cube	Uniform	1/7	6/7
	Chord Length	0/7	7/7
	Centripetal	0/7	7/7
	Exponential	0/7	7/7
Sphere	Uniform	2/7	5/7
	Chord Length	2/7	5/7
	Centripetal	2/7	5/7
	Exponential	2/7	5/7
Spindle	Uniform	0/7	7/7
	Chord Length	0/7	7/7
	Centripetal	0/7	7/7
	Exponential	0/7	7/7
Oiltank	Uniform	3/7	4/7
	Chord Length	0/7	7/7
	Centripetal	1/7	6/7
	Exponential	1/7	6/7
Talus Bone	Uniform	0/7	7/7
	Chord Length	0/7	7/7
	Centripetal	0/7	7/7
	Exponential	0/7	7/7
		14/140	126/140

Table 5.7 shows the summarised results of the methods in Appendix F that achieved the least surface error in conventional approach for various data when the n_x and n_y are 18 and 30 respectively. Since there are 7 CNs (4 × 16, 6 × 18, 8× 20, 10 × 22, 12 × 24, 14 × 26, 16 × 28) for each data and parameterisation method when the n_x and n_y are 18 and 30 respectively, the method that achieved the least surface error for 6 CNs is recorded as 6/7. The Uniform method outperformed the other methods given the cube data as it achieved the least surface error for 3/7 CNs. Meanwhile, the Chord Length method performed better than the other methods for the sphere data as it achieved the least surface error for 3/7 CNs. For the spindle and oiltank data,

the Exponential method outperformed the other methods because it achieved the least surface error for 4/7 and 4/7 CNs respectively. For the talus bone data, the Centripetal method performed better than the other method because it acquired the least surface error for 3/7 CNs. Based on the results in Table 5.7, the Exponential method outperformed the other methods for conventional approach as it achieved the highest number of minimum surface error which is 13/35.

Table 5.7: Summarised results of the methods that achieved the least surface error in conventional approach for various data and sizes of CN given the n_x = 18 and n_y = 30

Data	Uniform	Chord Length	Centripetal	Exponential
Cube	3/7	1/7	2/7	1/7
Sphere	0/7	3/7	2/7	2/7
Spindle	0/7	3/7	0/7	4/7
Oiltank	0/7	3/7	0/7	4/7
Talus Bone	1/7	1/7	3/7	2/7
Total	4/35	12/35	7/35	13/35

Table 5.8 is the summarised results of the methods in Appendix F that achieved the least surface error in the improved approach for various data and sizes of CN when the n_x and n_y are 18 and 30 respectively. Apart from that, the Uniform, Chord Length and Centripetal methods performed better than the Exponential method as they acquired the least surface error for most of the CN given the cube data which is 2/7. For the sphere data, the Chord Length and Centripetal methods performed the best because they achieved the least surface error for 3/7 CNs. For the spindle data, the Chord Length method also

performed better than the other method because it obtained the least surface error for 6/7 CNs. Besides, the Chord Length method achieved better results compared to other methods for the oiltank data because it acquired the least surface error for 4/7 CNs. Furthermore, the Centripetal method performed better than the other methods for the talus bone data as it obtained the least surface error for 3/7 CNs. According to the results shown in Table 5.8, Chord Length method is the most suitable method for the improved approach when the n_x and n_y are 18 and 30 because it achieved the highest number of least surface error which is 17/35.

Table 5.8: Summarised results of the methods that achieved the least surface error in the improved (*B*) approach for various data and sizes of CN given the n_x = 18 and n_y = 30

Data	Uniform	Chord Length	Centripetal	Exponential
Cube	2/7	2/7	2/7	1/7
Sphere	0/7	3/7	3/7	1/7
Spindle	0/7	6/7	0/7	1/7
Oiltank	0/7	4/7	0/7	3/7
Talus Bone	0/7	2/7	3/7	2/7
Total	2/35	17/35	8/35	8/35

Table 5.9 shows the summarised results of conventional and improved approaches when the n_x and n_y are 18 and 30 respectively in Appendix F. Based on the results in Table 5.9, the improved approach outperformed the conventional approach as it achieved the highest number of minimum surface error which is 118/140.

Table 5.9: Summarised results of the conventional (*A*) and improved (*B*) approaches given the n_x = 18 and n_y = 30

Data	Parameterisation Method	*A*	*B*
Cube	Uniform	2/7	5/7
	Chord Length	1/7	6/7
	Centripetal	1/7	6/7
	Exponential	1/7	6/7
Sphere	Uniform	1/7	6/7
	Chord Length	1/7	6/7
	Centripetal	0/7	7/7
	Exponential	1/7	6/7
Spindle	Uniform	2/7	5/7
	Chord Length	1/7	6/7
	Centripetal	1/7	6/7
	Exponential	1/7	6/7
Oiltank	Uniform	3/7	4/7
	Chord Length	1/7	6/7
	Centripetal	2/7	5/7
	Exponential	1/7	6/7
Talus Bone	Uniform	1/7	6/7
	Chord Length	0/7	7/7
	Centripetal	1/7	6/7
	Exponential	0/7	7/7
		22/140	118/140

According to the results in Table 5.4 and Table 5.7, the Exponential method was the best method for the conventional approach as it achieved the least surface error for most of the experiments given the same and different grid size. Based on the results in Table 5.5 and Table 5.8, the Chord Length method was the best method for the improved approach because it obtained the highest number of minimum surface error given the same and different grid size. The surface error of both approaches also decreases when the size of CN increases because there were more control points available to adjust the shapes. This proves that surfaces of the approaches were approximated more towards the output of the DNSOM model when the size of the CN increases and better surfaces were generated. Uniform method performed the worst because it was not suitable for data that were not distributed linearly [89]. Meanwhile, the Centripetal method was better than the Chord Length method in handling sharp turns [88] whereas the Exponential method has the property of Centripetal and Chord Length methods since the value of its α in Equation 2.8, Chapter 2 is 0.8, which is between the value of α set for the Centripetal and Chord Length methods. So, the Chord Length method performed better than the Exponential method for most of the experiments because most of the data in this experiment do not have sharp turn. Therefore, based on the results in Table 5.5 and Table 5.8, Chord Length method was considered the best method in the improved approach because it achieved the least surface error for 35/70 results.

The NURBS surface approximation approach with improvements were proposed to overcome the problem faced by the conventional NURBS surface approximation approach [23] when it was applied on the DNSOM surfaces data. Gaps were observed at the edges of both surfaces after the approach was applied on the output of the model as shown in Figure 5.5.

It was observed that when the size of the CN increases, the size of the gaps reduces and better surfaces were generated. Improvements were made to the perform parameterisation and perform control points calculation processes of the conventional approach to avoid the gaps from appearing. No gaps were found after the implementation of the improved approach as shown in Figure 5.6 because the boundaries between the NURBS surfaces are merged during the perform parameterisation and perform control points calculation steps.

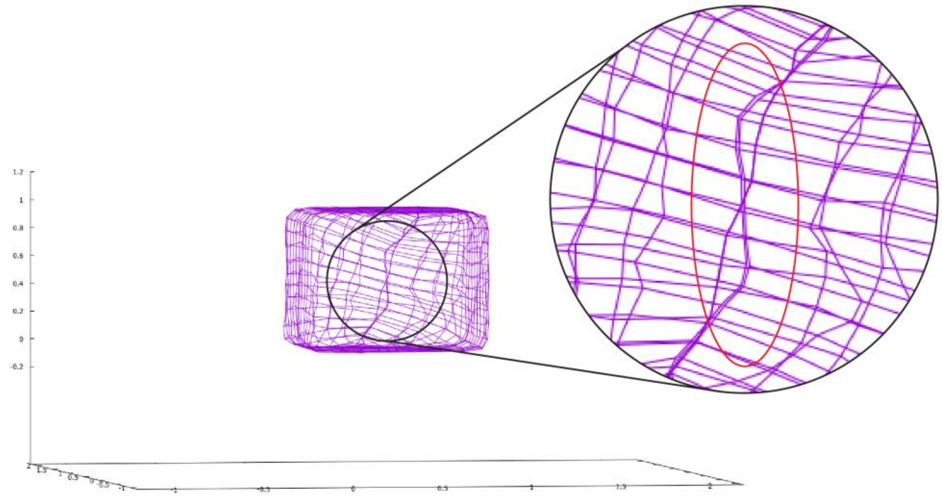


Figure 5.5: Image of NURBS surfaces generated with conventional approach for 18 × 18 CN and uniform parameterisation method given the cube data

Hence, lower surfaces error was obtained for the improved approach compared to the conventional approach.

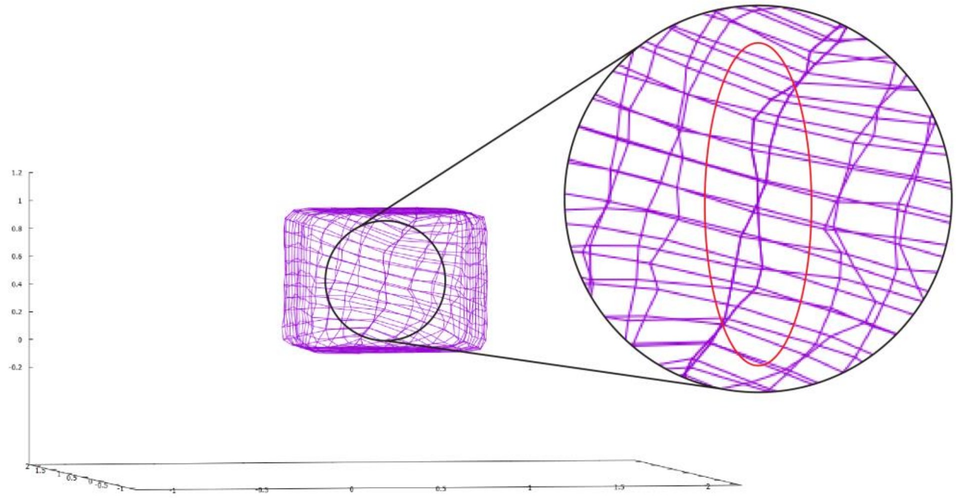


Figure 5.6: Image of NURBS surfaces generated with improved approach for 18 × 18 CN and uniform parameterisation method given the cube data

5.4 Summary

The conventional NURBS surface approximation approach was proven applicable on the output of the DNSOM model. Exponential method was the best method for the conventional approach. But gaps were observed at the edges of both surfaces when the approach was applied on the output of the model. The size of the gaps was reduced and better surfaces were generated when the size of CN increases for the conventional approach. Improvements were made to the perform parameterisation and perform control points calculation processes of the conventional approach to close the gaps. No gaps were found after the implementation of the improved NURBS surface approximation approach. The Chord Length method is the best method for the improved approach. Besides, the improved approach also performed better than the conventional approach because it achieved the least surface error for most of the experiments. Therefore, the improved approach performed better than the conventional approach. The surface error of both approaches decreases when the size of CN increases. This demonstrates that the surfaces generated is better when the size of the CN increases. However, more accurate surface can be generated by tuning the control points of NURBS with optimisation technique as shown in [28].

CHAPTER 6

OPTIMISATION OF THE IMPROVED NURBS SURFACE APPROXIMATION APPROACH

6.1 Overview

This chapter demonstrates the system flow, analyses and discusses the optimisation of control points of the improved NURBS surface approximation approach using optimisation techniques such as Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimisation (PSO). Optimisation techniques were applied because the NURBS surfaces generated were not the optimum as they were generated mathematically. Thus, optimisation techniques were applied to generate the improved NURBS surfaces with higher accuracy by optimising the control points of the improved NURBS surface approximation approach.

6.2 System Flow for the Optimisation of the Improved NURBS Surface Approximation Approach

This section discusses the system flow of GA, DE and PSO in optimising the control points. GA with Value Encoding, Tournament Selection, Uniform Crossover, Uniform Mutation and Weak Parent Replacement, DE and PSO with constriction factor and velocity clamping are used to optimise the control points. Based on the work in [141], GA with Tournament Selection, Uniform Crossover, flip mutation and Weak Parent Replacement achieved the highest number of minimum average fitness values. Therefore, the Tournament Selection and Uniform Crossover were used for the GA. The flip mutation was replaced with Uniform Mutation because flip mutation is not suitable for value encoding GA. Besides, GA with the abovementioned operations was used because this research focuses in proving that GA can be used to optimise the control points of the improved approach. DE [33] was employed for the same purpose. The PSO with constriction factor and velocity clamping was used in this research because it achieved the best result in the work of Lim, Hoon and Song [34].

6.2.1 Data Acquisition

The data used in this research were the control points, basis function, surface data from improved NURBS surface approximation approach and DNSOM surface data. These data were needed to optimise the control points. Figure 6.1 is an example of the bottom and top 3 × 4 CN with several chosen control points. 1-D array was used to represent the control points and it was used as an input for the GA, DE and PSO. The representation of the control points in a 1-D array is shown in Figure 6.2. The representation is based on Figure 6.1. Each control point, P_i is in their coordinate form (x_i, y_i, z_i), where i is the Class Number. The concept of Class Number was used in this research to group the redundant control points accordingly. In GA and DE, the 1-D array representing the control points is generally known as chromosome. Meanwhile, the array is known as particle in PSO. To optimise the control points, individuals or particles was generated initially by summing the value of x, y and z coordinates of each control point with their respective randomly generated value within the range of [-5×10^{-6}, 5×10^{-6}].

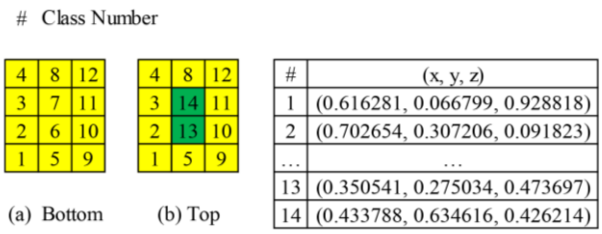


Figure 6.1: The bottom and top 3 × 4 CN

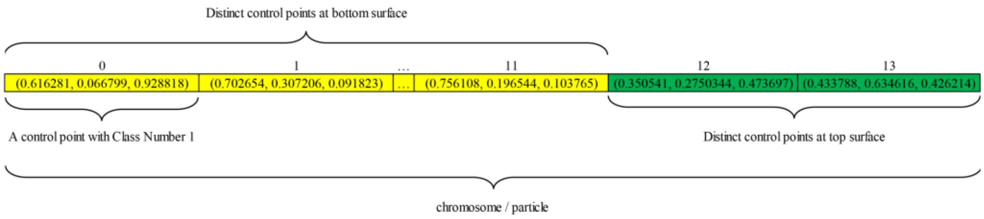


Figure 6.2: Representation of the control points in Figure 6.1 in chromosome or particle

6.2.2 Parameter Settings

Table 6.1 presents the common parameter settings for GA, DE and PSO, and their respective parameter settings except the Number of Dimension, *d* and the Control Point Coordinate Range are referred from [34], [141], [142]. The Number of Generation is the maximum generation and it was the termination criterion used in this research. The Population Size, *n* is set to 40. The Number of Dimension for the chromosome and particle is NOV, where NOV is the number of control point used to generate the surface data by the improved NURBS surface approximation approach. The concept of Class Number is utilised to group the redundant control points at the edges of the control net (CN) of the approach. Thus, the NOV would be the total number of Class Number. The equation used to compute the NOV can be referred from Equation 4.2 in Chapter 4, Section 4.2.3 by substituting the width, n_x and length, n_y of the grid for DNSOM model with the width, CN_x and length, CN_y of the CN respectively. For GA, the Crossover Probability, P_c and Mutation Probability, P_m used are 0.7 and 0.01. Meanwhile, the Crossover Probability and Differential Weight, F of DE is 0.01 and 0.8. The acceleration constants, c_1 and c_2 of the PSO was set to 2.05. The maximum velocity and position of the PSO was set to half of the range of the data set and the range of the data set respectively. Maximum velocity was used as the velocity clamping and maximum position is set to prevent the particle from moving beyond the boundaries of the search space. The velocity of each particle was initialised to 0. The Constriction Factor, K of the PSO was set to 0.729. The GA, DE and PSO were run 10 times for each experiment to show that different fitness value is acquired at each run. Average fitness value and CPU time were computed from the fitness value of all the run for each of the experiment. A small range was set as the control point coordinate range because there would be lesser combination of control points. Consequently, it would be easier to find the best control points.

6.2.3 Fitness Function

The DNSOM surfaces data, D^O is computed with Equation 6.1 and it is also defined by rewriting the Equation 5.12 in Chapter 5, Section 5.2.3.3.

$$N_u P^O N_v = D^O \tag{6.1}$$

where N_u and N_v are the basis function in u and v direction respectively, P^O is the control points, and D^O is the optimised NURBS surface data of the improved NURBS surface approximation approach.

Table 6.1: Parameter settings

No.	Parameter	Value
1.	Number of Generation	2000
2.	Population Size, n	40
3.	Number of Dimension, d	NOV
4.	Control Point Coordinate Range	$[-5 \times 10^{-6}, 5 \times 10^{-6}]$
5.	GA Crossover Probability, P_c	0.7
6.	GA Mutation Probability, P_m	0.01
7.	DE Crossover Probability	0.8
8.	DE Differential Weight, F	0.3
9.	PSO Constriction Factor, k	0.729
10.	PSO Random Number (r_1, r_2)	[0, 1]
11.	PSO Maximum Velocity	Half of the range of the dataset
12.	PSO Maximum Position	The range of the dataset
13.	PSO Acceleration Constant (c_1 and c_2)	2.05
14.	Number of Testing	10

The fitness value in this research is calculated with the fitness function defined in Equation 6.2. The equation is based on Euclidean distance and Class Number and it is defined by rewriting the Equation 5.13 in Chapter 5, Section 5.2.4.

$$f(x) = \sum_{i=1}^{\mathrm{NOV}} \left| D_i - D_i^O \right| \tag{6.2}$$

where D_i and D_i^O are the DNSOM surfaces data and optimised NURBS surface data from the improved NURBS surface approximation approach respectively.

6.2.4 Optimisation of Control Points

Optimisation of control points was performed with GA, DE and PSO. This section describes the system flow of GA, DE and PSO in optimising the control points.

6.2.4.1 Genetic Algorithm

The optimisation of control points with GA is described as follow:

1. A population of *n* chromosomes are generated by summing the value of *x*, *y* and *z* coordinates of the control points with their respective randomly generated value within the range of [-5×10^{-6}, 5×10^{-6}].

2. The fitness of each chromosome is evaluated with the fitness function, $f(x)$ in Equation 6.2.

3. Tournament Selection: 4 chromosomes are randomly selected and their fitness value are compared. Chromosome with the least fitness value is selected as the parent. Another round of tournament is conducted to find the next parent.

4. Uniform Crossover: A random number is generated and crossover will be executed if the random number is less than the Crossover Probability, P_c.

5. Uniform Mutation: A random number is generated and mutation will be executed if the random number is less than the Mutation Probability, P_m.

6. The fitness of the new offspring is evaluated with the fitness function, $f(x)$ in Equation 6.2.

7. Weak Parent Replacement: The fitness value of the parent will be compared with the fitness value of the offspring. If the fitness value of the offspring is smaller than that of its parent, the offspring will be included in the next generation. Else, the parent will be included.

8. If the termination criterion is not met, the algorithm continues with 3. Else, the algorithm is terminated and the chromosome with the least fitness

value was produced. The fitness value of the best chromosome and the optimised surface data calculated from the chromosome were also produced.

6.2.4.2 Differential Evolution

The optimisation of control points with DE is described as follow:

1. A population of n chromosomes are generated by summing the value of x, y and z coordinates of the control points with their respective randomly generated value within the range of $[-5 \times 10^{-6}, 5 \times 10^{-6}]$.

2. The fitness of chromosomes is evaluated using the fitness function, $f(x)$ in Equation 6.2.

3. Choose the target vector with index, $i = 1$. Target vector with index, $i = 1$ is the first chromosome in the population.

4. Mutation: For each target vector, a mutant vector is generated. Three indices r_1, r_2 and r_3 are randomly chosen from the population to form the mutant vector. The indices are integer and they are mutually different from one another. The indices are also different from the index, i.

5. Crossover: A random number is generated. If the random number is at least less than the DE Crossover Probability or equal to the randomly chosen index, the value at the current dimension of the mutant vector will be donated to the same dimension of the trial vector. Else, the value from the target vector will be donated to the trial vector. The crossover operation continues until the trial vector is constructed.

6. The fitness of trial vector is evaluated by using $f(x)$ in Equation 6.2.

7. Selection: If the fitness value of the trial vector is less than the target vector, the trial vector will be included in the next generation. Otherwise, the target vector is included.

8. If the index of the current target vector is at least less than *n*, choose the next target vector by incrementing the index by 1 and proceed with 4. Otherwise, proceed to 9.

9. If the termination criterion is not met, the algorithm continues with 3. Otherwise, the algorithm is terminated and the chromosome with the least fitness value was produced. The fitness value of the best chromosome and the optimised surface data calculated from the chromosome were also produced.

6.2.4.3 Particle Swarm Optimisation

The optimisation of control points with PSO is described as follow:

1. *n* particles are generated by summing the value of *x*, *y* and *z* coordinates of the control points with their respective randomly generated value within the range of $[-5 \times 10^{-6}, 5 \times 10^{-6}]$.

2. The fitness of the particles is evaluated with the fitness function, $f(x)$ in Equation 6.2.

3. The position of each particle is set as their best position. The particle with the smallest fitness value among *n* particles is selected as the global best particle.

4. The velocity and position of the *n* particles are computed and updated according to their best position and the position of the global best particle.

5. The fitness of the particles is evaluated with the fitness function, $f(x)$ in Equation 6.2.

6. The best position of each particle and the position of the best particle among the n particles are updated.

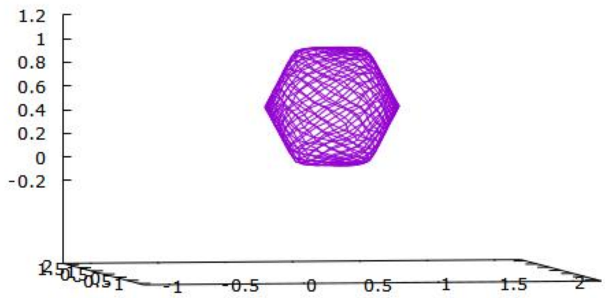
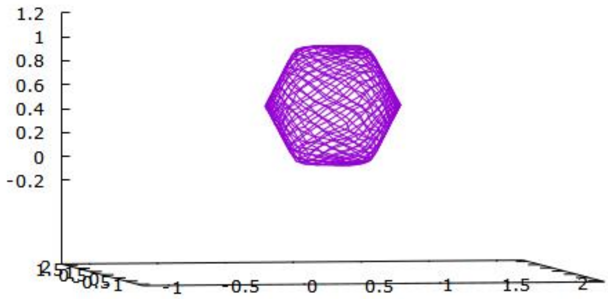
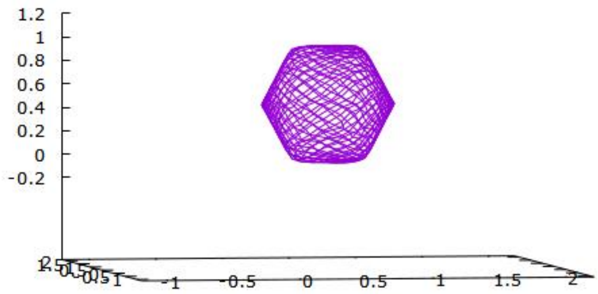
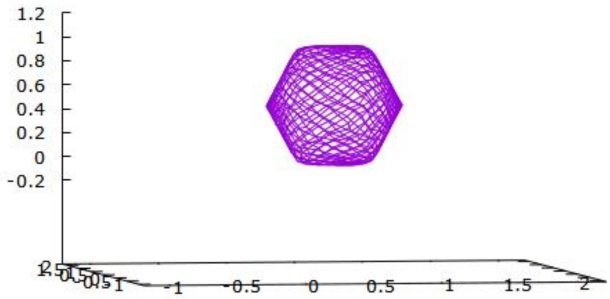
7. If the termination criterion is not met, the algorithm continues with 3. Otherwise, the algorithm will be terminated and the particle with the least fitness value was produced. The fitness value of the best particle and the optimised surface data calculated from the particle were also produced.

6.3 Analysis and Discussion

The outcome of the optimisation of control points using GA, DE and PSO were analysed and discussed in this section. The GA with Tournament Selection, Uniform Crossover, Uniform Mutation and Weak Parent Replacement, DE and PSO with constriction factor and velocity clamping were used to optimise the control points of the improved NURBS surface approximation approach.

Appendix I shows the visualisation of optimised surface data with the minimum (MIN) and maximum (MAX) optimised surface error obtained from the GA, DE and PSO and the visualisation of improved NURBS surface data (B) for the remaining data and parameterisation methods given the CNs of the same size. Table 6.2 shows the visualisation of optimised surface data with the minimum (MIN) optimised surface error obtained from the GA, DE and PSO and the visualisation of improved NURBS surface data (B) for the spindle data, CN with the size of 18×18 and Chord Length method, where each of the CN are of the same length and width.

Table 6.2: Visualisation of MIN optimised surface data of GA, DE and PSO with the same length and width

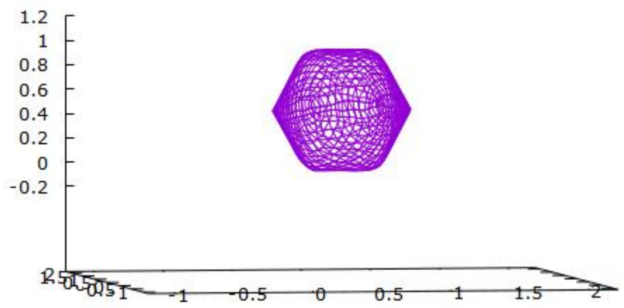
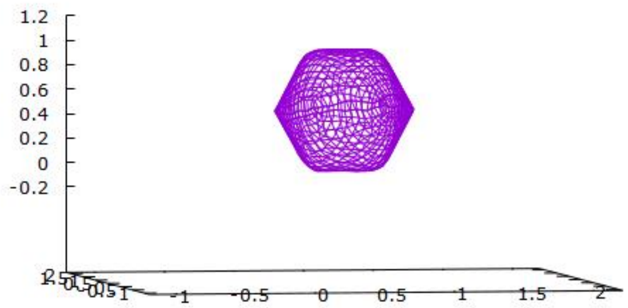
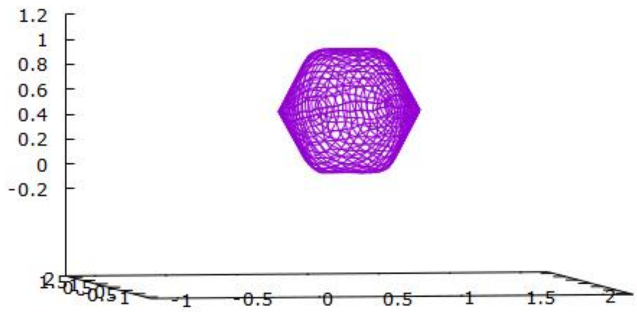
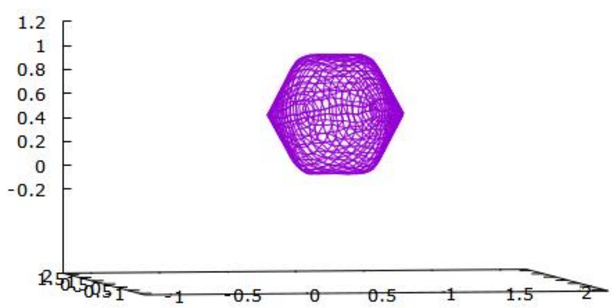
CN	18 × 18
GA	
DE	
PSO	
B	

Appendix J shows the visualisation of optimised surface data with the minimum (MIN) and maximum (MAX) optimised surface error obtained from the GA, DE and PSO and the visualisation of improved NURBS surface data (*B*) for the remaining data and parameterisation methods given the CNs of different size. Table 6.3 shows the visualisation of optimised surface data with the minimum (MIN) optimised surface error obtained from the GA, DE and PSO and the visualisation of improved NURBS surface data (*B*) for the spindle data, CN with the size of 16 × 28 and Chord Length method, where each of the CN are of the same length and width.

As shown from the visualisation in Table 6.2 and Table 6.3, GA, DE and PSO was successfully implemented on the improved approach. When the size of CN increases, better improved NURBS surfaces were generated. Based on the visualisation in Table 6.2 and Table 6.3, correct surfaces were generated and they were quite similar to the improved NURBS surfaces. Besides, the correct surfaces can be generated although different grid size, different parameterisation, different optimisation methods were used. No gaps were noticed after optimisation and the shape of the data was not affected. Therefore, optimisation techniques can be applied on the improved NURBS surface approximation approach. In addition, to further test the performance of the optimisation methods, quantitative measurement was used.

Appendix K and Appendix L show the experimental results for GA, DE and PSO via the optimisation of CN with same and different grid size.

Table 6.3: Visualisation of MIN optimised surface data of GA, DE and PSO with different length and width

CN	16 × 28
GA	
DE	
PSO	
B	

The experimental results provide information on average (AVG), minimum (MIN) and maximum (MAX) optimised surface error obtained for various parameterisation methods and data sets. Table 6.4 shows the summarised results of the techniques in Appendix K that achieved the least average (AVG) optimised surface error for various data, methods and sizes of CN when the n_x and n_y are 20. As shown in the Table 6.4, 7 is referring to the various sizes of CN (6 × 6, 8 × 8, 10 × 10, 12 × 12, 14 × 14, 16 × 16, 18 × 18) of each parameterisation method. Therefore, the total number of summarised results is 140.

Table 6.4: Summarised results of the techniques that achieved the least average (AVG) optimised surface error for various data, methods and sizes of CN given the n_x = 20 and n_y = 20

Data	Parameterisation Method	GA	DE	PSO
Cube	Uniform	0/7	7/7	0/7
	Chord Length	0/7	7/7	0/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	7/7	0/7
Sphere	Uniform	0/7	7/7	0/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Spindle	Uniform	0/7	7/7	0/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Oiltank	Uniform	0/7	7/7	0/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Talus Bone	Uniform	0/7	6/7	1/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Total	-	0/140	126/140	14/140

Besides, Table 6.5 shows the summarised results of the techniques in Appendix L that achieved the least average (AVG) optimised surface error for various data, methods and sizes of CN when n_x and n_y were 18 and 30 respectively.

Table 6.5: Summarised results of the techniques that achieved the least average (AVG) optimised surface error for various data, methods and sizes of CN given the n_x = 18 and n_y = 30

Data	Parameterisation Method	GA	DE	PSO
Cube	Uniform	0/7	6/7	1/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Sphere	Uniform	0/7	6/7	1/7
	Chord Length	0/7	5/7	2/7
	Centripetal	0/7	5/7	2/7
	Exponential	0/7	6/7	1/7
Spindle	Uniform	0/7	6/7	1/7
	Chord Length	0/7	5/7	2/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Oiltank	Uniform	0/7	6/7	1/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Talus Bone	Uniform	0/7	6/7	1/7
	Chord Length	0/7	6/7	1/7
	Centripetal	0/7	6/7	1/7
	Exponential	0/7	6/7	1/7
Total	-	0/140	117/140	23/140

Appendix M and Appendix N show the average (AVG), minimum (MIN) and maximum (MAX) CPU time obtained using GA, DE and PSO via the optimisation of CN and the surface error of improved (B) NURBS surface approximation approach for various parameterisation methods and data sets where each of the CN involved are of the same length and width, and different length and width respectively. Table 6.6 is the summarised results of the techniques in Appendix M that achieved the least average (AVG) CPU time for various data, methods and sizes of CN given the n_x and n_y are 20.

Table 6.6: Summarised results of the techniques that achieved the least average (AVG) CPU time for various data, methods and sizes of CN given the $n_x = 20$ and $n_y = 20$

Data	Parameterisation Method	GA	DE	PSO
Cube	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Sphere	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Spindle	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Oiltank	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Talus Bone	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Total	-	140/140	0/140	0/140

Furthermore, Table 6.7 shows the summarised results of the techniques in Appendix N that achieved the least average (AVG) CPU time for various data, methods and sizes of CN given the n_x and n_y are 18 and 30 respectively.

Table 6.7: Summarised results of the techniques that achieved the least average (AVG) CPU time for various data, methods and sizes of CN given the $n_x = 18$ and $n_y = 30$

Data	Parameterisation Method	GA	DE	PSO
Cube	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Sphere	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Spindle	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Oiltank	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Talus Bone	Uniform	7/7	0/7	0/7
	Chord Length	7/7	0/7	0/7
	Centripetal	7/7	0/7	0/7
	Exponential	7/7	0/7	0/7
Total	-	140/140	0/140	0/140

Thus, the overall results for optimised surface errors of GA, DE and PSO are reflected in Table 6.8. The overall number of results given the Table 6.4 and Table 6.5 is 280.

Table 6.8: Overall results for optimised surface errors of GA, DE and PSO

Result	GA	DE	PSO
Table 6.4	0/140	126/140	14/140
Table 6.5	0/140	117/140	23/140
Overall Total	0/280	243/280	37/280

Thus, the overall results for CPU time of GA, DE and PSO are reflected in Table 6.9. The overall number of results given the Table 6.6 and Table 6.7 is also 280.

Table 6.9: Overall results for CPU time of GA, DE and PSO

Result	GA	DE	PSO
Table 6.6	140/140	0/140	0/140
Table 6.7	140/140	0/140	0/140
Overall Total	280/280	0/280	0/280

Based on the results in Table 6.8, DE obtained the least average optimised surface error for most of the experiments which is 243/280 (126/140 for the same grid size, 117/140 for different grid size). Meanwhile, PSO achieved the second-most total number of least average optimised surface error which is 37/280 (14/140 for the same grid size, 23/140 for different grid size) as shown in Table 6.8. In contrast, GA achieved 0/280 least surface error (0/140 for the same grid size, 0/140 for different grid size).

Based on the results in Table 6.9, GA achieved 280/280 least average CPU time (140/140 for the same grid size, 140/140 for different grid size). Meanwhile, both DE and PSO achieved 0/280 least average CPU time (0/140 for the same grid size, 0/140 for different grid size).

In terms of the average optimised surface error, DE performed better than GA and PSO because the individuals in DE can easily explore the new region in the search space [149] and converged to the optimal solution better than GA and PSO. Besides, DE performed better than GA because mutation operation in DE was performed on the target vector at every iteration while the mutation operation in GA was performed according to the mutation probability applied [113]. Thus, DE can maintain the population diversity better than GA. Furthermore, the difference between the mutation operation in DE and GA is that the mutation operation in DE is the outcome of arithmetic combinations of individuals while the mutation operation in GA is the outcome of small perturbations to the genes of an individual [151]. In terms of the CPU time, DE performed slower than GA because each target vector in DE was compared with the trial vector to produce a new target vector at each iteration.

The PSO can perform better than DE for certain cases because technique with better exploration property such as DE is more efficient in finding the optimal solution when the size of CN is small. As the size of the CN increases, technique with a good balance between exploration and exploitation property such as PSO with velocity clamping is better in finding the optimal solution. PSO with velocity clamping can control the position and velocity of each particle and it was developed to make a good balance between exploration and exploitation [152]. The PSO has better accuracy for high-dimensional problems [150]. The PSO performed slower than DE because the particles in PSO must update their position and velocity both locally and globally at each iteration [142].

GA performed the worst because there are only two new individuals that can be introduced at each iteration [142]. Two new individuals are introduced when both selected parents achieved a higher optimised surface error than their offspring [153]. Meanwhile, *n* new individuals and particles are introduced in DE and PSO at every iteration. Thus, DE and PSO are better than GA in finding the optimal solution. GA performed the fastest despite having a greater number of operations involved such as selection, crossover, mutation and replacement [141] because not every individual was updated at each iteration unlike the DE and PSO in which every individual in DE and particle in PSO were updated at each iteration. Thus, GA performed the best in terms of the CPU time followed by the DE and PSO. The information can be referred in Appendix M and Appendix N.

In short, DE is the best optimisation technique because it can achieve the least optimised surface error for most of the experiment and generate the output at a moderate CPU time.

6.4 Summary

Optimisation methods such as GA, DE and PSO can be implemented to optimise the control points of the improved NURBS surface approximation approach. The DE performed the best followed by PSO and GA. However, the PSO can perform better than DE for certain cases. When the size of CN is small, optimisation method with exploration property such as DE is more efficient in finding the optimal solution, whereas optimisation method with both exploration and exploitation property is better in finding the optimal solution when the size of the CN is large. The GA took the shortest CPU time among the techniques, followed by DE and PSO. Therefore, the results show that the optimisation methods can optimise the surface produced by the improved approach and be used in surface reconstruction.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 Conclusion

A model inspired by Lim and Haron [25] dubbed Double Net Self-Organising Map (DNSOM), was proposed through the merging of two 2-D Self-Organising Map (SOM). The model was designed to organise the unstructured data and to regain the connectivity information of the data. The model overcomes the drawbacks of 2-D SOM, 3-D SOM and Cube Kohonen SOM (CKSOM). Besides, the model organised the unstructured data successfully. It was able to preserve the topology of the data as it achieved the lowest Topographic Error (TE) compared to all the SOM models. Additionally, the DNSOM model can produce the correct surface with fewer neurons compared to the CKSOM model and the length and width of its grid can be tuned with different values. Besides, the improved NURBS surface approximation approach was integrated successfully on the DNSOM surfaces data and four equations were derived to identify the four corners of the CNs and DNSOM surfaces data. The improved NURBS surface approximation approach can generate the improved NURBS surfaces without gaps. The Chord Length method is the best method because it achieved the least surface error for most of the experiments. GA performed the best in term of CPU time although it has more operations compared to DE and PSO. Meanwhile, PSO achieved the highest CPU time among the optimisation methods.

7.2 Contributions

The ability of the DNSOM model to organise the unstructured data of talus bone would contribute to the field of medical imaging in a way that the model can help the medical practitioner to create an artificial talus bone for patients whom their talus bone is damaged. The model can also be utilised in reverse engineering to recover the digital file of an object. Hence, the designers can save their time from redesigning the object from scratch. In addition, it does not require huge amount of data for training. Thus, it can reconstruct the surface of an object faster. Besides, three equations were derived based on the model so that the researcher can identify the NON, NOV and NOR for the model and can construct the model based on the equations. Furthermore, the improved NURBS surface approximation approach can generate multiple surfaces without gaps and four equations were derived to identify the four corners of the DNSOM surfaces data and CN. Besides, this research proves that the concept of applying the GA, DE and PSO to optimise the control points of the improved NURBS surface approximation approach is applicable.

7.3 Limitations

Although the DNSOM model can organise the unstructured data, it suffers from several limitations. It has higher QE than 3-D SOM and CKSOM models and it fails to reconstruct the ear of the Stanford bunny data correctly. Besides, the height of the DNSOM model is fixed. The lack of flexibility in setting the height of the model may cause the model to be inappropriate to present longer objects.

7.4 Future works

Future work can consider minimising the QE with optimisation methods and enhancing the Detecting Neighbours (Section 4.2.4) for the adaptation process as demonstrated in [24] to overcome the limitations mentioned previously. Besides, more complex data such as the happy Buddha, dragon and armadillo from can be used to test the performance of the model. Other parameterisation methods such as the hybrid method [87], deep learning method [143] and dynamic centripetal method [89] can also be applied to further examine their influence on the improved NURBS surface approximation approach. The optimisation techniques can also be used to optimise the weights of the improved approach. Furthermore, optimisation techniques such as Bat algorithm [144] and Firefly algorithm [145] can also be used to optimise the control points of the improved approach and examine the performance between the techniques.

References

[1] C. Shang, J. Fu, J. Feng, Z. Lin and B. Li, “Effective re-parameterisation and GA based knot structure optimisation for high quality T-spline surface fitting,” *Comput. Methods Appl. Mech. Eng.*, vol. 351, pp. 836-859, July 2019, doi: 10.1016/j.cma.2019.03.033.

[2] A. B. Makhlouf, B. Louhichi, D. Deneux and M. A. Mahjoub, “Reconstruction of a CAD model using TPS surface,” in *2019 23rd International Conference Information Visualisation (IV)*, 2019, pp. 417-424, doi: 10.1109/IV.2019.00077.

[3] R. Palomar, F. A. Cheikh, B. Edwin, A. Beghdadhi and O. J. Elle, “Surface reconstruction for planning and navigation of liver resections,” *Comput. Med. Imaging Graph.*, vol. 53, pp. 30-42, July 2016, doi: 10.1016/j.compmedimag.2016.07.003.

[4] P. Liu, N. Hewitt, W. Shadid and A. Willis, “A system for 3-D reconstruction of comminuted tibial plafond bone fractures,” *Comput. Med. Imaging Graph.*, vol. 89, Apr. 2021, Art. no. 101884, doi: 10.1016/j.compmedimag.2021.101884.

[5] Q. Tian, B. Bilgic, Q. Fan, C. Ngamsombat, N. Zaretskaya, N. E. Fultz, N. A. Ohringer, A. S. Chaudhari, Y. Hu, T. Witzel, K. Setsompop, J. R. Polimeni and S. Y. Huang, “Improving in vivo human cerebral cortical surface reconstruction using data-driven super-resolution,” *Cereb. Cortex*, vol. 31, no. 1, pp. 463-482, Jan. 2021, doi: 10.1093/cercor/bhaa237.

[6] D. -Y. Zhong, L. -G. Wang, M. -T. Jia, L. Bi and J. Zhang, “Orebody modelling from non-parallel cross sections with geometry constraints,” *Minerals*, vol. 9, no. 4, Apr. 2019, Art. no. 229, doi: 10.3390/min9040229.

[7] W. Cao, Y. Shi, D. Mei, M. Li and D. Liu, “Reconstruction of ancient stone arch bridge via terrestrial LiDAR technology,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 960, no. 5, Dec. 2020, Art. no. 042063, doi: 10.1088/1757-899X/960/4/042063.

[8] T. Mikita, M. Balková, A. Bajer, M. Cibulka and Z. Patočka, "Comparison of different remote sensing methods for 3-D modeling of small rock outcrops," *Sensors*, vol. 20, no. 6, Art. no. 1663, 2020.

[9] C. Serazio, M. Tamburini, F. Verga and S. Berrone, "Geological surface reconstruction from 3-D point clouds," *MethodsX*, vol. 8, Art. no. 101398, 2021.

[10] S. Yu, T. Chen and G. Hu, "Confidence-constrained support vector regression for geological surface uncertainty modelling," *IEEE Access*, vol. 8, pp. 182451-182461, 2020.

[11] M. Li, F. Rottensteiner and C. Heipke, "Modelling of buildings from aerial LiDAR point clouds using TINs and label maps," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 154, pp. 127-138, 2019.

[12] X. J. Qin, Z. T. Hu, H. B. Zheng, and M. Y. Zhang, "Surface reconstruction from unorganised point clouds based on edge growing," *Adv. Manuf.*, vol. 7, no. 3, pp. 343-352, Sept. 2019, doi: 10.1007/s40436-019-00262-5.

[13] J. Liu, "An adaptive process of reverse engineering from point clouds to CAD models," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 9, pp. 840-858, 2020.

[14] Ö. Verim and M. Yumurtacı, "Application of reverse engineering on spacecraft assembly process simulation," *IOP Conference Series: Materials Science and Engineering*, vol. 727, no. 1, 2020, Art. no. 012017.

[15] Y. Xu, C. Yang, K. Xu, W. Wang, X. Fan and J. Hou, "Application of reverse engineering on spacecraft assembly process simulation," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 727, no. 1, Jan. 2020, Art. no. 012017, doi: 10.1088/1757-899X/727/1/012017.

[16] C. Y. Tung, R. Sallehuddin, H. Haron and A. Malyshev, "An introductory review to surface reconstruction," *IOP Conference Series: Materials Science and Engineering*, vol. 864, no. 1, Art. no. 012052, 2020.

[17] P. Guerrero, Y. Kleiman, M. Ovsjanikov and N. J. Mitra, "PCPNet: learning local shape properties from raw point clouds," *Computer Graphics Forum*, vol. 37, no. 2, pp. 75-85, 2018.

[18] Y. Ben-Shabat, M. Lindenbaum and A. Fischer, "Nesti-Net: normal estimation for unstructured 3-D point clouds using convolutional neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] S. P. Lim and H. Haron, "Surface reconstruction techniques: a review," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 59-78, 2014.

[20] R. L. M. E. do Rêgo, A. F. R. Araújo and F. B. de Lima Neto, "Growing self-organising maps for surface reconstruction from unstructured point clouds," in *2017 International Joint Conference on Neural Networks*, 2017, pp. 1900-1905, doi: 10.1109/IJCNN.2007.4371248.

[21] S. Orts-Escolano, J. Garcia-Rodriguez, V. Morell, M. Cazorla, J. A. S. Perez and A. Garcia-Garcia, "3-D surface reconstruction of noisy point clouds using growing neural gas: 3-D object/scene reconstruction," *Neural Process. Lett.*, vol. 43, no. 2, pp. 401-423, 2016, doi: 10.1007/s11063-015-9421-x.

[22] V. L. D. Mole and A. F. R. Araújo, "Growing self-organising surface map: learning a surface topology from a point cloud," *Neural Comput.*, vol. 22, no. 3, pp. 689-729, Mar. 2010, doi: 10.1162/neco.2009.08-08-842.

[23] S. P. Lim and H. Haron, "Applying NURBS surfaces approximation with different parameterisation methods on CKSOM model closed surfaces data," in *Image and Signal Processing*, A. Elmoataz, O. Lezoray, F. Nouboud, and D. Mammass, Eds., Cham, Switzerland: Springer, 2014, pp. 602-611, doi: 10.1007/978-3-319-07998-1_69.

[24] G. K. Knopf and A. P. Sangole, "Freeform surface reconstruction from scattered points using a deformable spherical map," *Int. J. Image Graph.*, vol. 6, no. 3, pp. 641-356, 2006, doi: 10.1142/S0219467806002343.

[25] S. P. Lim, and H. Haron, “Cube Kohonen self-organising map (CKSOM) model with new equations in organising unstructured data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1414-1424, May 2013, doi: 10.1109/TNNLS.2013.2259259.

[26] G. S. Kumar, P. K. Kalra and S. G. Dhande, “Parameter optimisation for B-spline curve fitting using genetic algorithms,” in *2003 Congress on Evolutionary Computation*, vol. 3, 2003, pp. 1871-1878, doi: 10.1109/CEC.2003.1299902.

[27] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero and H. Rostro-Gonzalez, “Hierarchical genetic algorithm for B-spline surface approximation of smooth explicit data,” *Math. Probl. Eng.*, vol. 2014, Art. no. 706247, 2014, doi: 10.1155/2014/706247.

[28] P. Pandunata, F. Forkan, and S. M. H. Shamsuddin, “Growing grid-evolutionary algorithm for surface reconstruction,” in *Proceedings of the 2013 10th International Conference Computer Graphics, Imaging and Visualisation*, 2013, pp. 68-74, doi: 10.1109/CGIV.2013.35.

[29] P. Pandunata and S. M. H. Shamsuddin, “Differential evolution optimisation for Bezier curve fitting,” in *Proceedings of the 2010 Seventh International Conference on Computer Graphics, Imaging and Visualisation*, 2010, pp. 68-72, doi: 10.1109/CGIV.2010.18.

[30] A. Gálvez and A. Iglesias, “Particle swarm optimisation for non-uniform rational B-spline surface reconstruction from clouds of 3D data points,” *Inf. Sci.*, vol. 192, pp. 174-192, June 2012, doi: 10.1016/j.ins.2010.11.007.

[31] A. Gálvez, A. Cobo, J. Puig-Pey and A. Iglesias, “Particle swarm optimisation for Bézier surface reconstruction,” in *Computational Science – ICCS 2008*, M. Bubak, G. D. van Albada, J. Dongarra and P. M. A. Sloot, Eds., Berlin, Heidelberg, Germany: Springer, 2008, pp. 116-125, doi: 10.1007/978-3-540-69387-1_13.

[32] F. Forkan and S. M. Shamsuddin, “Kohonen-swarm algorithm for unstructured data in surface reconstruction,” in *Proceedings of the Fifth International Conference on Computer Graphics, Imaging and Visualisation*, 2008, pp. 5-11, doi: 10.1109/CGIV.2008.58.

[33] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimisation over continuous spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341-359, Dec. 1997, doi: 10.1023/A:1008202821328.

[34] S. P. Lim, H. Hoon and W. H. Song, “A comparative study on different parameter factors and velocity clamping for particle swarm optimisation,” *IOP Conf. Ser.: Eng.*, vol. 864, no. 1, Art. no. 012068, 2020, doi: 10.1088/1757-899X/864/1/012068.

[35] C. H. Garcia-Capulin, G. Trejo-Caballero, H. Rostro-Gonzalez and J. G. Avina-Cervantes, “Noisy data fitting with B-splines using hierarchical genetic algorithm,” in *23rd International Conference on Electronics, Communications and Computing*, 2013, pp. 62-66, doi: 10.1109/CONIELECOMP.2013.6525760.

[36] J. E. Lara-Ramirez, C. H. Garcia-Capulin, M. d. J. Estudillo-Ayala, J. G. Avina-Cervantes, R. E. Sanchez-Yanez and H. Rostro-Gonzalez, “Parallel hierarchical genetic algorithm for scattered data fitting through B-splines,” *Applied Sciences*, vol. 9, no. 11, Art. no. 2336, 2019.

[37] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero and H. Rostro-Gonzalez, “A hierarchical genetic algorithm approach for curve fitting with B-splines,” *Genet. Program. Evolvable Mach.*, vol. 16, no. 2, pp. 151-166, 2015, doi: 10.1007/s10710-014-9231-3.

[38] S. Wang et al., “Surface reconstruction from unoriented point clouds by a new triangle selection strategy,” *Comput. Graph.*, vol. 84, pp. 144-159, Nov. 2019, doi: 10.1016/j.cag.2019.08.002.

[39] R. Li, X. Li, C. -W. Fu, D. Cohen-Or and P. -A. Heng, “PU-GAN: a point cloud upsampling adversarial network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7202-7211, doi: 10.1109/ICCV.2019.00730.

[40] R. L. M. E. do Rêgo, A. F. R. Araújo, and F. B. de Lima Neto, “Growing self-reconstruction map,” *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 211-223, Feb. 2010, doi: 10.1109/TNN.2009.2035312.

[41] Y. Yu, “Surface reconstruction from unorganised points using self-organising neural networks,” in *Proc. IEEE Vis.*, vol. 99, Jan. 1999, pp. 61-64.

[42] W. P. Lee, S. Hasan, S. M. Shamsuddin, and N. Lopes, “GPUMLib: deep learning SOM library for surface reconstruction,” *Int. J. Adv. Soft Compu. Appl.*, col. 9, no. 2, pp. 1-16, July. 2017.

[43] F. Boudjemaï, P. B. Enberg, and J.-G. Postaire, “Surface modelling by using self organising maps of Kohonen,” in *Proc. 2003 IEEE Int. Conf. Syst. Man Cybern.*, vol. 3, pp. 2418-2423, 2003, doi: 10.1109/ICSMC.2003.1244246.

[44] S. P. Lim, “Multiple 2D self organising map network for surface reconstruction of 3D unstructured data,” Ph.D. dissertation, Universiti Teknologi Malaysia, 2015.

[45] R. Daud, M. R. Abdul Kadir, S. Izman, A. P. Md Saad, M. H. Lee, and A. Che Ahmad, “Three-dimensional morphometric study of the trapezium shape of the trochlea tali,” *J. Foot Ankle Surg.*, vol. 52, no. 4, pp. 426-431, July 2013, doi: 10.1053/j.jfas.2013.03.007.

[46] *The Stanford 3D scanning repository*, Stanford University Computer Graphics Laboratory, Aug. 2014. [Online]. Available: http://graphics.stanford.edu/data/3Dscanrep/

[47] J. Barhak, A. Fischer, “Parameterisation and reconstruction from 3-D scattered points based on neural network and PDE techniques,” *IEEE Trans. Vis. Comput. Graph.*, vol. 7, no. 1, pp. 1-16, Jan-March 2001, doi: 10.1109/2945.910817.

[48] J. Morel, A. Bac, and C. Véga, “Surface reconstruction of incomplete datasets: a novel poisson surface approach based on CSRBF,” *Comput. Graph.*, vol. 74, pp. 44-45, Aug. 2018, doi: 10.1016/j.cag.2018.05.004.

[49] J. Zhang, J. Duan, K. Tang, J. Cao, and X. Liu, “Point cloud normal estimation by fast guided least squares representation,” *IEEE Access*, vol. 8, pp. 101580-101590, May 2020, doi: 10.1109/ACCESS.2020.2998468.

[50] M. Berger et al., “A survey of surface reconstruction from point clouds,” *Comput Graph Forum*, vol. 36, no. 1, pp. 301-329, Jan. 2017, doi: 10.1111/cgf.12802.

[51] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, “Deep geometric prior for surface reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10122–10131. doi:10.1109/CVPR.2019.01037.

[52] H. K. Zhao, S. Osher, and R. Fedkiw, “Fast surface reconstruction using the level set method,” in *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001, pp. 194-201, doi: 10.1109/VLSM.2001.938900.

[53] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: learning continuous signed distance functions for shape representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 165-174, doi: 10.1109/CVPR.2019.00025.

[54] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganised point clouds,” *Comput. Graph.*, vol. 26, no. 2, pp. 71-78, July 1992, doi: 10.1145/142920.134011.

[55] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’96)*, Aug. 1996, pp. 303-312, doi: 10.1145/237170.237269.

[56] P. Mullen, F. de Goes, M. Desbrun, D. Cohen-Steiner, and P. Alliez, “Signed the unsigned: robust surface reconstruction from raw pointsets,” *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1733-1741, 2010, doi: 10.1111/j.1467-8659.2010.01782.x.

[57] F. Calakli and G. Taubin, “SSD: smooth signed distance surface reconstruction,” *Comput. Graph. Forum*, vol. 30, no. 7, pp. 1993-2022, Nov. 2011, doi: 10.1111/j.1467-8659.2011.02058.x.

[58] Y. Tang and J. Feng, “Multi-scale surface reconstruction based on a curvature-adaptive signed distance field,” *Comput. Graph.*, vol. 70, pp. 28-38, Feb. 2018, doi: 10.1016/j.cag.2017.07.015.

[59] J. C. Carr et al., “Reconstruction and representation of 3D objects with radial basis function,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Technique (SIGGRAPH ’01)*, Aug. 2001, pp. 67-76, doi: 10.1145/383259.383266.

[60] J. Yang, J. Wang, Z., C. Zhu, and Q. Peng, “Implicit surface reconstruction with radial basis functions,” in *Computer Vision and Computer Graphics. Theory and Applications,* J. Braz, A. Ranchordas, H. J. Araújo, and J. M. Pereira, Eds., Berlin, Heidelberg, Germany: Springer, 2008, pp. 5-12, doi: 10.1007/978-3-540-89682-1_1.

[61] X. Y. Liu, H. Wang, C. S. Chen, Q. Wang, X. Zhou and Y. Wang, “Implicit surface reconstruction with radial basis functions via PDEs,” *Eng. Anal. Bound. Elem.*, vol. 110, pp. 95-103, Jan. 2020, doi: 10.1016/j.enganabound.2019.09.021.

[62] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. P. Seidel, “Multi-level partition of unity implicits,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463-470, July 2003, doi: 10.1145/882262.882293.

[63] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, June 2006, pp. 61-70, doi: 10.2312/SGP/SGP06/061-070.

[64] M. Kazhdan and H. Hoppe, “Screened Poisson surface reconstruction,” ACM Trans. Graph., vol. 32, no. 3, pp. 1-13, June 2013, doi: 10.1145/2487228.2487237.

[65] M. Kazhdan, “Reconstruction of solid models from oriented point sets,” in *Eurographics Symposium on Geometry Processing*, M. Desbrun and H. Pottmann, Eds., The Eurographics Association, 2005, doi: 10.2312/SGP/SGP05/073-082.

[66] J. Manson, G. Petrova, and S. Schaefer, “Streaming surface reconstruction using wavelets,” *Comput. Graph. Forum,* vol. 27, no. 5, pp. 1411-1420, 2008, doi: 10.1111/j.1467-8659.2008.01281.x.

[67] X. Ren et al., “Biorthogonal wavelet surface reconstruction using partial integrations,” *Comput. Graph. Forum*, vol. 37, no. 2, pp. 13-24, Oct. 2018, doi: 10.1111/cgf.13543.

[68] N. Amenta, M. Bern, and M. Kamvysselis, “A new Voronoi-based surface reconstruction algorithm,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH’98)*, July 1998, pp. 415-421, doi: 10.1145/280814.280947.

[69] N. Amenta, S. Choi, T. K. Dey, and N. Leekha, “A simple algorithm for homeomorphic surface reconstruction,” in *Proceedings of the Sixteenth Annual Symposium on Computational Geometry (SCG ‘00)*, May 2000, pp. 212-222, doi: 10.1145/336154.336207.

[70] N. Amenta, S. Choi, and R. K. Kolluri, “The power crust,” in *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, May 2001, pp. 249-266, doi: 10.1145/376957.376986.

[71] T. K. Dey, J. Giesen, and J. Hudson, “Delaunay based shape reconstruction from large data,” in *Proceedings IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*, 2001, pp. 19-27, doi: 10.1109/PVGS.2001.964399.

[72] T. K. Dey and S. Goswami, “Tight Cocone: a water-tight surface reconstructor,” *J. Comput. Inf. Sci. Eng.*, vol. 3, no. 4, pp. 302-307, Dec. 2003, doi: 10.1115/1.1633278.

[73] T. K. Dey, R. Dyer, and L. Wang, “Localised Cocone surface reconstruction,” *Comput. Graph.*, vol. 35, no. 3, pp. 483-491, June 2011, doi: 10.1016/j.cag.2011.03.014.

[74] A. Gálvez and A. Iglesias, “Firefly algorithm for polynomial Bézier surface parameterisation,” *J. Appl. Math.*, vol. 2013, Sept. 2013, Art. no. 237984, doi: 10.1155/2013/237984.

[75] Y. Kineri, M. Wang, H. Lin, and T. Maekawa, “B-spline surface fitting by iterative geometric interpolation/approximation algorithms,” *Comput Aided Des*, vol. 44, no. 7, pp. 697-708, July 2012, doi: 10.1016/j.cad.2012.02.011.

[76] E. K. Ueda and M. S. G. Tsuzuki, “Piecewise Bézier curve fitting of a point cloud boundary by simulated annealing,” in *Proceedings of the*

13th IEEE International Conference on Industry Applications (INDUSCON), 2018, pp. 1335-1340, doi: 10.1109/INDUSCON.2018.8627161.

[77] M. Hoffmann and L. Várady, "Free-form surfaces for scattered data by neural networks," *J Geom. Graph.*, vol. 2, no. 1, pp. 1-6, 1998.

[78] A. Gálvez, A. Iglesias, A. Cobo, J. Puig-Pey, and J. Espinola, "Bézier curve and surface fitting of 3D point clouds through genetic algorithms, functional networks and least-squares approximation," in *Computational Science and its Applications – ICCSA 2007*, O. Gervasi and M. L. Gavrilova, Eds., Berlin, Heidelberg: Springer, 2007, pp. 680-693, doi: 10.1007/978-3-540-74477-1_62.

[79] M. Ammad and A. Ramli, "Cubic B-Spline curve interpolation with arbitrary derivatives on its data points," in *Proceedings of the 23rd International Conference in Information Visualisation – Part II*, 2019, pp. 156-159, doi: 10.1109/IV-2.2019.00038.

[80] A. Gálvez and A. Iglesias, "Efficient particle swarm optimisation approach for data fitting with free knot B-splines," *Comput. Aided Des.*, vol. 43, no. 12, pp. 1683-1692, Dec. 2011, doi: 10.1016/j.cad.2011.07.010.

[81] T. R. Wang, N. Liu, L. Yuan, K. X. Wang, and X. J. Sheng, "Iterative least square optimisation for the weights of NURBS curve," *Math. Probl. Eng.*, vol. 2022, Art. no. 5690564, Apr. 2022, doi: 10.1155/2022/5690564.

[82] Q. Mao, S. Liu, S. Wang, and X. Ma, "Surface fitting for quasi scattered data from coordinate measuring systems," *Sensors*, vol. 18, no. 1, Art. no. 214, Jan 2018, doi: 10.3390/s18010214.

[83] H. Zhou, B. Feng, Z. Liu, H. Chang, and X. Cheng, "NURBS-based parametric design for shop hull form," *J. Mar. Sci. Eng.*, vol. 10, no. 5, May 2022, Art no. 686, doi: 10.3390/jmse10050686.

[84] H. B. Jung and K. Kim, "A new parameterisation method for NURBS surface interpolation," *Int. J. Adv. Manuf. Technol.*, vol. 16, no. 11, pp. 784-790, Sept. 2000, doi: 10.1007/s001700070012.

[85] W. Ma and J.-P. Kruth, “NURBS curve and surface fitting for reverse engineering,” *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 12, pp. 918-927, Dec. 1998, doi: 10.1007/BF01179082.

[86] L. Yang, and X. M. Zeng, “Bézier curves and surfaces with shape parameters,” *International Journal of Computer Mathematics*, vol. 86, no. 7, pp. 1253-1263, July 2009, doi: 0.1080/00207160701821715.

[87] H. Haron, A. Rehman, D. I. S. Adi, S. P. Lim and T. Saba, “Parameterisation method on B-spline curve,” *Math. Probl. Eng.*, vol. 2012, Art. no. 640472, 2012, doi: 10.1155/2012/640472.

[88] E. T. Y. Lee, “Choosing nodes in parametric curve interpolation,” *Comput. Aided Des.*, vol. 21, no. 6, pp. 363-370, July 1989, doi: 10.1016/0010-4485(89)90003-1.

[89] C. Balta, S. Öztürk, M. Kuncan and I. Kandilli, “Dynamic centripetal parameterisation method for B-spline curve interpolation,” *IEEE Access*, vol. 8, pp. 589-598, 2019, doi: 10.1109/ACCESS.2019.2961412.

[90] J. Zhang, S. Feng and H. Cui, “Optimised NURBS curve and surface fitting using simulated annealing,” in *Proceedings of the Second International Symposium on Computational Intelligence and Design*, 2009, pp. 324-329, doi: 10.1109/ISCID.2009.227.

[91] J. Song et al., “Unorganised point classification for robust NURBS surface reconstruction using a point-based neural network,” *J. Comput. Des. Eng.*, vol. 8, no. 1, Feb. 2021, doi: 10.1093/jcde/qwaa086.

[92] T. Kohonen “The self-organising map,” *Proc. IEEE*, vol. 78, no. 9, pp. 1464-1480, Sept. 1990, doi: 10.1109/5.58325.

[93] S. M. Guthikonda. “Kohonen self-organising maps.” Academia.edu. https://www.academia.edu/7880511/Kohonen_self_organizing_maps_shyam_guthikonda (accessed Jul. 31, 2022).

[94] T. Kohonen, *Self-organisation and associative memory*. Springer, 1989. Accessed: Apr. 28, 2023. [Online]. doi: 10.1007/978-3-642-88163-3.

[95] G. T. Breard, “Evaluating self-organising map quality measures as convergence criteria,” M.S. thesis, Computer Science and Statistics, 2017.

[96] C. C. Hsu and S. H. Lin, "Visualised analysis of mixed numeric and categorical data via extended self-organising map," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 72-86, Jan. 2012, doi: 10.1109/TNNLS.2011.2178323.

[97] S. Ng and M. Chan, "Effect of neighbourhood size selection in SOM-based image feature extraction," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 2, pp. 195-200, Apr. 2019, doi: 10.18178/ijmlc.2019.9.2.786.

[98] A. de Medeiros Brito, A. Dória Neto, J. Dantas de Melo, and L. M. Garcia Gonçalves, "An adaptive learning approach for 3-D surface reconstruction from point clouds," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 1130-1140, June 2008, doi: 10.1109/TNN.2008.2000390.

[99] V. Chaudhary, R. S. Bhatia, and A. K. Ahlawat, "The self-organising map learning algorithm with inactive and relative winning frequency of active neurons," *HKIE Trans.*, vol. 21, no. 1, pp. 62-67, 2014, doi: 10.1080/1023697X.2014.883680.

[100] N. J. Mount and D. Weaver, "Self-organising maps and boundary effects: quantifying the benefits of torus wrapping for mapping SOM trajectories," *Pattern Anal. Applic.*, vol. 14, no. 2, pp. 139-148, Mar. 2011, doi: 10.1007/s10044-011-0210-5.

[101] A. Sen, B. Suleymanoglu, and M. Soycan, "Unsupervised extraction of urban features from airborne lidar data by using self-organising maps," *Surv. Rev.*, vol. 52, no. 371, pp. 150-158, 2020, doi: 10.1080/00396265.2018.1532704.

[102] F. Boudjemaï, P. B. Enberg, and J. G. Postaire, "Dynamic adaptation and subdivision in 3D-SOM application to surface reconstruction," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '05)*, 2005, pp. 425-430, doi: 10.1109/ICTAI.2005.61.

[103] M. Hoffmann, "Modified Kohonen neural network for surface reconstruction," *Publicationes Mathematicae*, vol. 54, pp. 857-864, 1999.

[104] C. Y. Liou and Y. T. Kuo, “Conformal self-organising map for a genus-zero manifold,” *Vis. Comput.*, vol. 21, no. 5, pp. 340-353, June 2005, doi: 10.1007/s00371-005-0290-6.

[105] C. Y. Liou and W. P. Tai, “Conformal self-organisation for continuity on a feature map,” *Neural Netw.*, vol. 2, no. 6, pp. 893-905, July 1999, doi: 10.1016/S0893-6080(99)00034-9.

[106] C. Y. Liou, Y. T. Kuo, and J. C. Huang, “Smooth seamless surface construction based on conformal self-organising map,” in *Neural Information Processing*, I. King, J. Wang, L.-W. Chan, and D. Wang, Eds., Berlin, Heidelberg, Germany: Springer, 2006, pp. 1012-1021, doi: 10.1007/11893028_113.

[107] S. N. Sivanandam and S. N. Deepa, “Terminologies and Operators of GA,” in *Introduction to Genetic Algorithms*, Berlin, Heidelberg: Springer, 2007, ch. 4, pp. 39-81, doi: 10.1007/978-3-540-73190-0.

[108] N. M. Razali and J. Geraghty, “Genetic algorithm performance with different selection strategies in solving TSP,” in *Proceedings of the World Congress on Engineering*, vol. 2, 2011.

[109] D. E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” in *Foundation of Genetic Algorithms*, vol. 1, G. J. E. Rawlins, Ed, Elsevier, 1991, pp. 69-93, doi: 10.1016/B978-0-08-050684-5.50008-2.

[110] A. Shukla, H. M. Pandey, and D. Mehtrotra, “Comparative review of selection techniques in genetic algorithm,” in *Proceedings of the 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, 2015, pp. 515-519, doi: 10.1109/ABLAZE.2015.7154916.

[111] S. Mirjalili, J. S. Dong, A. S. Sadiq, and H. Faris, “Genetic algorithm: theory, literature review, and application in image reconstruction,” in *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*, S. Mirjalili, J. S. Dong, and A. Lewis, Eds., Cham, Switzerland: Springer, 2020, pp. 69-85, doi: 10.1007/978-3-030-12127-3_5.

[112] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, “Choosing mutation and crossover ratios for genetic algorithms – a review with a new dynamic approach,” *Information*, vol. 10, no. 12, Art. no. 390, Dec. 2019, doi: 10.3390/info10120390.

[113] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091-8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[114] V. Kachitvichyanukul, “Comparison of three evolutionary algorithms: GA, PSO, and DE,” *Ind. Eng. Manag. Syst.*, vol. 11, no. 3, pp. 215-223, Sept. 2012, doi: 10.7232/iems.2012.11.3.215.

[115] N. Soni and T. Kumar, “Study of various mutation operators in genetic algorithm,” *Int. J. Comput. Sci. Inf. Technol. Res.*, vol. 5, no. 3, pp. 4519-4521, 2014.

[116] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4-31, Feb. 2011, doi: 10.1109/TEVC.2010.2059031.

[117] T. Eltaeib and A. Mahmood, “Differential evolution: a survey and analysis,” *Appl. Sci.*, vol. 8, no. 10, Art. no. 1945, Oct. 2018, doi: 10.3390/app8101945.

[118] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, and J. Xu, “An improved differential evolution algorithm and its application in optimisation problem,” *Soft Comput.*, vol. 25, pp. 5277-5298, Apr. 2021, doi: 10.1007/s00500-020-05527-x.

[119] R. Eberhart and J. Kennedy, “Particle swarm optimisation,” in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942-1948, doi: 10.1109/ICNN.1995.488968.

[120] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proc. IEEE Int. Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, 1998, pp. 69-73, doi: 10.1109/ICEC.1998.699146.

[121] M. Clerc, “The swarm and the queen: towards a deterministic and adaptive particle swarm optimisation,” in *Proc. Congr. Evol. Comput.*, vol. 3, 1999, pp. 1951-1957, doi: 10.1109/CEC.1999.785513.

[122] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimisation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009, doi: 10.1109/TSMCB.2009.2015956.

[123] R. C. Eberhart, and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimisation," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, 2000, pp. 84 - 88, doi: 10.1109/CEC.2000.870279.

[124] E. T. Oldewage, A. P. Engelbrecht, and C. W. Cleghorn, "The merit of velocity clamping particle swarm optimisation in high dimensional spaces," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1-8, doi: 10.1109/SSCI.2017.8280887.

[125] A. Gálvez, A. Iglesias, and J. Puig-Pey, "Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction," *Inf. Sci.*, vol. 182, no. 1, pp. 56-76, Jan. 2012, doi: 10.1016/j.ins.2010.09.031.

[126] E. Madeley, "Talus," KenHub, 2022. [Online]. Available: https://www.kenhub.com/en/library/anatomy/talus. [Accessed 22 August 2022].

[127] J. Alejandrino, E. Trinidad, R. Concepcion, E. Sybingco, M. G. Palconit, L. Materum and E. Dadios, "Utilization of self-organizing maps for map depiction of multipath clusters," in *Intelligent Computing and Optimization*, 2022.

[128] W. Natita, W. Wiboonsak, and S. Dusadee, "Appropriate learning rate and neighbourhood function of self-organising map (SOM) for specific humidity pattern classification over southern Thailand," *Int. J. Model. Optim.*, vol. 6, no. 1, pp. 61–65, doi: 10.7763/IJMO.2016.V6.504.

[129] W. Zhang, J. Wang, D. Jin, L. Oreopoulos, and Z. Zhang, "A deterministic self-organising map approach and its application on satellite data based cloud type classification," in *Proceedings of the 2018 IEEE International Conference on Big Data*, 2018, pp. 2027–2034, doi: 0.1109/BigData.2018.8622558.

[130] R. Mancini, A. Ritacco, G. Lanciano, and T. Cucinotta, "XPySom: High-Performance Self-Organizing Maps," in *Proceedings of the 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2020, pp. 209–216, doi: 10.1109/SBAC-PAD49847.2020.00037.

[131] V. Govindaraj, A. Thiyagarajan, P. Rajasekaran, Y. Zhang, and R. Krishnasamy, "Automated unsupervised learning-based clustering approach for effective anomaly detection in brain magnetic resonance imaging (MRI)," *IET Image Process.*, vol. 14, no. 14, pp. 3516–3526, Nov. 2020, doi: 10.1049/iet-ipr.2020.0597.

[132] M. N. M. Othman, Y. Yusoff, H. Haron, and L. You, "An overview of surface reconstruction using partial differential equation (PDE)," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 551, no. 1, Aug. 2019, Art. no. 012054, doi: 10.1088/1757-899X/551/1/012054.

[133] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 1997. Accessed: Apr. 28, 2023. [Online]. doi: 10.1007/978-3-642-59223-2.

[134] D. Jiang and L. Wang, "An algorithm of NURBS surface fitting for reverse engineering," *Int. J. Adv. Manuf. Technol.*, vol. 31, no. 1, pp. 92-97, Nov. 2006, doi: 10.1007/s00170-005-0161-3.

[135] A. Iglesias, A. Gálvez and M. Collantes, "Multilayer embedded bat algorithm for B-spline curve reconstruction," *Integr. Comput. Aided Eng.*, vol. 24, no. 4, pp. 385-399, Sept. 2017, doi: 10.3233/ICA-170550.

[136] C.-K. Shene, "Parameters and knot vectors for surfaces," pages.mtu.edu. https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/INT-APP/PARA-surface.html (accessed Mar. 3, 2023).

[137] A. B. Makhlouf, N. Elloumi, B. Louhichi and D. Deneux, "Approach for CAD model reconstruction basing on 3D points insertion and surface approximation," in *Proceedings of the 2021 25th International Conference Information Visualisation*, 2021, pp. 310-319, doi: 10.1109/IV53921.2021.00057.

[138] M. Sarfraz and M. Riyazuddin, "Curve fitting with NURBS using simulated annealing," in *Applied Soft Computing Technologies: The Challenge of Complexity*, A. Abraham, B. de Baets, M. Köppen and B.

Nickolay, Eds., Berlin, Heidelberg, Germany: Springer, 2006, pp. 99 – 112, doi: 10.1007/3-540-31662-0_8.

[139] L. A. Piegl and W. Tiller, “Least-squares B-spline curve approximation with arbitrary end derivatives,” *Engineering with Computers*, vol. 16, no. 2, pp. 109-116, doi: 10.1007/PL00007188.

[140] D. I. S. Adi, S. M. Shamsuddin and A. Ali, “Particle swarm optimisation for NURBS curve fitting,” in *Proceedings of the 2009 Sixth International Conference on Computer Graphics, Imaging and Visualisation*, 2009, pp. 259-263, doi: 10.1109/CGIV.2009.64.

[141] S. P. Lim, H. Hoon and C. H. Ong, "Performance comparison of different operation techniques in genetic algorithm towards benchmark functions," in *Proceedings of the 8th IEEE international conference on control system, computing and engineering*, 2018, pp. 59-64.

[142] S. P. Lim and H. Haron, “Performance comparison of genetic algorithm, differential evolution and particle swarm optimization towards benchmark functions,” in *2013 IEEE Conference on Open Systems (ICOS)*, 2013, pp. 41-46, doi: 10.1109/ICOS.2013.6735045.

[143] P. Laube, M. O. Franz and G. Umlauf, “Deep learning parametrisation for B-spline curve approximation,” in *Proceedings of the 2018 International Conference on 3D Vision (3DV)*, 2018, pp. 691-699, doi: 10.1109/3DV.2018.00084.

[144] X.-S. Yang and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimisation,” *Engineering Computations*, vol. 29, no. 5, pp. 464-483, July 2012, doi: 10.1108/02644401211235834.

[145] X.-S. Yang, “Firefly algorithms for multimodal optimisation,” in *Stochastic Algorithms: Foundations and Applications*, O. Watanabe and T. Zeugmann, Eds., Berlin, Heidelberg, Germany: Springer, 2009, pp. 169-178, doi: 10.1007/978-3-642-04944-6_14.

[146] D. Pickell. “Supervised vs unsupervised learning – what’s the difference?.” g2.com. https://www.g2.com/articles/supervised-vs-unsupervised-learning (accessed Nov. 30, 2022).

[147] E. Dimas and D. Briassoulis, "3D geometric modelling based on NURBS: a review," *Adv. Eng. Softw.*, vol. 30, no. 9, pp. 741-751, Sept. 1999, doi: 10.1016/S0965-9978(98)00110-0.

[148] A. Iglesias et al., "Cuckoo search algorithm with Lévy Flights for global-support parametric surface approximation in reverse engineering," *Symmetry*, vol. 10, no. 3, Mar. 2018, Art. no. 58, doi: 10.3390/sym10030058.

[149] Y. Zhang, G. Chen, L. Cheng, Q. Wang and Q. Li, "Methods to balance the exploration and exploitation in differential evolution from different scales: A survey," *Neurocomputing*, vol. 561, Dec. 2023, Art. no. 126899, doi: 10.1016/j.neucom.2023.126899.

[150] E. K. Nyarko, R. Cupec and D. Filko, "A comparison of several heuristic algorithms for solving high dimensional optimization problems," *International Journal of Electrical and Computer Engineering Systems*, vol. 5, no. 1, pp. 1 – 8, 2014.

[151] M. Ali and M. Pant, "Improving the performance of differential evolution algorithm using Cauchy mutation," *Soft Comput.*, vol. 15, no. 5, pp. 991 – 1007, May 2011, doi: 10.1007/s00500-010-0655-2.

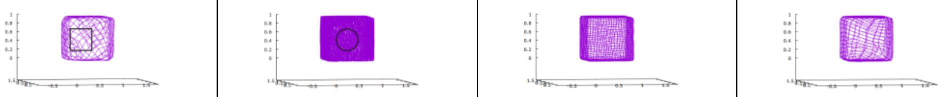
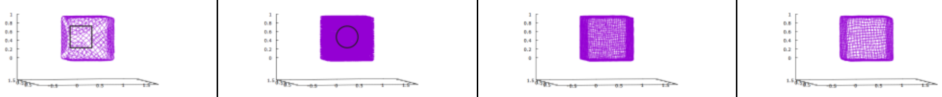
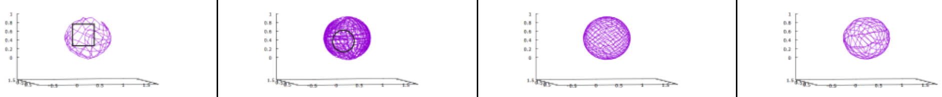
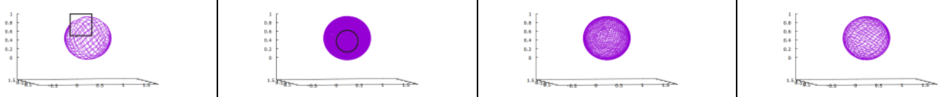
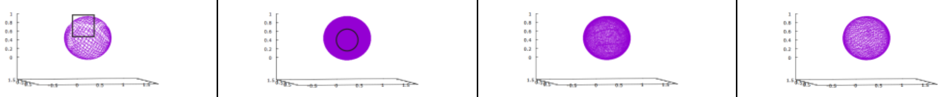
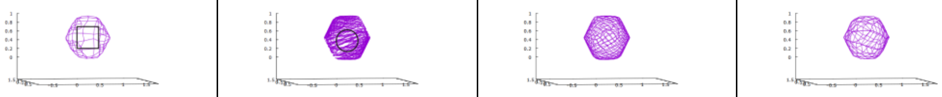
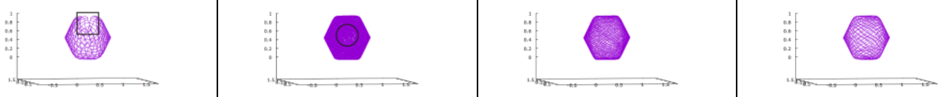
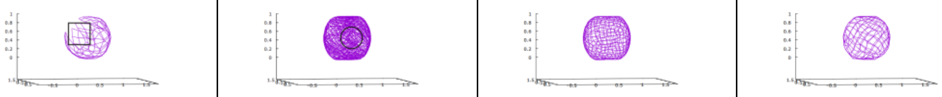
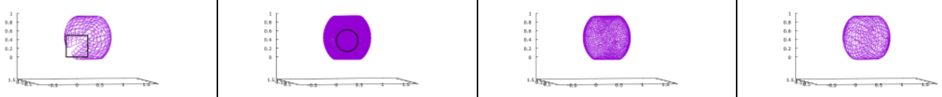
[152] M. H. Majorrad and P. Ayubi, "Particle swarm optimisation with chaotic velocity clamping (CVC-PSO)," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, 2015, pp. 1 – 6, doi: 10.1109/IKT.2015.7288811.

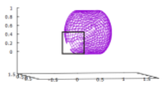
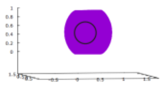
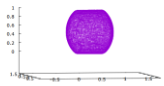
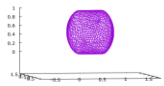
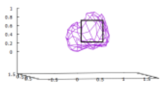
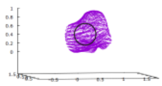
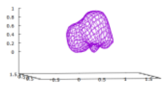
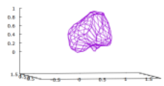
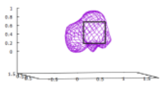
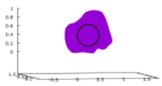
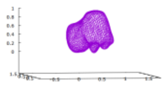
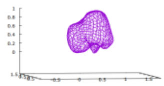
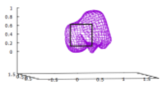
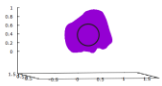
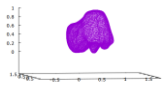
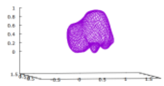
[153] S. P. Lim and H. Haron, "Performance of different techniques applied in genetic algorithm towards benchmark functions," in A. Selamat, N. T. Nguyen and H. Haron, Eds., Berlin, Heidelberg, Germany: Springer, 2013, pp. 255-264, doi: 10.1007/978-3-642-36546-1_27.

LIST OF PUBLICATIONS

1. C. C. You, S. P. Lim, S. C. Lim, J. S. Tan, C. K. Lee, and Y. M. J. Khaw, "A survey on surface reconstruction techniques for structured and unstructured data," in *2020 IEEE Conference on Open Systems (ICOS)*, 2020, pp. 37-42, doi: 10.1109/ICOS50156.2020.9293685.
2. S. P. Lim, C. K. Lee, J. S. Tan, S. C. Lim, and C. C. You, "Implementing self organising map to organise the unstructured data," *J. Phys.: Conf. Ser.*, vol. 2129, no. 1, Dec 2021, Art. no. 012046, doi: 10.1088/1742-6596/2129/1/012046.
3. C. C. You, S. P. Lim, C. K. Lee, J. S. Tan, and S. C. Lim, "Performance evaluation of self-organising map model in organising the unstructured data," in *2022 IEEE International Conference on Computing (ICOCO)*, 2022, pp. 6-11, doi: 10.1109/ICOCO56118.2022.10032038.

APPENDIX A: VISUALISATION FOR DIFFERENT SOM MODEL AND DATA SETS

Data	Grid Size, *n*	2-D SOM	3-D SOM	CKSOM	DNSOM
Cube	10				
	20				
	30				
Sphere	10				
	20				
	30				
Spindle	10				
	20				
	30				
Oiltank	10				
	20				

	30				
Talus Bone	10				
	20				
	30				

APPENDIX B: METRIC EVALUATION FOR DIFFERENT SOM MODEL AND DATA SETS

Data	Grid Size, *n*	Min Error				Max Error			
		2-D SOM	3-D SOM	CKSOM	DNSOM	2-D SOM	3-D SOM	CKSOM	DNSOM
Cube	10	0.002339	0.000932	0.000331	0.001206	0.946736	0.645208	0.613486	0.870413
	20	0.001412	0.000242	0.000132	0.000274	0.924689	0.619527	0.540496	0.805349
	30	0.000972	0.000106	0.000061	0.000255	0.904238	0.584974	0.459987	0.777626
Sphere	10	0.005445	0.001349	0.000958	0.001943	0.714700	0.426053	0.416009	0.636790
	20	0.001970	0.000581	0.000461	0.001223	0.669834	0.406771	0.382448	0.533180
	30	0.001024	0.000293	0.000111	0.000841	0.636785	0.404172	0.356824	0.520385
Spindle	10	0.004991	0.001225	0.001017	0.002142	0.750138	0.459577	0.453888	0.612911
	20	0.001050	0.000085	0.000030	0.001004	0.651950	0.436344	0.425152	0.528028
	30	0.000675	0.000025	0.000024	0.000059	0.636605	0.402486	0.401306	0.505220
Oiltank	10	0.006479	0.001904	0.000978	0.004432	0.731639	0.478516	0.441673	0.675554
	20	0.002691	0.000728	0.000683	0.001942	0.702260	0.465476	0.392644	0.557512
	30	0.001833	0.000464	0.000352	0.001115	0.615532	0.463063	0.393145	0.533917
Talus Bone	10	0.004374	0.001312	0.001264	0.001556	0.808128	0.526657	0.468956	0.668278
	20	0.002214	0.000589	0.000580	0.001173	0.769459	0.449088	0.441589	0.608013
	30	0.001159	0.000274	0.000265	0.000650	0.750302	0.411097	0.405427	0.582180

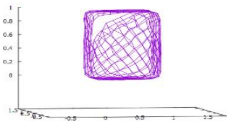
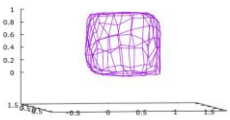
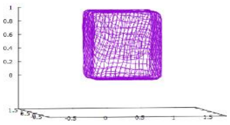
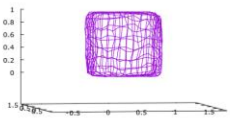
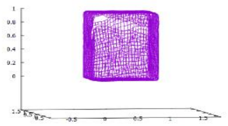
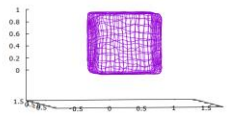
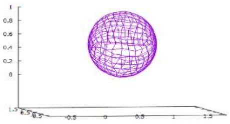
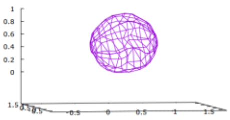
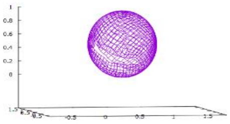
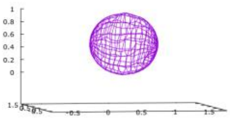
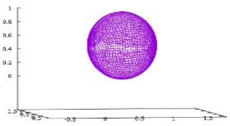
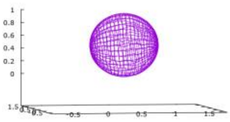
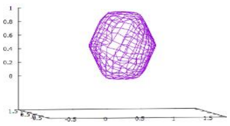
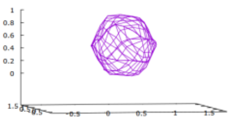
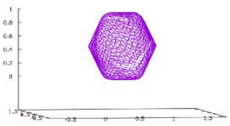
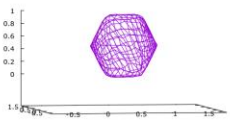
Data	Grid Size, *n*	QE				TE				CPU Time (s)			
		2-D SOM	3-D SOM	CKSOM	DNSOM	2-D SOM	3-D SOM	CKSOM	DNSOM	2-D SOM	3-D SOM	CKSOM	DNSOM
Cube	10	0.187931	0.074207	0.079324	0.138079	0.243033	0.568433	0.220400	0.218967	0.335528	2.913710	1.729668	0.489413
	20	0.126418	0.042222	0.047466	0.085877	0.229133	0.611367	0.213267	0.205167	1.493339	27.812584	8.467364	2.323140
	30	0.104045	0.032381	0.036973	0.067436	0.225167	0.623900	0.208967	0.197733	3.321563	134.344833	21.170973	4.972094
Sphere	10	0.135517	0.052833	0.055651	0.098825	0.242033	0.633200	0.255000	0.235833	0.413248	2.811455	1.680223	0.581725
	20	0.090704	0.029862	0.033220	0.061035	0.225733	0.679267	0.238733	0.212000	1.387953	26.838444	8.428613	2.284522
	30	0.074211	0.023113	0.026132	0.047605	0.225200	0.718933	0.236967	0.211433	3.182952	151.613586	19.346643	5.405772
Spindle	10	0.134581	0.053078	0.056098	0.099202	0.238033	0.634333	0.257967	0.231400	0.062082	2.822090	1.747945	0.505617
	20	0.090243	0.030270	0.033947	0.060576	0.222100	0.678600	0.251267	0.221100	1.329022	26.724297	7.936086	2.241000
	30	0.074296	0.023467	0.026625	0.047371	0.220600	0.707933	0.253800	0.218300	3.365511	189.238892	19.378843	5.309147
Oiltank	10	0.150892	0.058629	0.061965	0.109936	0.244400	0.651467	0.235933	0.219400	0.380674	2.810697	1.700103	0.538561
	20	0.101191	0.033420	0.036973	0.067115	0.241367	0.693233	0.230733	0.203767	1.541303	27.456812	8.537663	2.200606

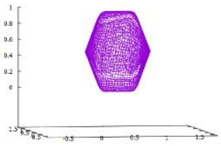
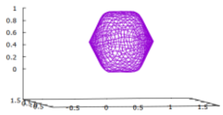
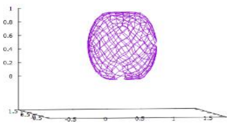
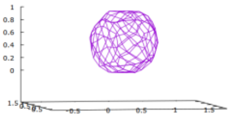
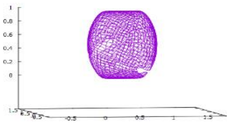
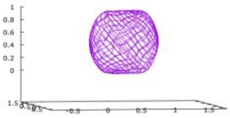
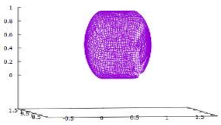
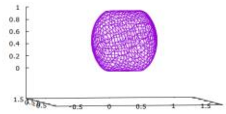
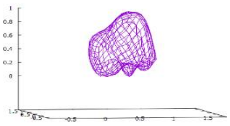
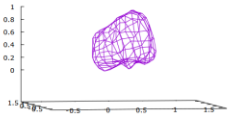
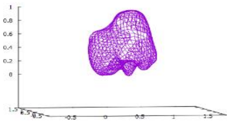
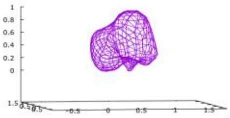
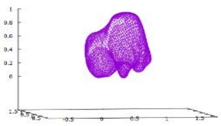
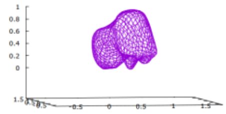
	30	0.082984	0.025624	0.028992	0.052517	0.240433	0.716633	0.222767	0.193733	3.267899	193.515076	20.582836	5.293958
Talus Bone	10	0.142370	0.057148	0.061301	0.104913	0.239433	0.598300	0.257233	0.232033	0.329207	3.050086	1.687947	0.496557
	20	0.096793	0.032491	0.037149	0.065414	0.239167	0.623033	0.244967	0.218333	1.439106	28.803869	7.594567	2.297114
	30	0.079069	0.025129	0.029380	0.052135	0.225433	0.674800	0.240133	0.214100	3.295924	186.662079	21.240513	5.264114

APPENDIX C: METRIC EVALUATION FOR CKSOM AND DNSOM MODEL WITH VARIOUS SIZES OF WIDTH (n_x) AND LENGTH (n_y) OF GRID AND DATA SETS

Data	n_x	n_y	Min Error		Max Error		QE		TE		CPU Time	
			CKSOM	DNSOM	CKSOM	DNSOM	CKSOM	DNSOM	CKSOM	DNSOM	CKSOM	DNSOM
Cube	10	8	0.000849	0.001217	0.829653	0.748264	0.094651	0.145361	0.351967	0.236300	1.750415	0.419171
	20	12	0.000658	0.000536	0.657015	0.698404	0.061053	0.091301	0.348533	0.231400	5.056553	1.330686
	18	30	0.000253	0.000443	0.630518	0.616849	0.048754	0.069085	0.348433	0.220867	12.350095	3.027687
Sphere	10	8	0.001571	0.003588	0.569852	0.588397	0.067360	0.103998	0.342167	0.242533	1.659040	0.455780
	20	12	0.000757	0.001082	0.477261	0.520727	0.043373	0.063982	0.341700	0.225967	5.094960	1.334136
	18	30	0.000466	0.000819	0.400630	0.474567	0.033999	0.047869	0.339367	0.208333	13.210606	3.153576
Spindle	10	8	0.001051	0.003025	0.575847	0.556720	0.068216	0.103635	0.354700	0.239133	1.807986	0.428606
	20	12	0.000602	0.001107	0.493366	0.534296	0.043618	0.063878	0.350600	0.227667	5.579232	1.391338
	18	30	0.000078	0.000270	0.450791	0.463108	0.034528	0.048394	0.350400	0.222067	13.621258	3.096171
Oiltank	10	8	0.001449	0.004716	0.585763	0.602103	0.075364	0.115772	0.348967	0.232967	1.552437	0.434447
	20	12	0.000995	0.001430	0.490223	0.597031	0.048225	0.072605	0.342900	0.232800	5.071808	1.291023
	18	30	0.000685	0.000740	0.426784	0.543275	0.037788	0.054163	0.342667	0.219600	12.610228	3.073069
Talus Bone	10	8	0.002246	0.001825	0.743837	0.591497	0.073348	0.111010	0.360433	0.248467	1.684480	0.421401
	20	12	0.001059	0.000885	0.460263	0.505954	0.047542	0.068569	0.359667	0.247133	5.629281	1.299775
	18	30	0.000447	0.000753	0.447727	0.421743	0.037655	0.052197	0.357467	0.244133	13.947246	3.186108

APPENDIX D: VISUALISATION OF CKSOM AND DNSOM MODELS FOR VARIOUS DATA SETS USING DIFFERENT WIDTH AND LENGTH OF GRID

Data	Grid Size	CKSOM	DNSOM
Cube	10 × 8		
	20 × 12		
	18 × 30		
Sphere	10 × 8		
	20 × 12		
	18 × 30		
Spindle	10 × 8		
	20 × 12		

	18 × 30		
Oiltank	10 × 8		
	20 × 12		
	18 × 30		
Talus Bone	10 × 8		
	20 × 12		
	18 × 30		

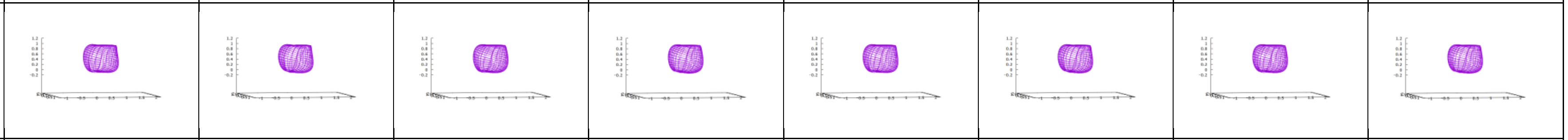
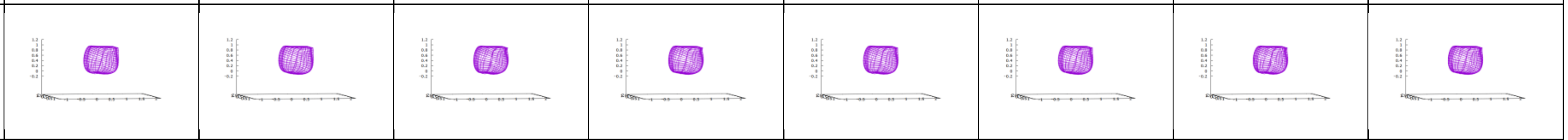
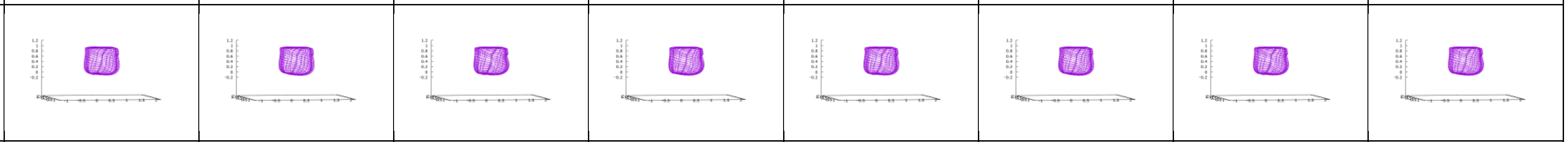
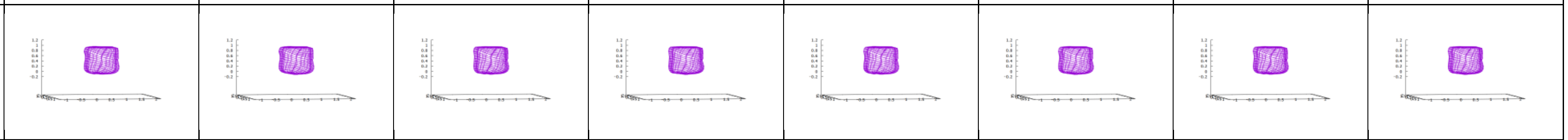
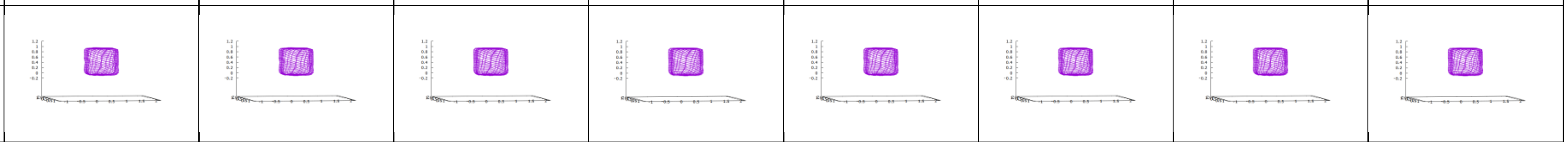
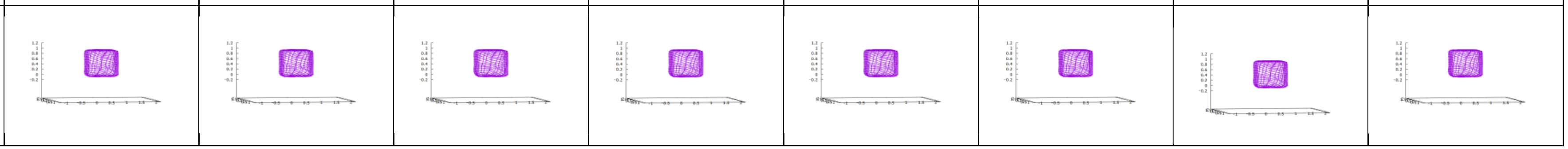
APPENDIX E: SURFACE ERROR OF CONVENTIONAL (*A*) AND IMPROVED (*B*) APPROACH FOR THE CN WITH THE SAME WIDTH AND LENGTH

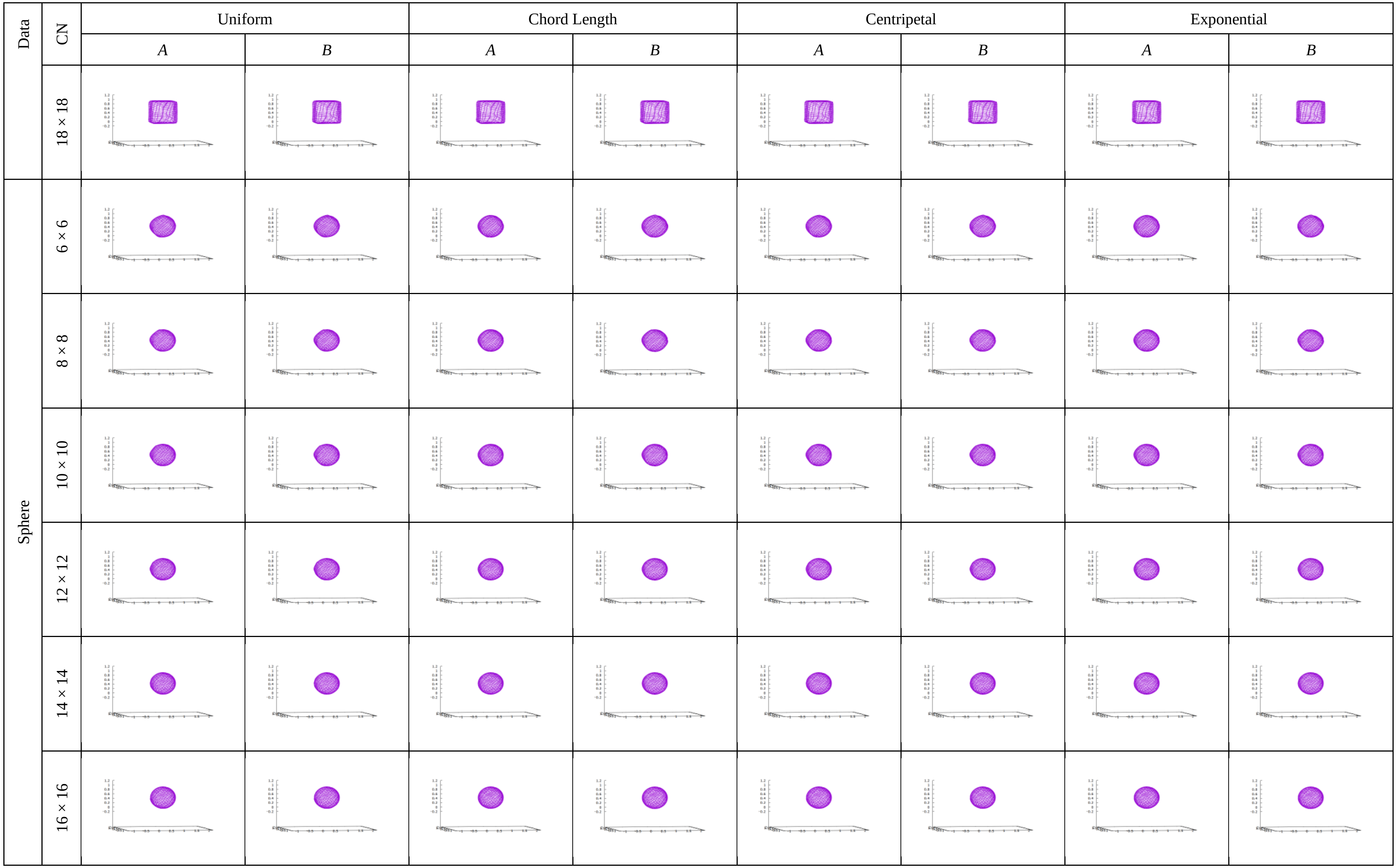
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
Cube	6 × 6	30.965167	32.847252	30.636514	28.750110	30.373630	29.935218	30.402689	28.952805
	8 × 8	23.997210	23.989527	25.349100	23.294933	24.107798	22.707770	24.699217	22.833759
	10 × 10	19.257970	18.809090	20.439738	19.142886	19.280049	18.199131	19.840364	18.608152
	12 × 12	15.359828	14.743104	15.599429	14.770796	14.993467	14.135343	15.243678	14.375113
	14 × 14	10.887558	10.320493	10.267518	9.495545	10.147066	9.429036	10.129432	9.372678
	16 × 16	7.187893	6.762235	6.319849	5.790958	6.310254	5.873534	6.218519	5.735893
	18 × 18	3.713430	3.489357	3.086438	2.818562	3.054088	2.844669	3.016986	2.781822
Sphere	6 × 6	18.577789	20.204454	14.486324	15.357078	15.432899	16.432150	14.525555	15.386218
	8 × 8	14.137940	14.560331	11.361610	11.483114	11.856062	11.901011	11.295439	11.358364
	10 × 10	11.319547	11.072429	9.588623	9.260175	9.717226	9.326677	9.437529	9.097750
	12 × 12	9.249949	8.769725	7.824251	7.401567	7.915990	7.451317	7.696266	7.272313
	14 × 14	7.370246	6.826963	5.653392	5.380782	6.030013	5.667360	5.677499	5.387289
	16 × 16	5.562201	5.121749	3.774779	3.548917	4.252580	3.975747	3.863775	3.627790
	18 × 18	2.972712	2.803381	2.033189	1.934270	2.241047	2.126338	2.067013	1.966122
Spindle	6 × 6	17.236641	16.618302	15.567186	15.317307	15.949442	15.585659	15.626590	15.344327
	8 × 8	14.547514	14.029662	13.576166	13.190979	13.733906	13.317136	13.560586	13.173216
	10 × 10	11.810122	11.427097	11.072895	10.650167	11.174899	10.780897	11.040477	10.637621
	12 × 12	9.486557	9.053436	8.754341	8.318483	8.894491	8.474809	8.749209	8.323326
	14 × 14	7.103165	6.704238	6.267812	5.905985	6.484181	6.126793	6.302085	5.946391
	16 × 16	4.886699	4.583246	3.969280	3.768353	4.191150	3.971729	3.998704	3.797467
	18 × 18	2.658669	2.514656	1.985376	1.916830	2.141893	2.050128	2.008200	1.936817
Oiltank	6 × 6	21.373959	22.129226	19.722526	19.660172	20.003289	20.215953	19.646872	19.651440
	8 × 8	17.242362	17.706700	15.737389	15.407933	15.965394	15.906888	15.656382	15.392426
	10 × 10	13.097058	13.164787	11.660973	11.206745	11.877594	11.575495	11.584774	11.151213
	12 × 12	9.945930	9.669914	8.620465	8.182572	8.811125	8.406920	8.560507	8.123378
	14 × 14	7.424647	7.025431	5.959859	5.659485	6.300094	5.980997	5.989466	5.689804
	16 × 16	4.917906	4.602955	3.546750	3.368280	3.888654	3.673360	3.602344	3.412576
	18 × 18	2.853839	2.702826	1.964867	1.892390	2.172211	2.069978	2.003908	1.921040
Talus Bone	6 × 6	28.117199	27.184267	26.532124	25.675751	27.057877	26.202397	26.662685	25.831448
	8 × 8	20.313741	19.791257	19.416785	18.593093	19.473586	18.876482	19.335066	18.626535
	10 × 10	14.853240	14.466086	15.041301	14.150769	14.560862	13.965549	14.758073	13.996782
	12 × 12	11.665266	11.274231	11.540746	10.854089	11.262056	10.768179	11.348336	10.754879
	14 × 14	8.072549	7.631480	6.988868	6.618030	7.256829	6.920247	7.021873	6.676524
	16 × 16	5.570709	5.218926	4.477426	4.252484	4.820020	4.560049	4.561460	4.326967
	18 × 18	2.989422	2.833904	2.142126	2.045113	2.382017	2.273533	2.197927	2.099644

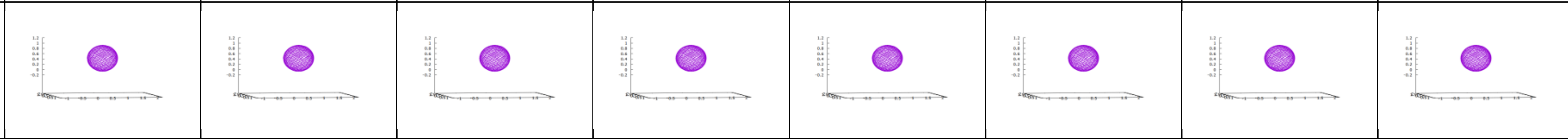
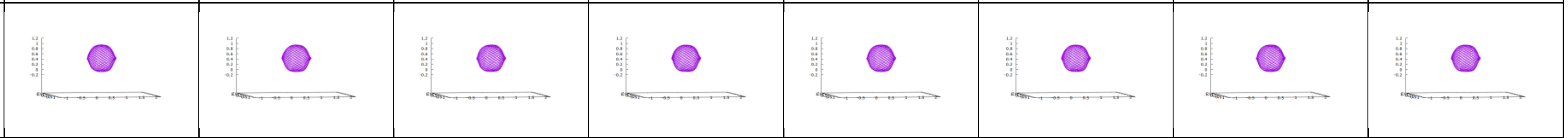
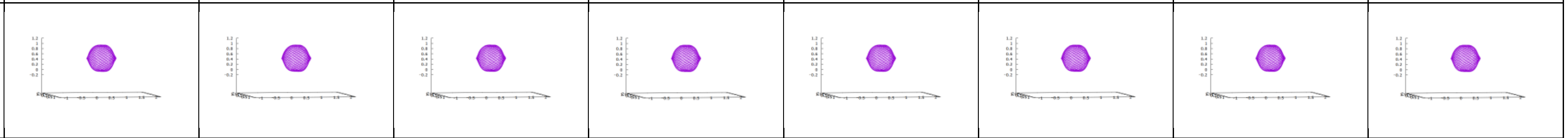
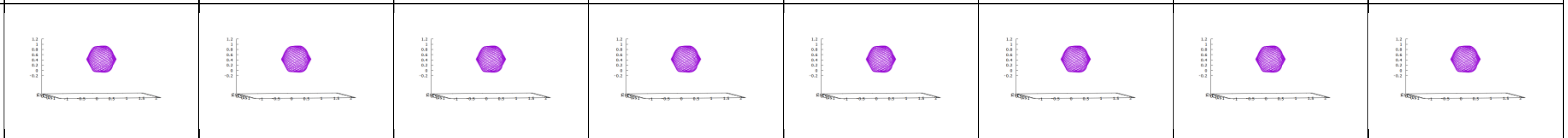
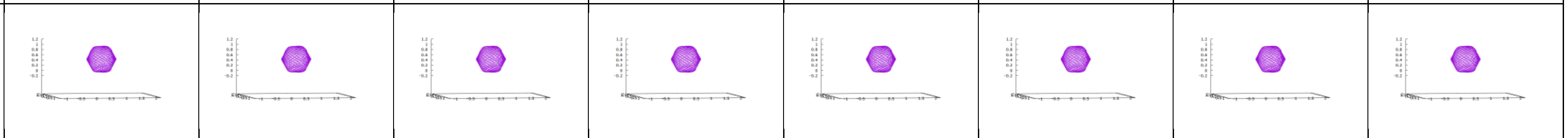
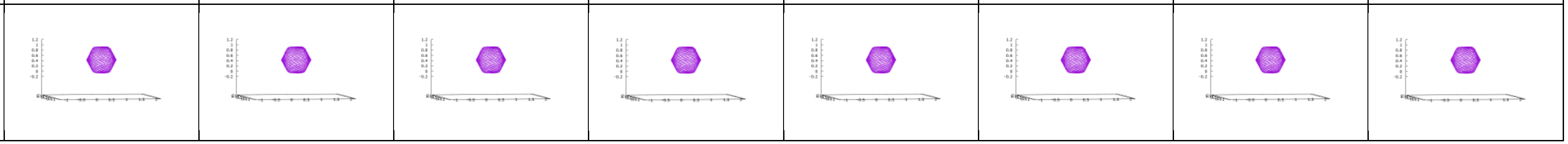
APPENDIX F: SURFACE ERROR OF CONVENTIONAL (*A*) AND IMRPOVED (*B*) APPROACH FOR THE CN WITH DIFFERENT WIDTH AND LENGTH

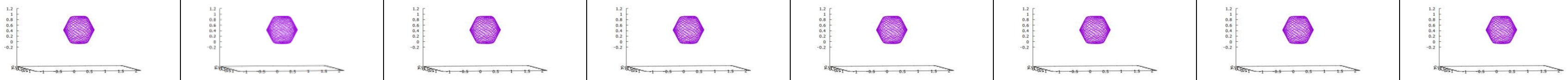
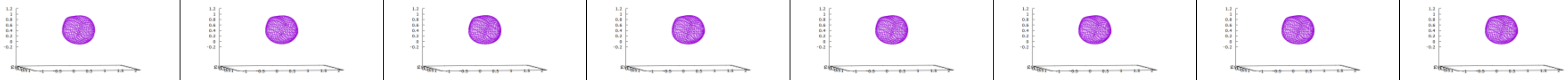
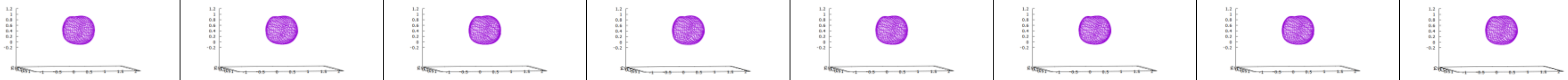
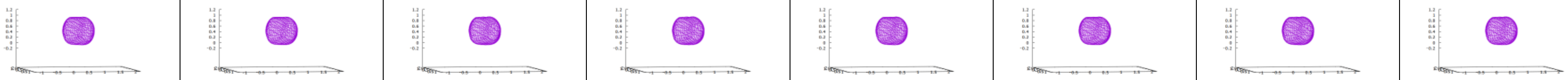
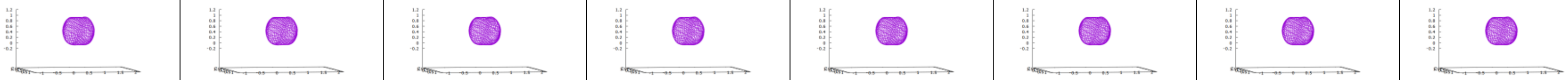
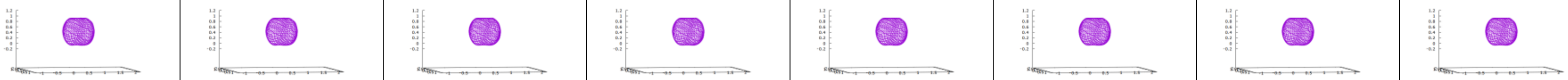
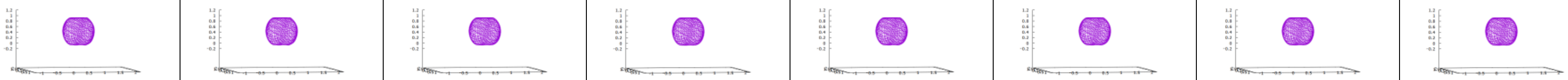
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
Cube	4 × 16	40.230698	45.796012	38.255321	38.416722	38.687754	41.098115	38.251124	39.102375
	6 × 18	29.265079	29.282095	29.634424	28.368689	28.872274	28.055285	29.180071	28.038141
	8 × 20	20.094533	19.656191	21.668636	20.598706	20.161769	19.360944	20.895639	19.932207
	10 × 22	15.865542	15.254454	17.951800	16.926239	16.273906	15.469367	17.155825	16.221275
	12 × 24	11.516729	10.924310	12.479170	11.865611	11.557332	10.978313	12.030080	11.437806
	14 × 26	7.701103	7.262420	7.283711	6.947204	7.165188	6.827783	7.174372	6.843952
	16 × 28	4.130263	3.874686	3.040097	2.906806	3.345493	3.173775	3.116590	2.970490
Sphere	4 × 16	14.426392	14.903185	14.962448	15.469229	13.906297	13.902640	14.314098	14.539610
	6 × 18	11.415998	11.262854	11.274682	11.148058	10.954939	10.732737	11.040056	10.864263
	8 × 20	9.612656	9.378245	9.229240	9.058099	9.133723	8.908904	9.117493	8.923461
	10 × 22	7.823158	7.544603	7.329037	7.138703	7.368741	7.138793	7.295527	7.089783
	12 × 24	6.071819	5.801424	5.417908	5.243158	5.583794	5.380170	5.443785	5.258722
	14 × 26	4.349236	4.142489	3.641002	3.534287	3.846675	3.703292	3.689829	3.571697
	16 × 28	2.472867	2.364249	1.826803	1.762445	2.024669	1.946529	1.877888	1.809801
Spindle	4 × 16	19.845835	20.303638	18.724757	19.170128	18.699755	19.024116	18.548353	18.873429
	6 × 18	15.404810	15.444657	14.417751	14.050667	14.606325	14.433540	14.406941	14.087943
	8 × 20	11.967224	11.905378	11.055362	10.725676	11.222107	11.027065	11.032031	10.743960
	10 × 22	9.177770	8.973447	8.280211	8.003751	8.476576	8.254017	8.278814	8.024803
	12 × 24	6.737922	6.464457	5.860567	5.649897	6.088677	5.869683	5.892565	5.683107
	14 × 26	4.507876	4.250889	3.825630	3.649081	3.973669	3.776540	3.847059	3.665819
	16 × 28	2.469162	2.351708	2.037711	1.954901	2.124319	2.027645	2.053448	1.965280
Oiltank	4 × 16	25.846648	29.149654	24.510704	26.909804	24.394025	26.933113	24.209219	26.566622
	6 × 18	19.224721	19.829836	18.007970	17.811067	18.134828	18.275420	17.922452	17.829253
	8 × 20	14.682994	14.744166	13.680633	13.121388	13.726034	13.398294	13.565043	13.070835
	10 × 22	11.226418	10.948206	10.104141	9.595360	10.265219	9.848888	10.052864	9.580364
	12 × 24	8.359407	7.936696	7.088412	6.754982	7.349947	6.999715	7.093771	6.760065
	14 × 26	5.964788	5.592089	4.731021	4.535360	5.022123	4.765088	4.776450	4.561951
	16 × 28	3.364587	3.190289	2.481551	2.367612	2.685663	2.555727	2.516029	2.400655
Talus Bone	4 × 16	26.057345	28.027628	26.366348	26.198621	25.242394	25.535604	25.638985	25.449586
	6 × 18	18.478712	18.234296	19.795644	18.895910	18.468515	17.760659	19.105159	18.268279
	8 × 20	14.538241	14.132923	15.662584	14.782553	14.632952	13.875225	15.149266	14.300186
	10 × 22	11.377503	10.995319	11.171745	10.601987	10.912597	10.358928	10.987395	10.416476
	12 × 24	8.337822	7.964382	7.270235	6.996009	7.470290	7.170618	7.269268	6.993589
	14 × 26	5.628576	5.262074	4.661210	4.346587	4.815578	4.535553	4.653507	4.367835
	16 × 28	3.255125	3.080396	2.341418	2.253805	2.562756	2.450417	2.388454	2.295462

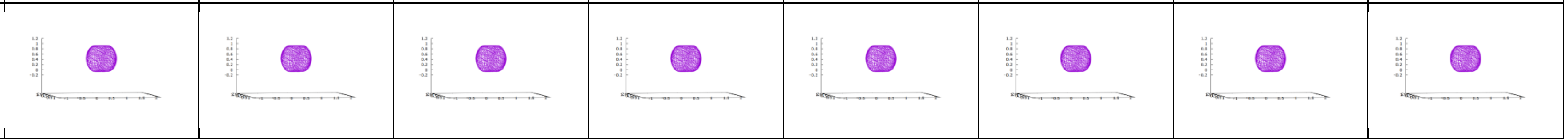
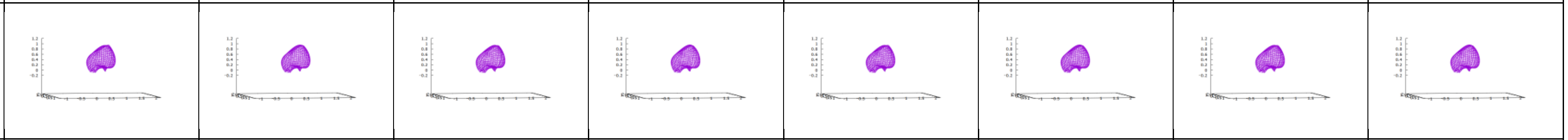
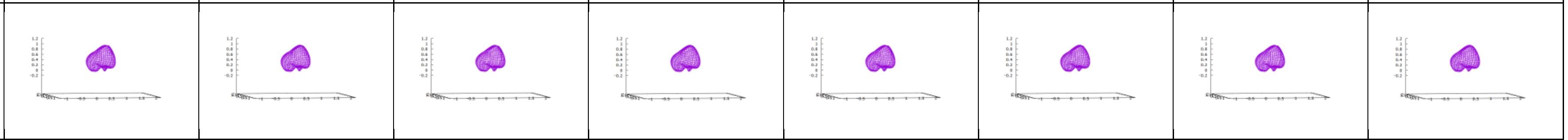
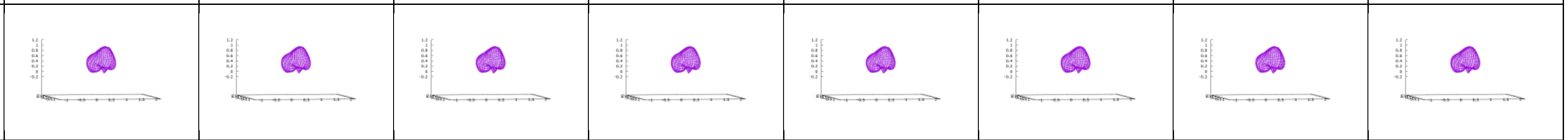
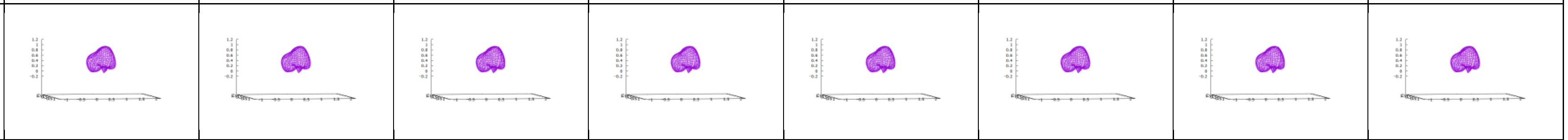
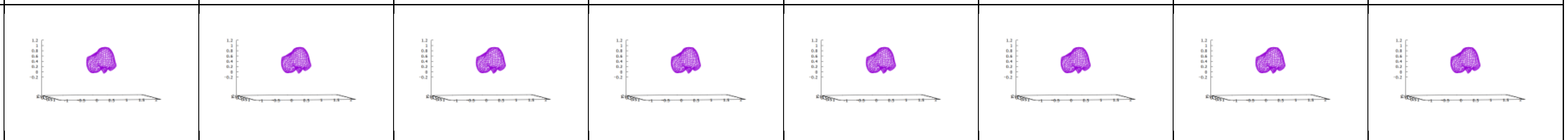
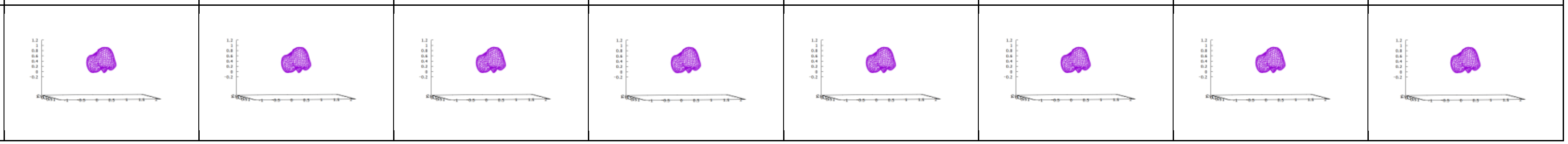
APPENDIX G: IMAGE RESULTS OF CONVENTIONAL (*A*) AND IMPROVED (*B*) SURFACE APPROXIMATION APPROACHES FOR CN WITH THE SAME WIDTH AND LENGTH

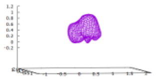
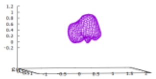
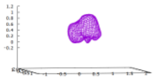
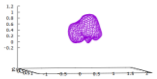
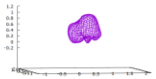
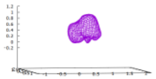
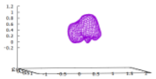
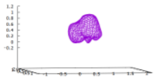
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
Cube	6 × 6								
	8 × 8								
	10 × 10								
	12 × 12								
	14 × 14								
	16 × 16								


Data
CN
Uniform
Chord Length
Centripetal
Exponential
A
B
A
B
A
B
A
B
18 × 18
Sphere
6 × 6
8 × 8
10 × 10
12 × 12
14 × 14
16 × 16

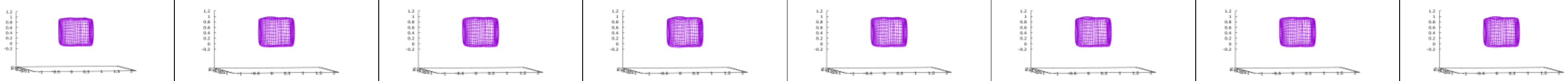
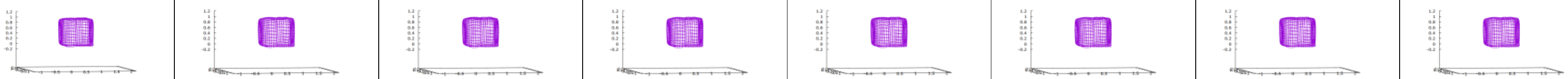
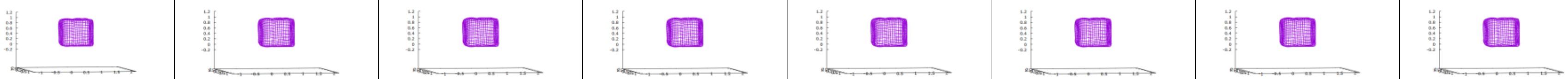
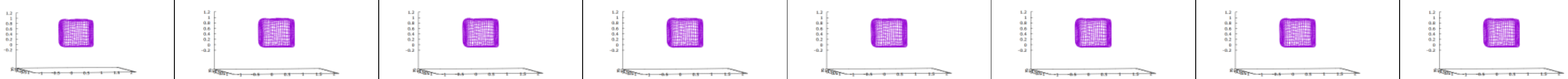
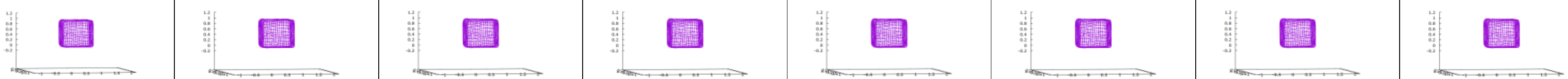
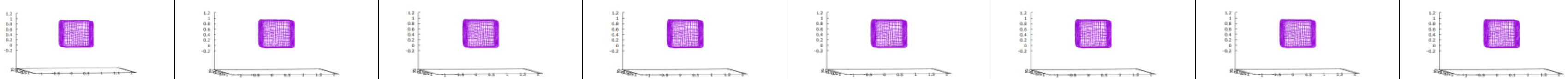
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	18 × 18								
Spindle	6 × 6								
	8 × 8								
	10 × 10								
	12 × 12								
	14 × 14								
	16 × 16								

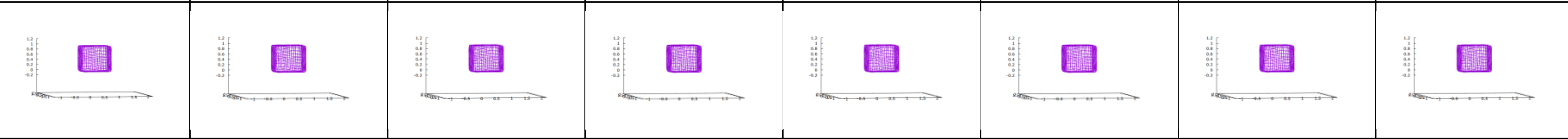
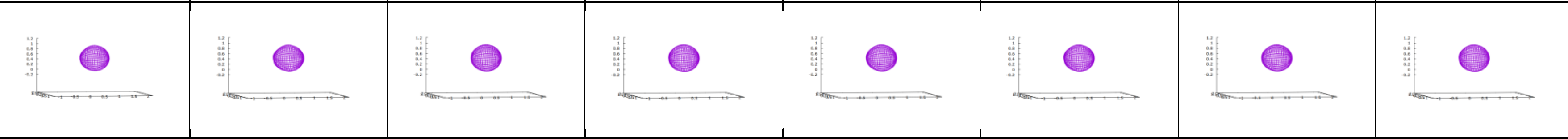
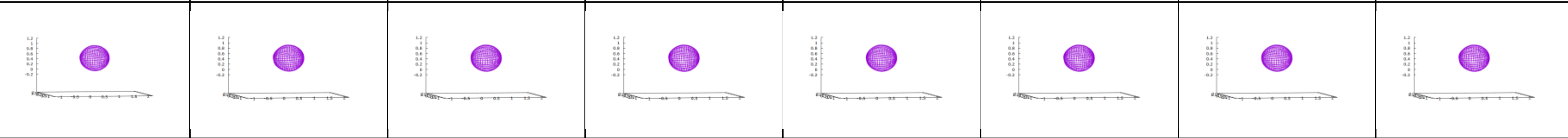
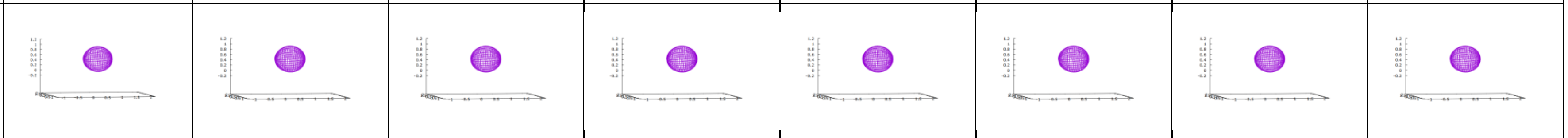
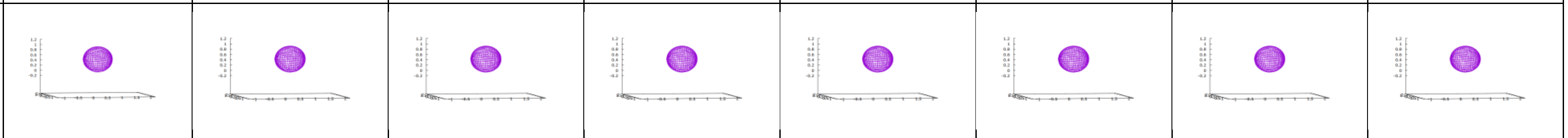
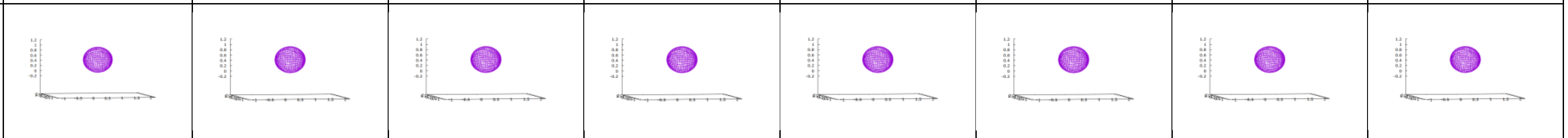
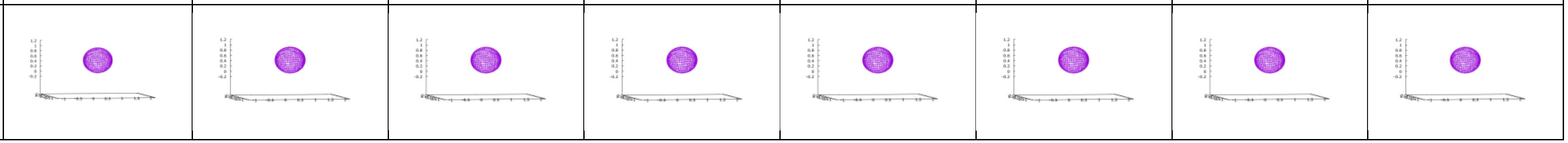
<table>
<tr><th rowspan="2">Data</th><th rowspan="2">CN</th><th colspan="2">Uniform</th><th colspan="2">Chord Length</th><th colspan="2">Centripetal</th><th colspan="2">Exponential</th></tr>
<tr><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th></tr>
<tr><td></td><td>18 × 18</td><td colspan="8"></td></tr>
<tr><td rowspan="6">Oiltank</td><td>6 × 6</td><td colspan="8"></td></tr>
<tr><td>8 × 8</td><td colspan="8"></td></tr>
<tr><td>10 × 10</td><td colspan="8"></td></tr>
<tr><td>12 × 12</td><td colspan="8"></td></tr>
<tr><td>14 × 14</td><td colspan="8"></td></tr>
<tr><td>16 × 16</td><td colspan="8"></td></tr>
</table>

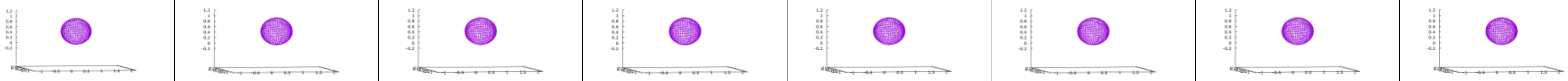
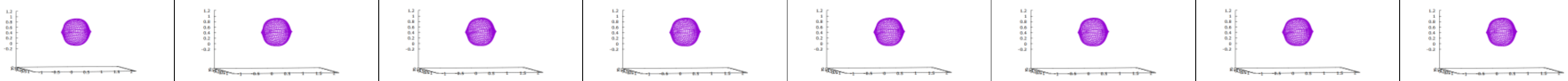
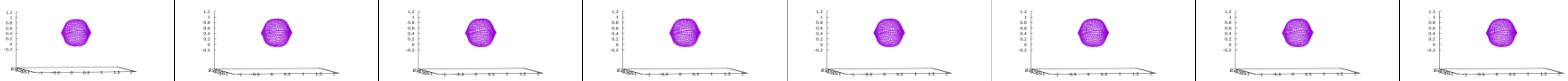
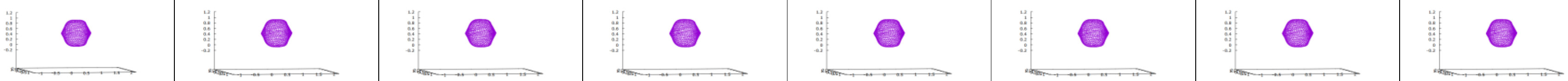
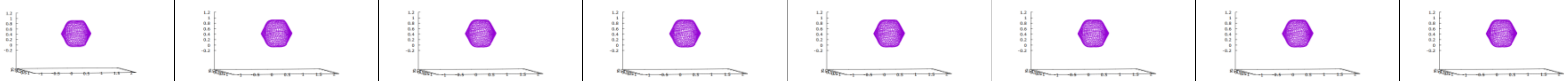
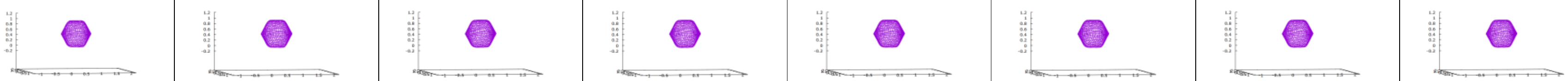
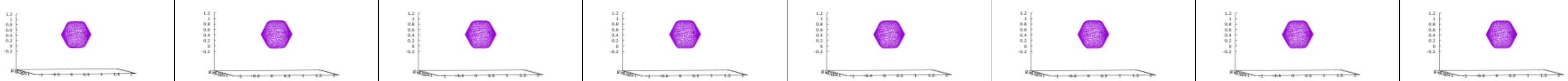
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	18 × 18								
Talus Bone	6 × 6								
	8 × 8								
	10 × 10								
	12 × 12								
	14 × 14								
	16 × 16								

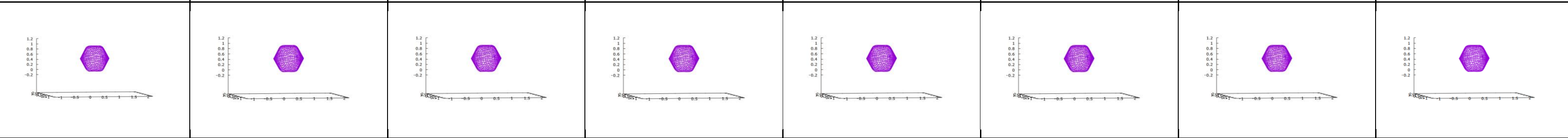
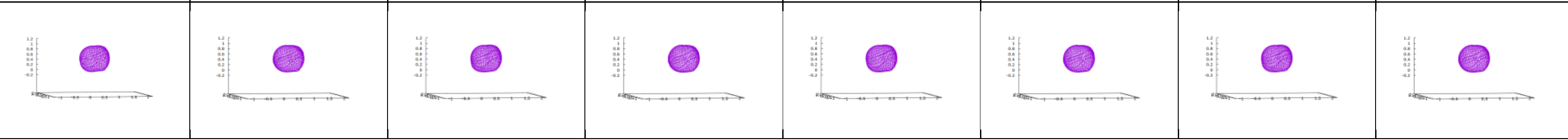
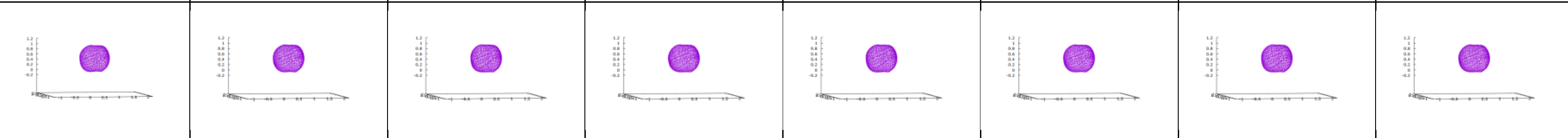
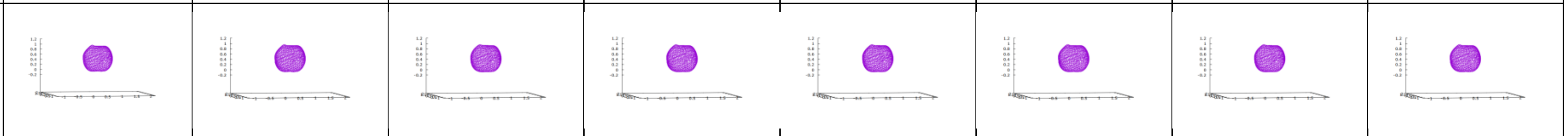
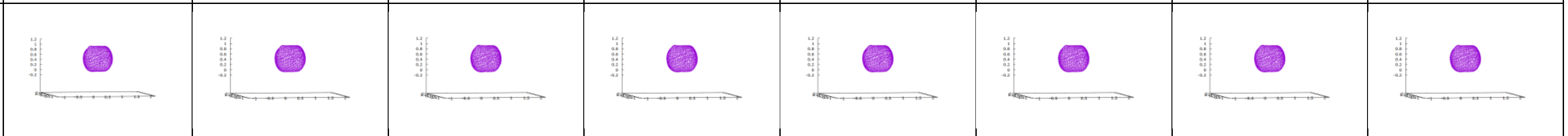
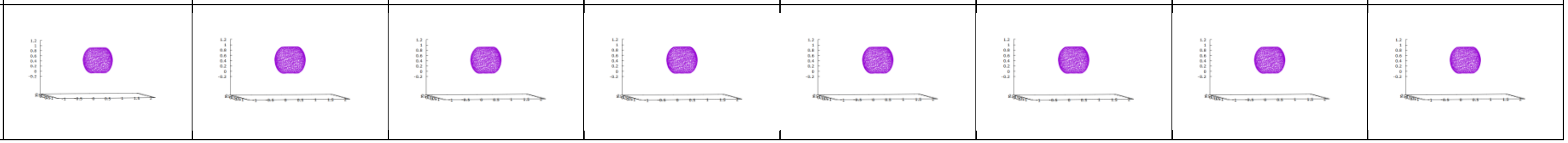
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	18×18								

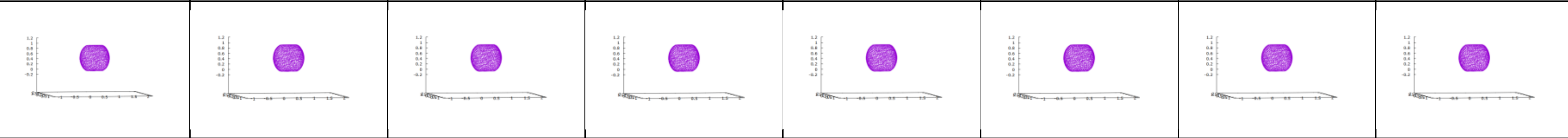
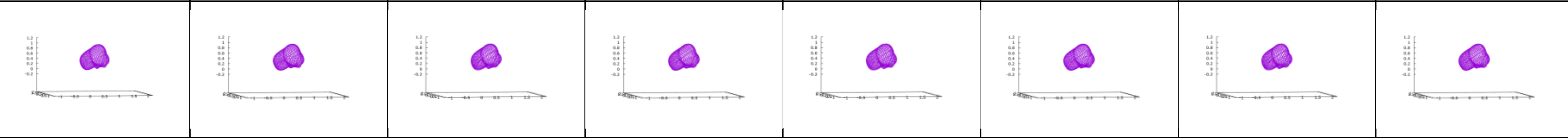
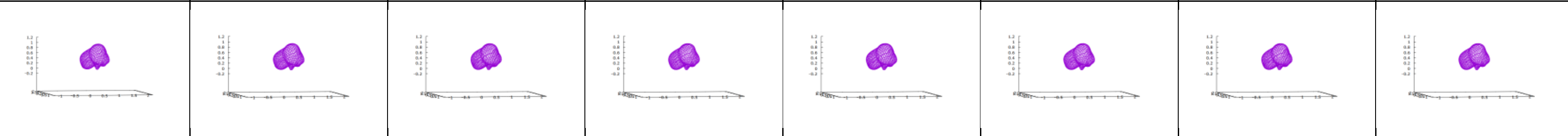
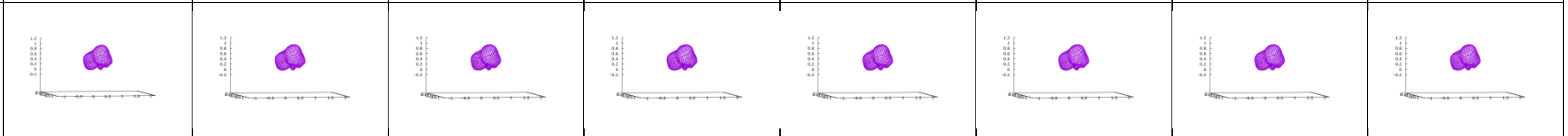
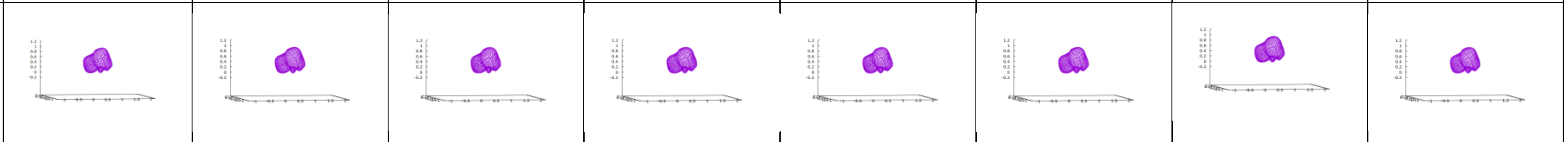
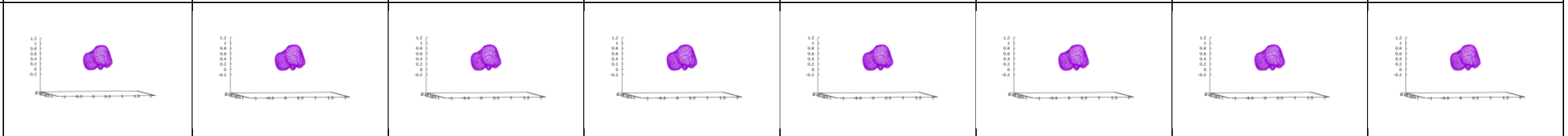
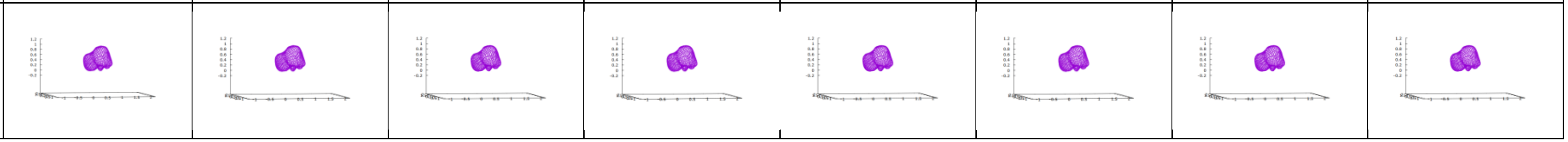
APPENDIX H: IMAGE RESULTS OF THE CONVENTIONAL (*A*) AND IMPROVED (*B*) NURBS SURFACE APPROXIMATION APPROACHES FOR THE CN WITH DIFFERENT WIDTH AND LENGTH

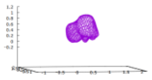
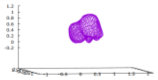
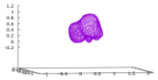
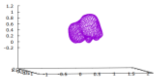
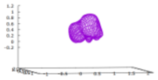
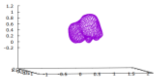
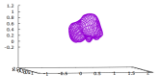
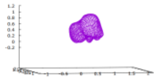
<table>
<tr><th rowspan="2">Data</th><th rowspan="2">CN</th><th colspan="2">Uniform</th><th colspan="2">Chord Length</th><th colspan="2">Centripetal</th><th colspan="2">Exponential</th></tr>
<tr><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th></tr>
<tr><td rowspan="6">Cube</td><td>4 × 16</td><td colspan="8"></td></tr>
<tr><td>6 × 18</td><td colspan="8"></td></tr>
<tr><td>8 × 20</td><td colspan="8"></td></tr>
<tr><td>10 × 22</td><td colspan="8"></td></tr>
<tr><td>12 × 24</td><td colspan="8"></td></tr>
<tr><td>14 × 26</td><td colspan="8"></td></tr>
</table>

Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	16 × 28								
Sphere	4 × 16								
	6 × 18								
	8 × 20								
	10 × 22								
	12 × 24								
	14 × 26								

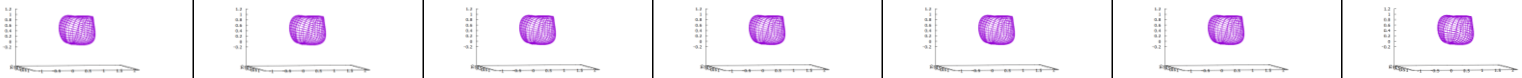
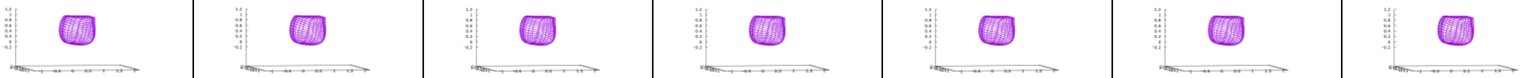
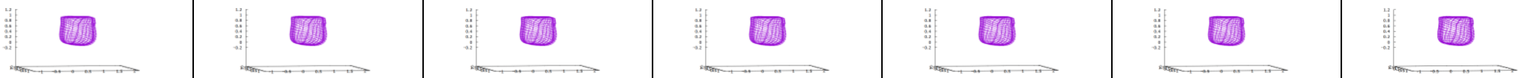
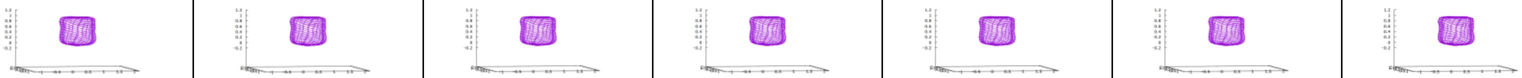
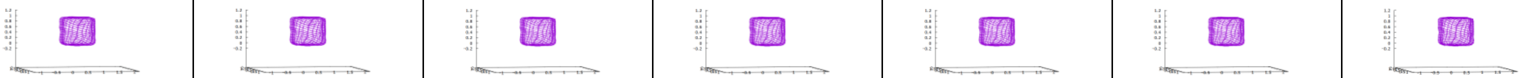
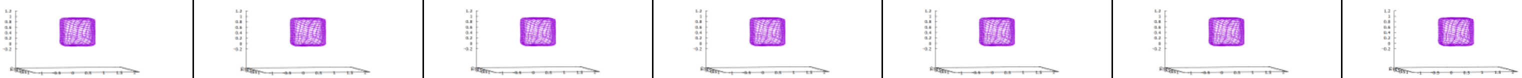
<table>
<tr><th rowspan="2">Data</th><th rowspan="2">CN</th><th colspan="2">Uniform</th><th colspan="2">Chord Length</th><th colspan="2">Centripetal</th><th colspan="2">Exponential</th></tr>
<tr><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th><th>A</th><th>B</th></tr>
<tr><td></td><td>16 × 28</td><td colspan="8"></td></tr>
<tr><td rowspan="6">Spindle</td><td>4 × 16</td><td colspan="8"></td></tr>
<tr><td>6 × 18</td><td colspan="8"></td></tr>
<tr><td>8 × 20</td><td colspan="8"></td></tr>
<tr><td>10 × 22</td><td colspan="8"></td></tr>
<tr><td>12 × 24</td><td colspan="8"></td></tr>
<tr><td>14 × 26</td><td colspan="8"></td></tr>
</table>

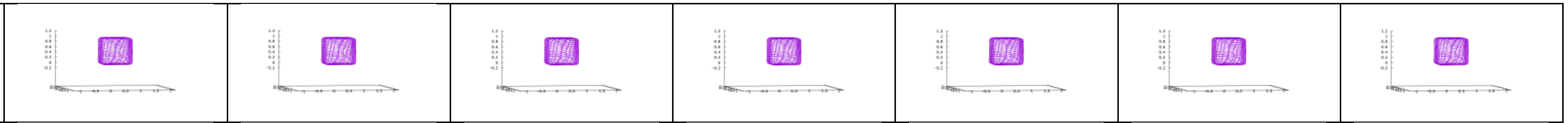
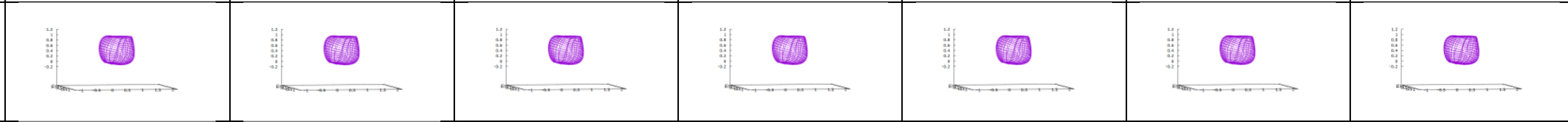
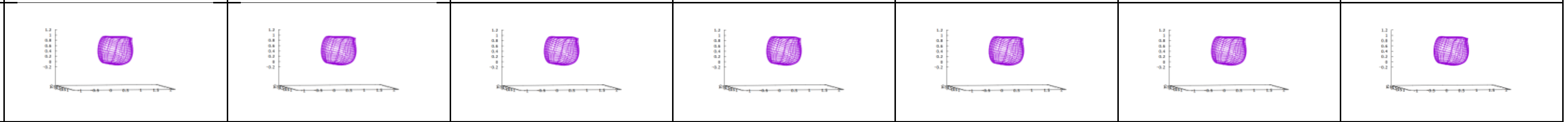
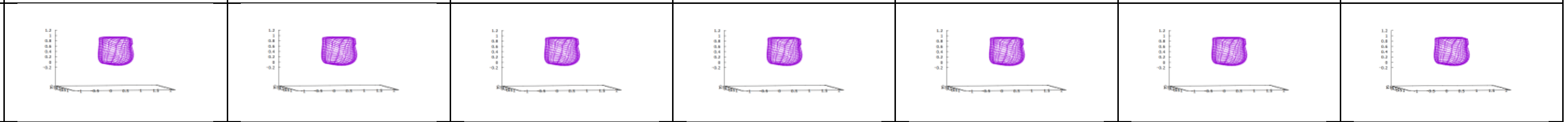
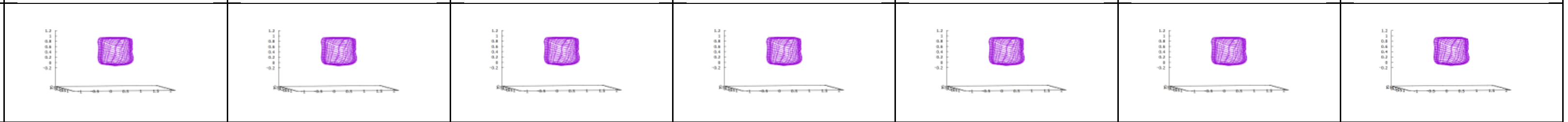
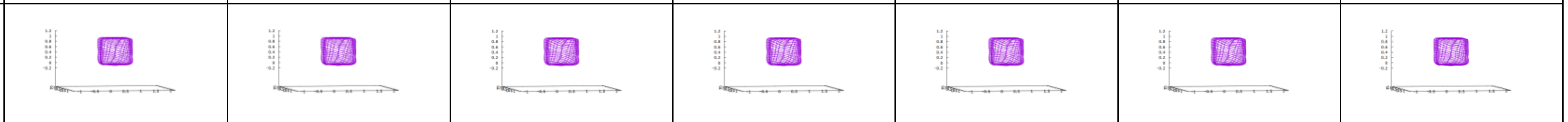
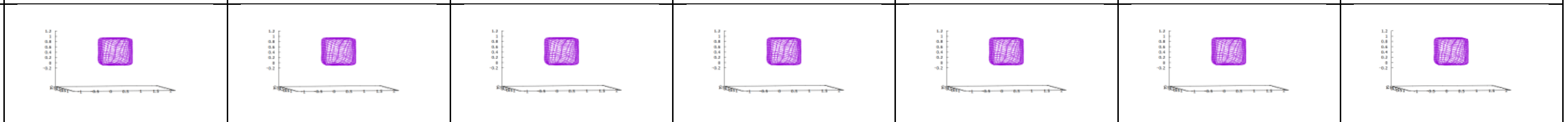
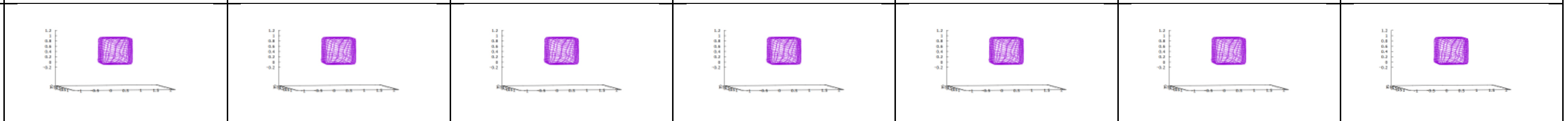
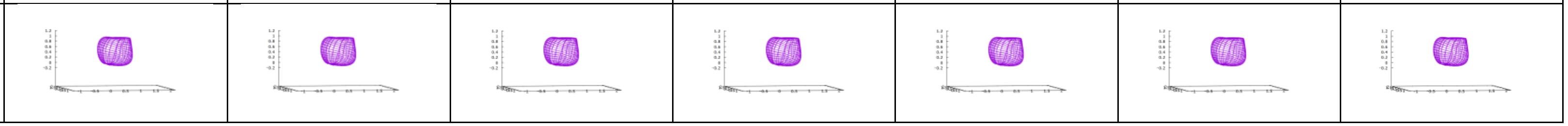
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	16 × 28								
Oiltank	4 × 16								
	6 × 18								
	8 × 20								
	10 × 22								
	12 × 24								
	14 × 26								

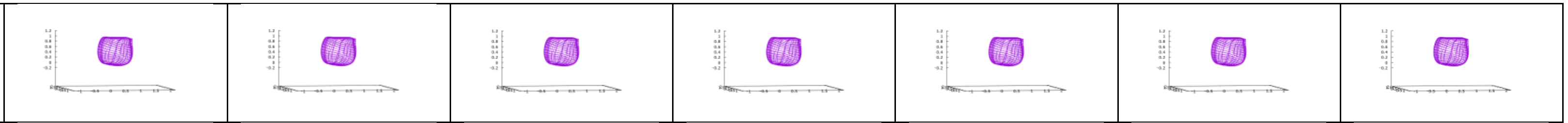
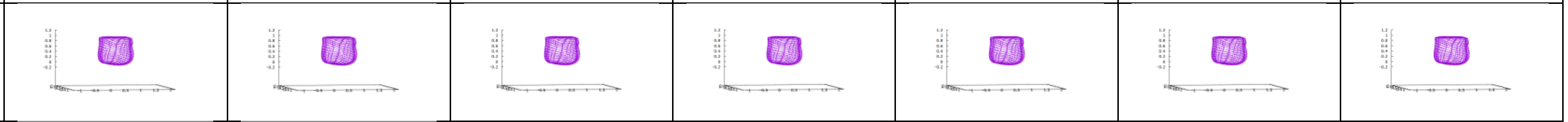
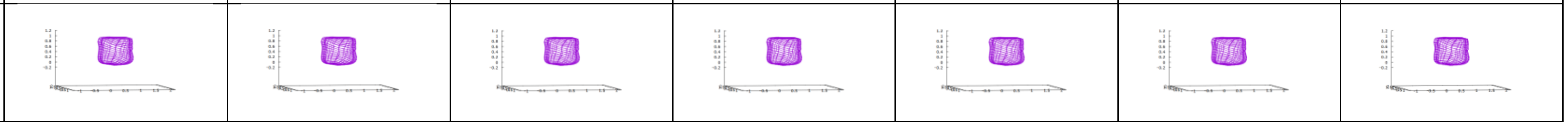
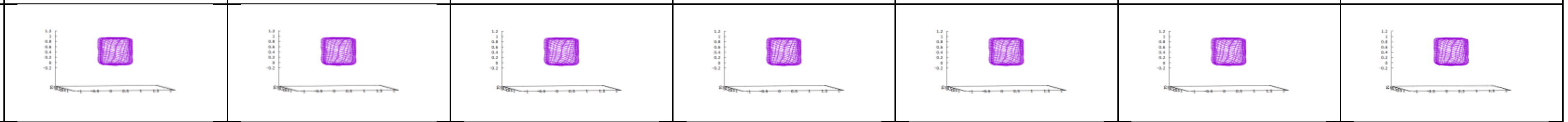
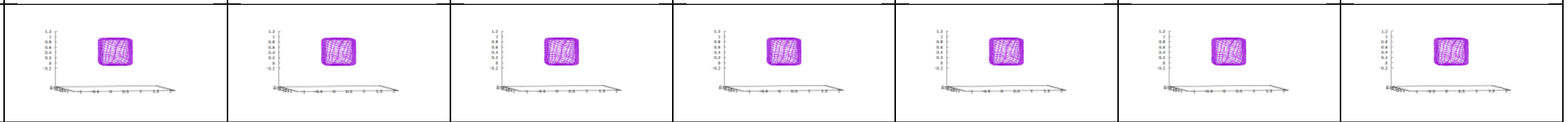
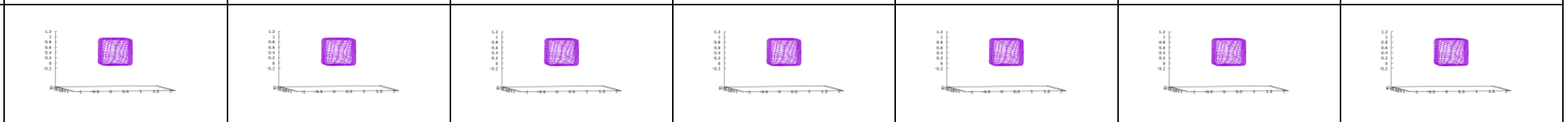
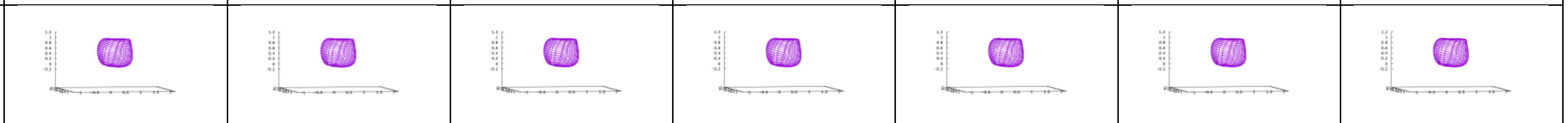
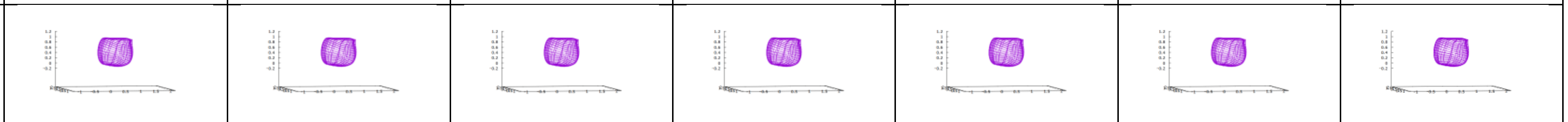
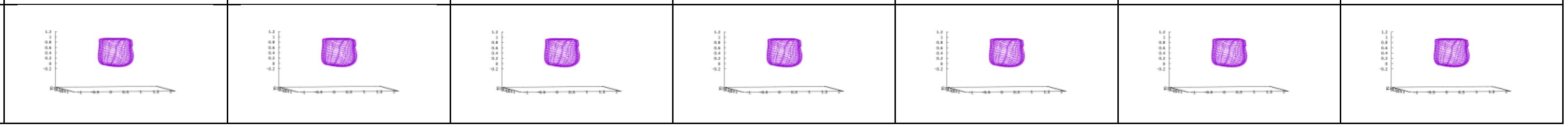
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	16 × 28								
Talus Bone	4 × 16								
	6 × 18								
	8 × 20								
	10 × 22								
	12 × 24								
	14 × 26								

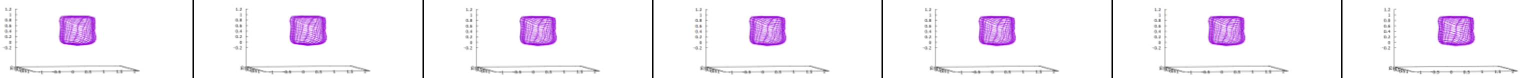
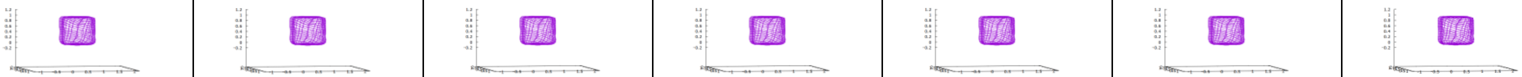
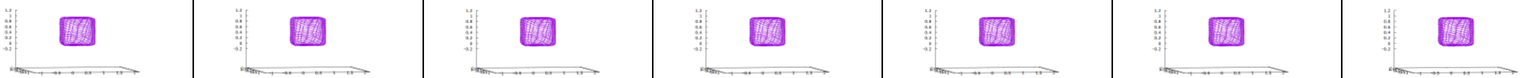
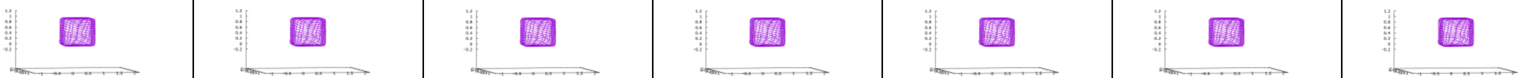
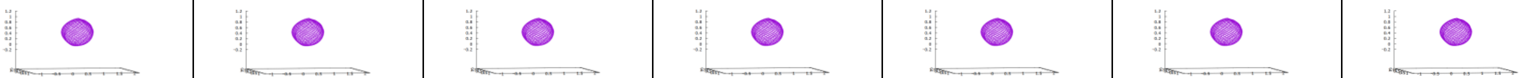
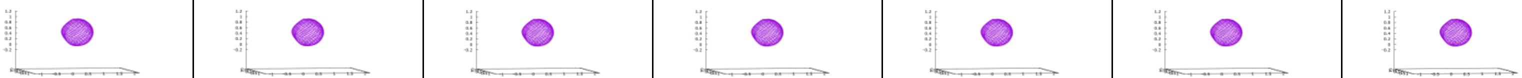
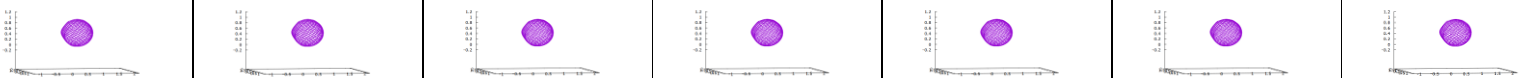
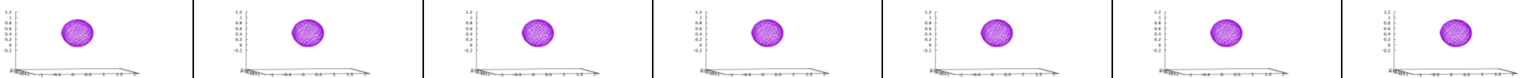
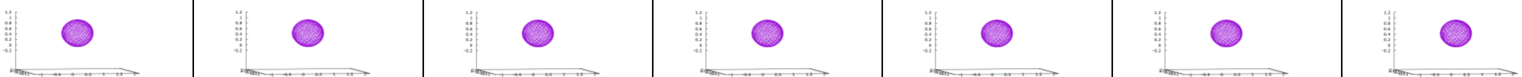
Data	CN	Uniform		Chord Length		Centripetal		Exponential	
		A	*B*	*A*	*B*	*A*	*B*	*A*	*B*
	16 × 28								

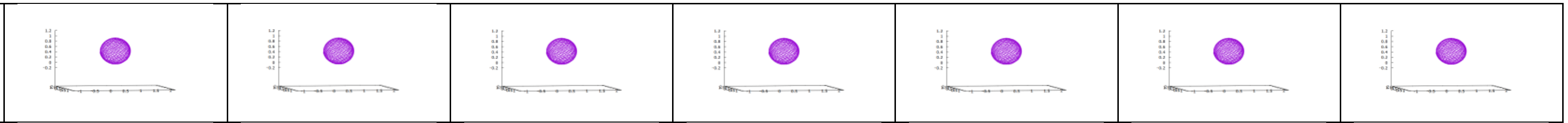
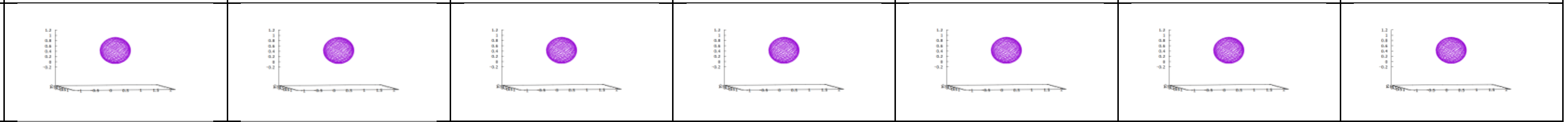
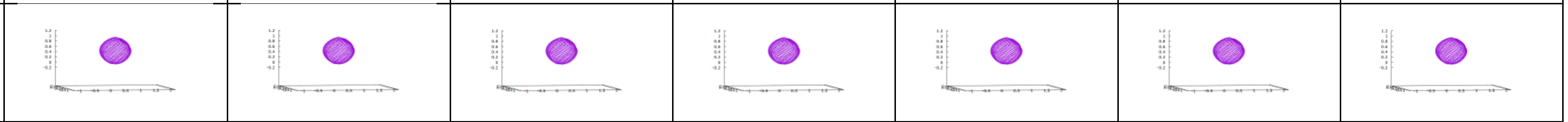
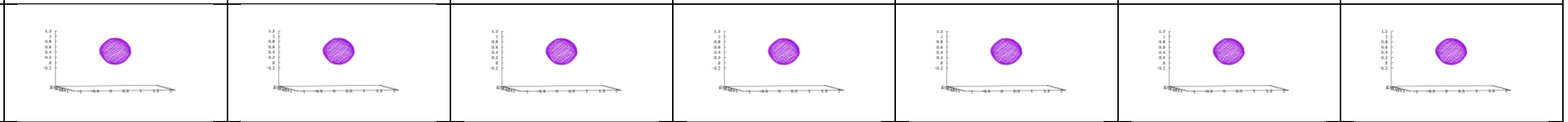
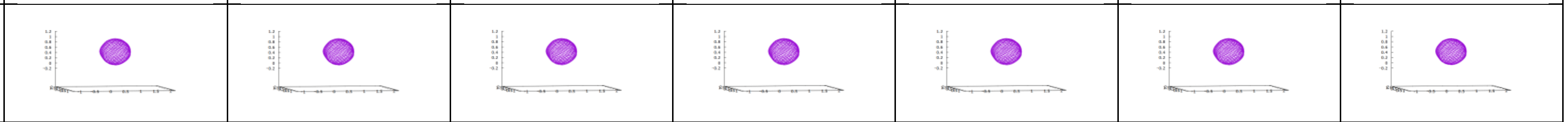
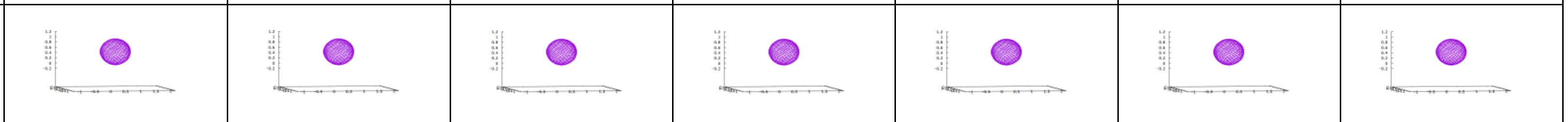
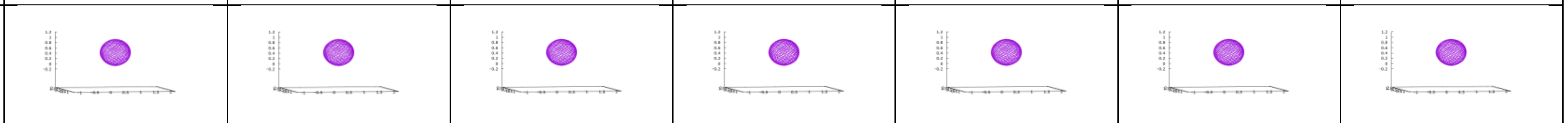
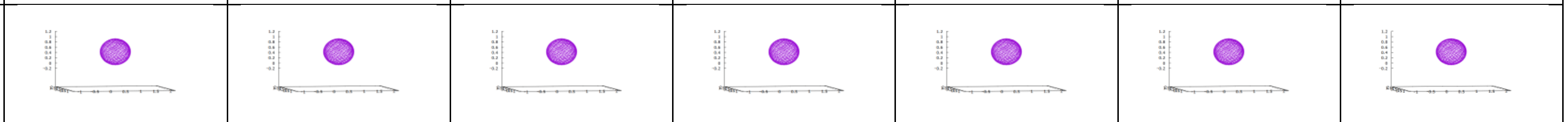
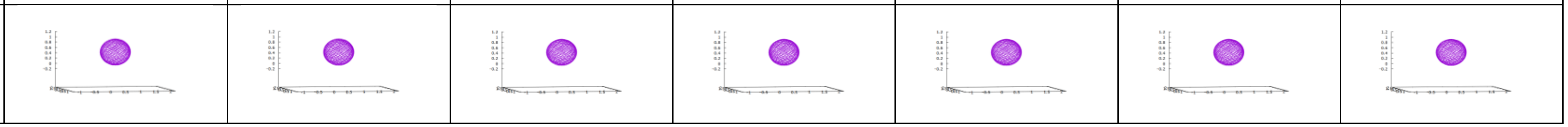
APPENDIX I: VISUALISATION OF OPTIMISED SURFACE DATA AND IMAGE RESULTS OF THE IMPROVED (*B*) SURFACE APPROXIMATION APPROACH FOR CN WITH THE SAME WIDTH AND LENGTH

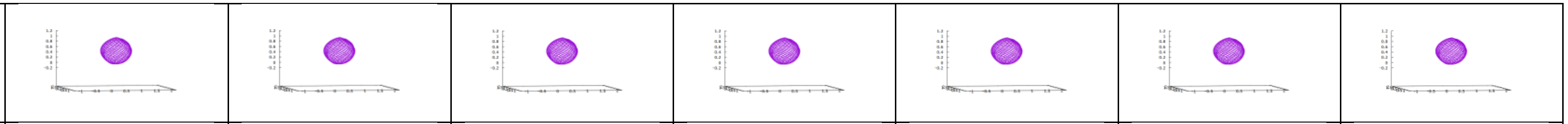
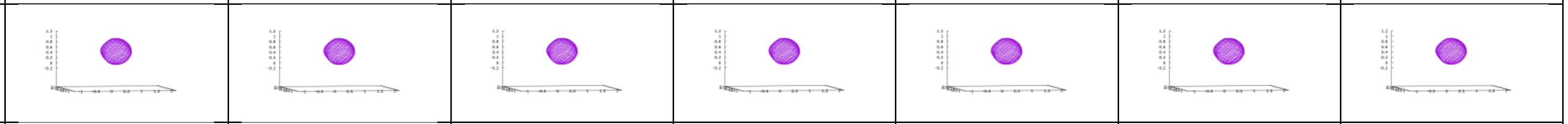
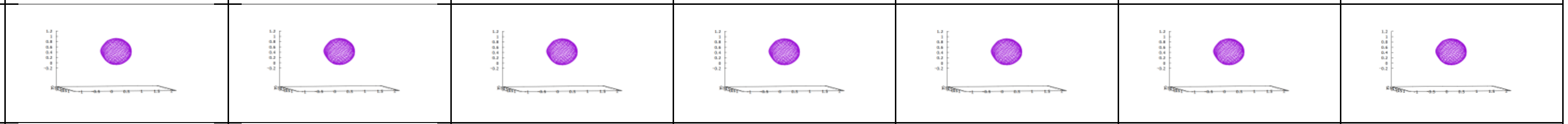
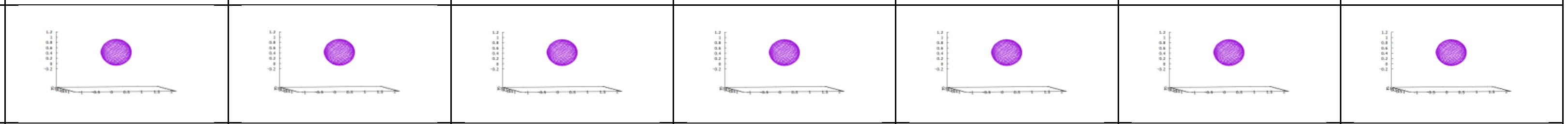
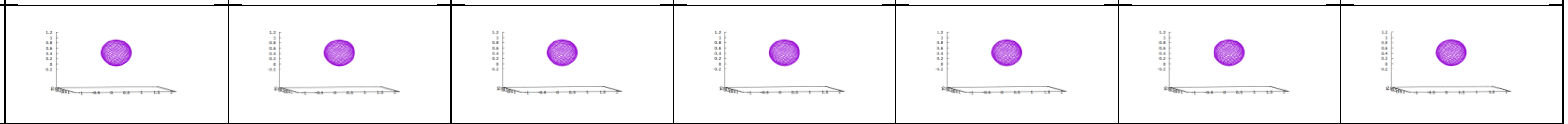
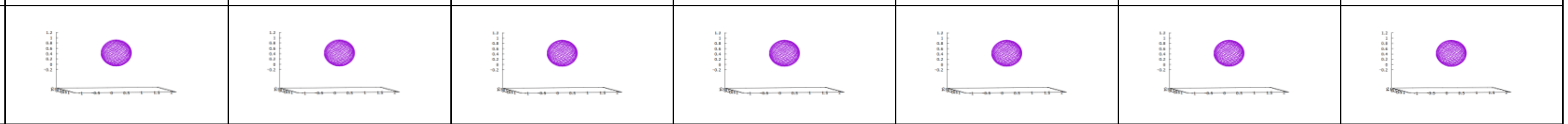
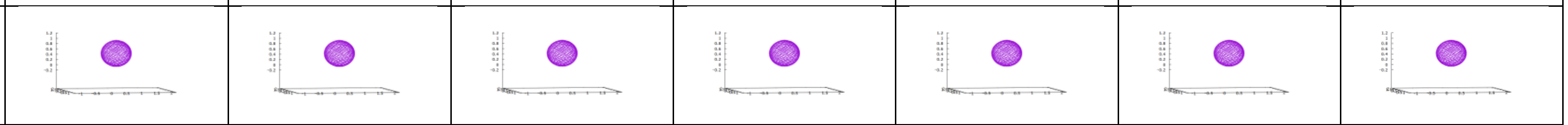
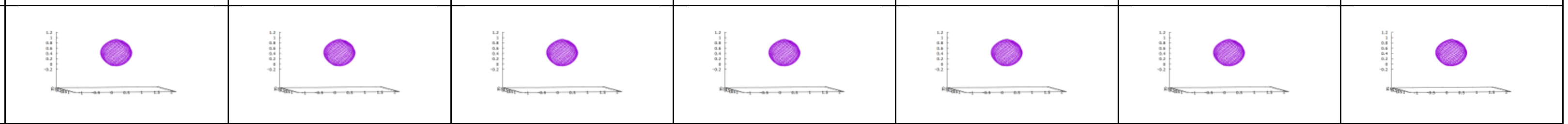
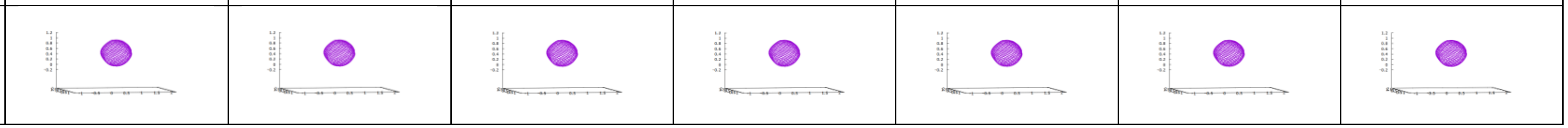
Data	Parameterisation Method	CN	GA		DE		PSO		*B*
			MIN	MAX	MIN	MAX	MIN	MAX	
Cube	Uniform	6 × 6							
		8 × 8							
		10 × 10							
		12 × 12							
		14 × 14							
		16 × 16							

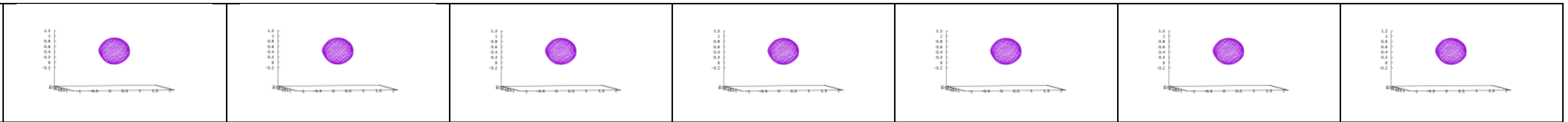
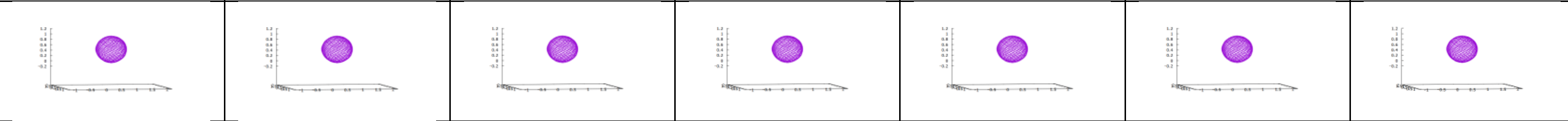
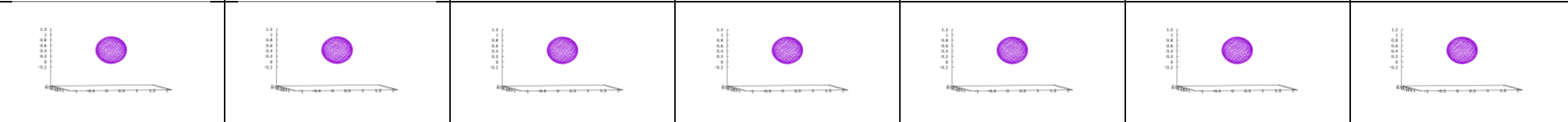
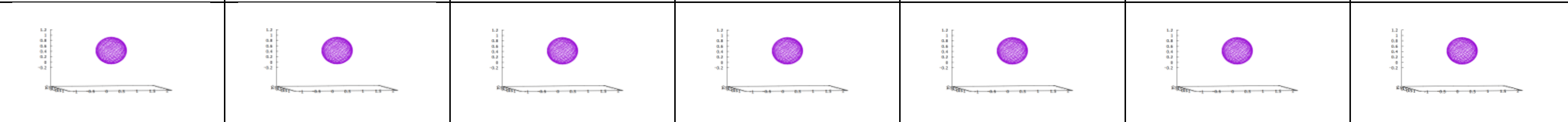
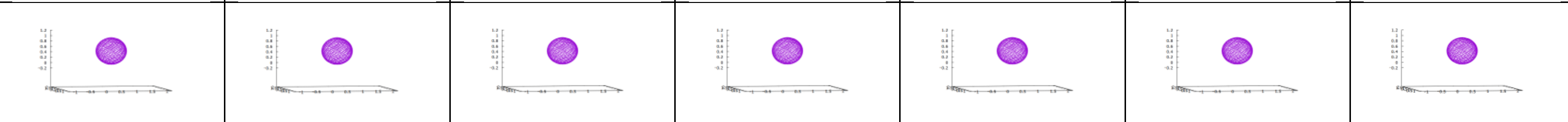
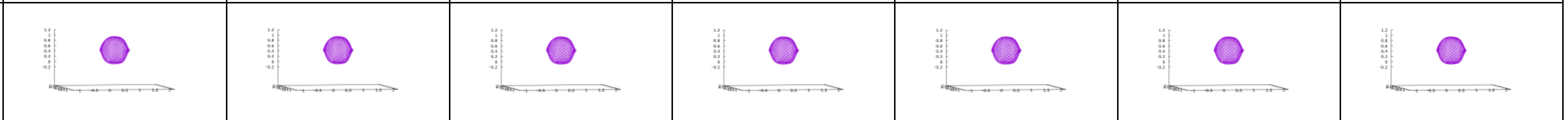
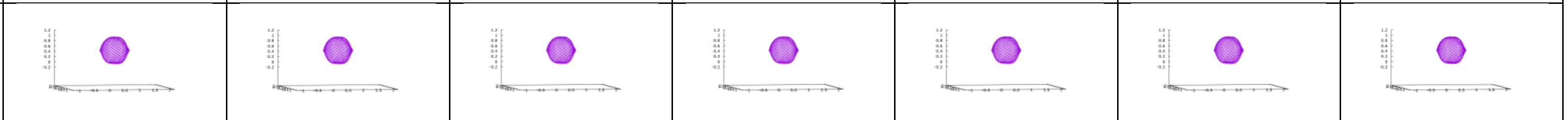
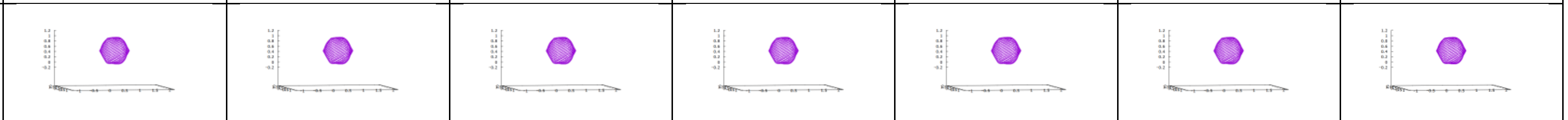
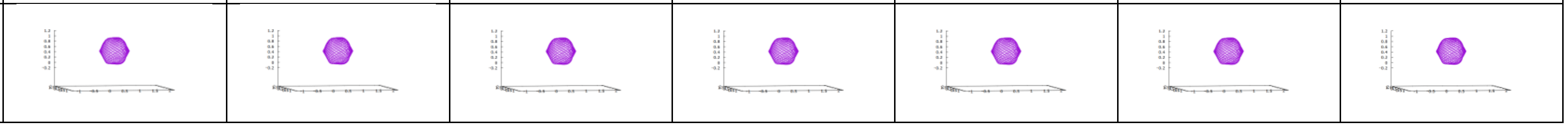
		18 × 18	
	Chord Length	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
	Centripetal	6 × 6	

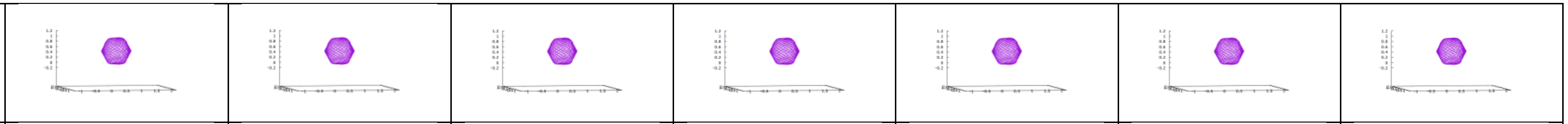
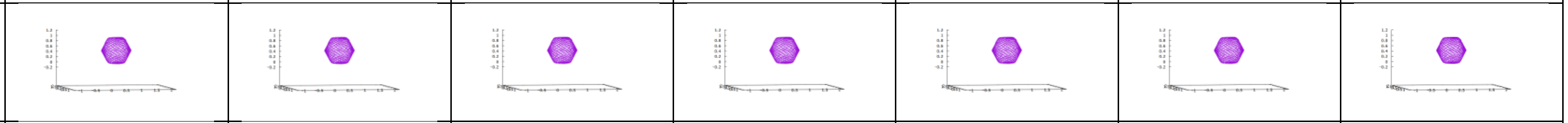
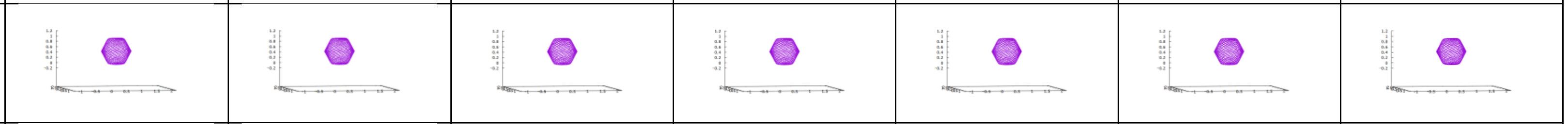
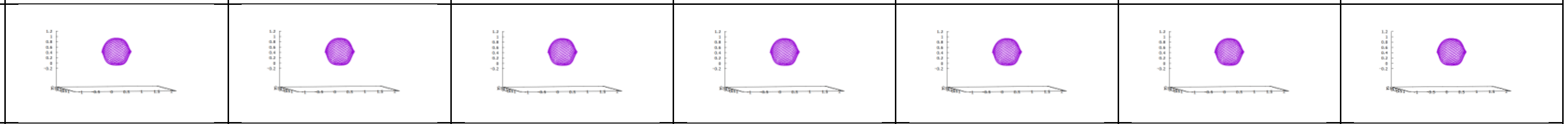
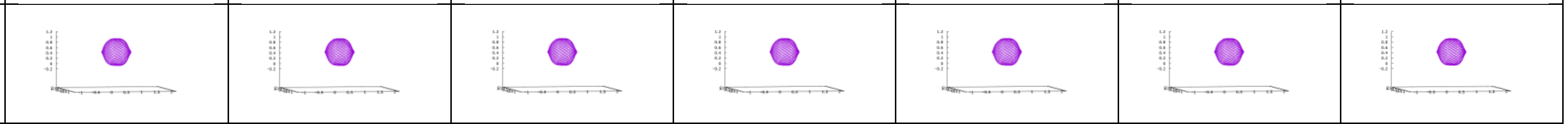
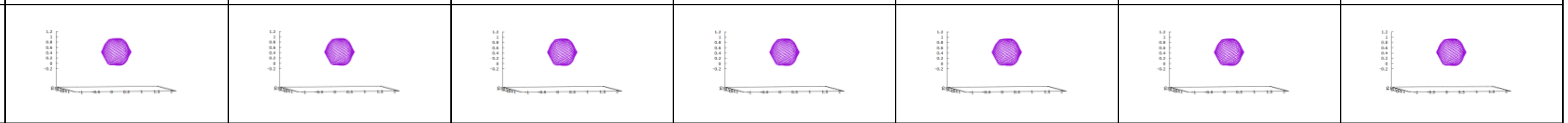
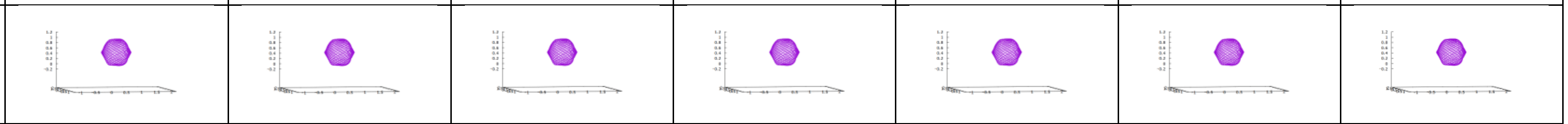
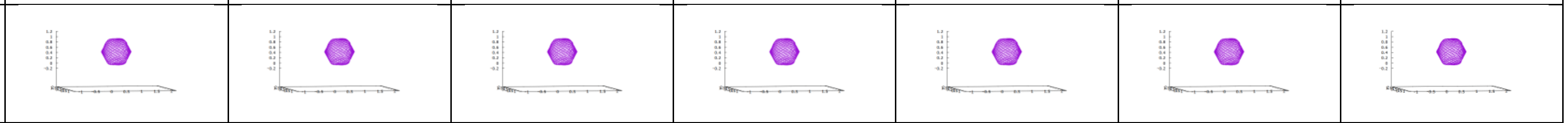
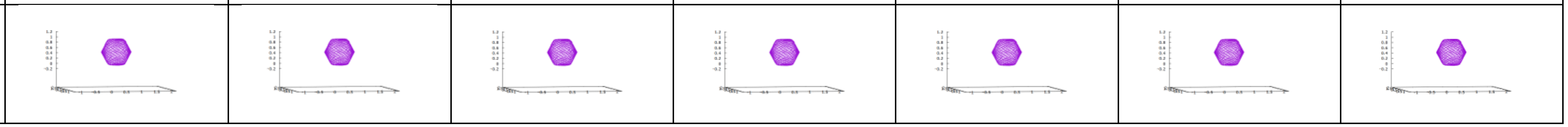
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
	Exponential	6 × 6	
		8 × 8	
		10 × 10	

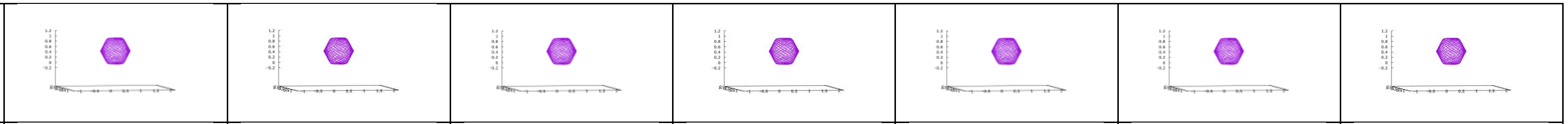
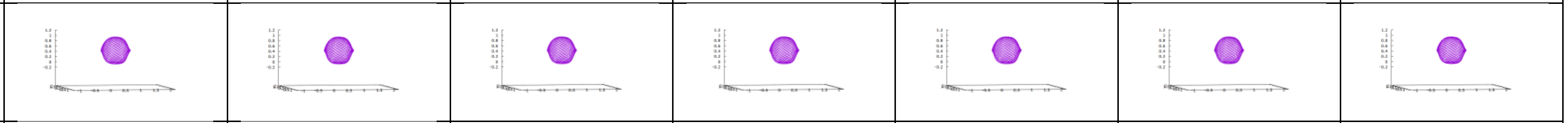
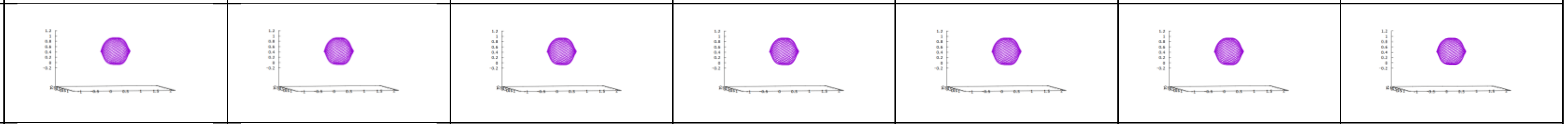
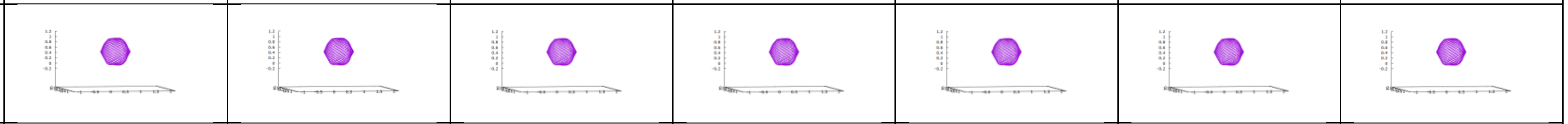
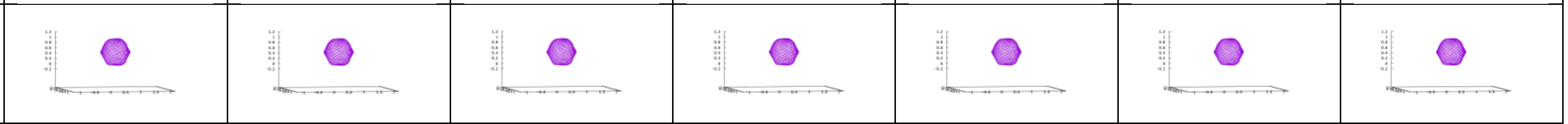
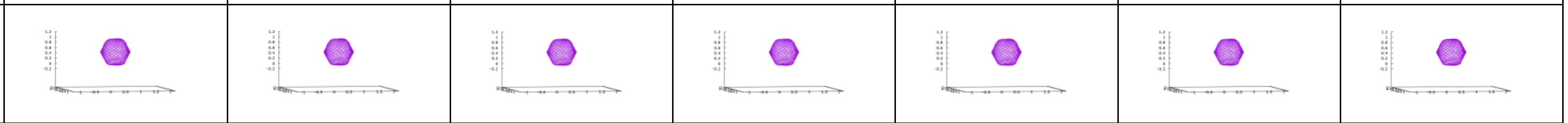
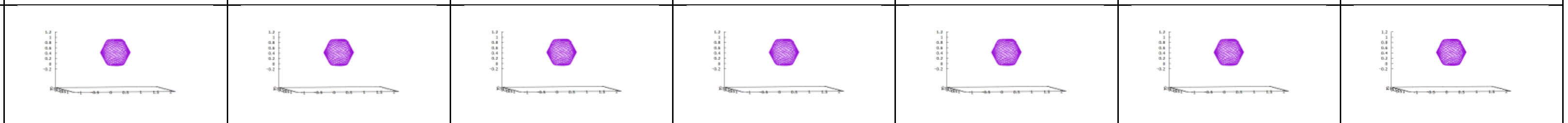
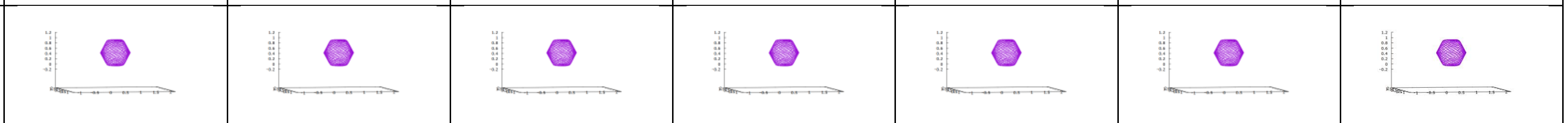
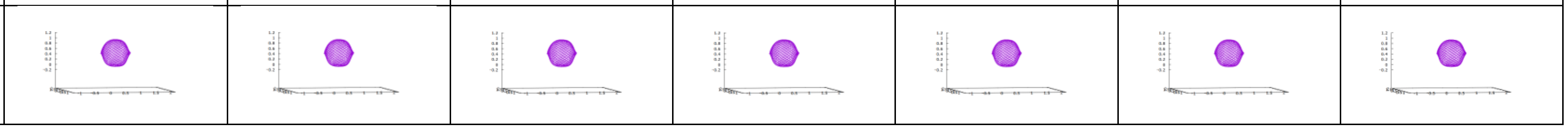
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
Sphere	Uniform	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	

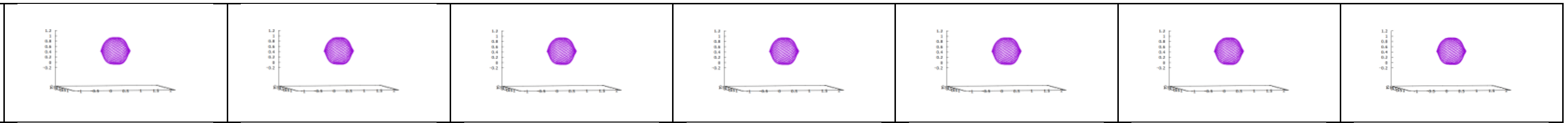
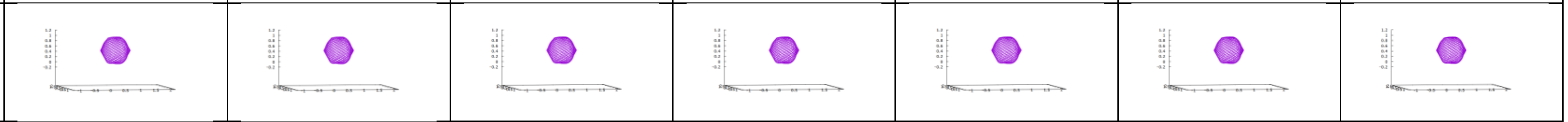
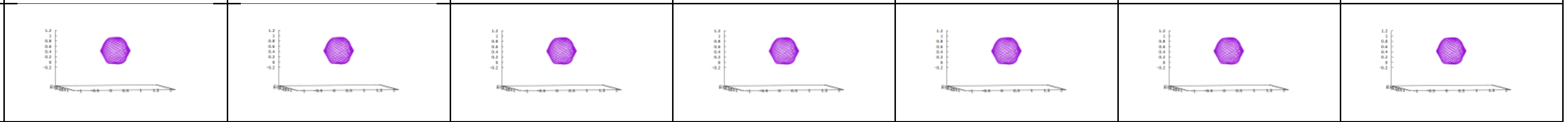
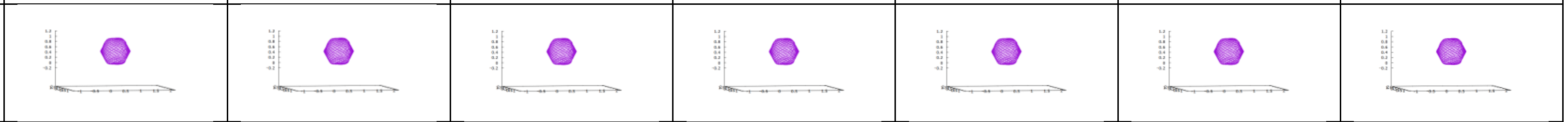
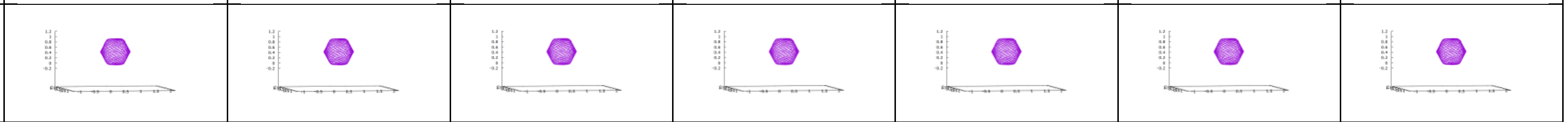
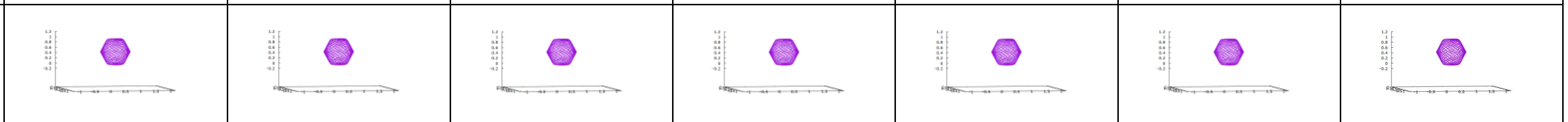
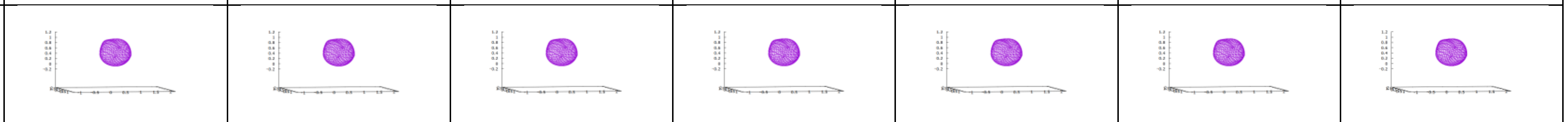
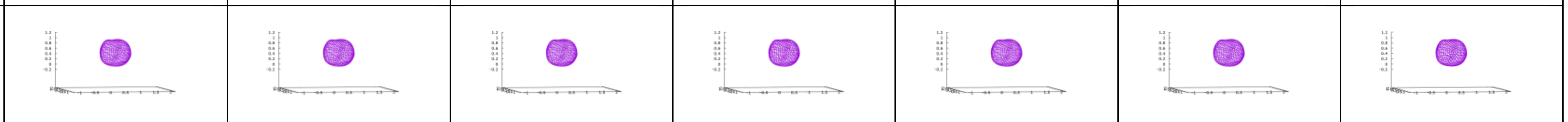
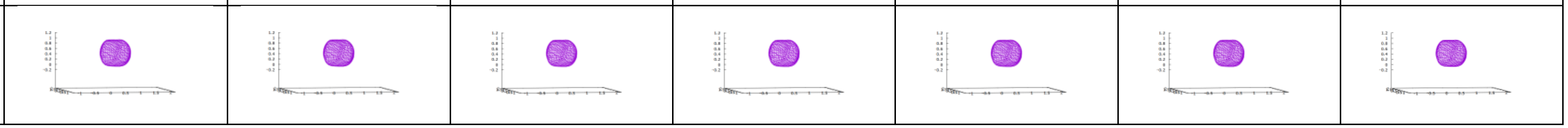
		16 × 16	
		18 × 18	
	Chord Length	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	

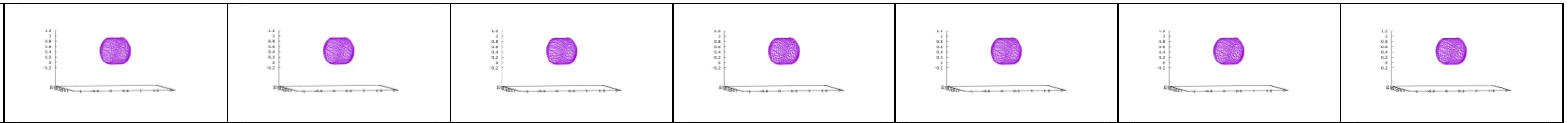
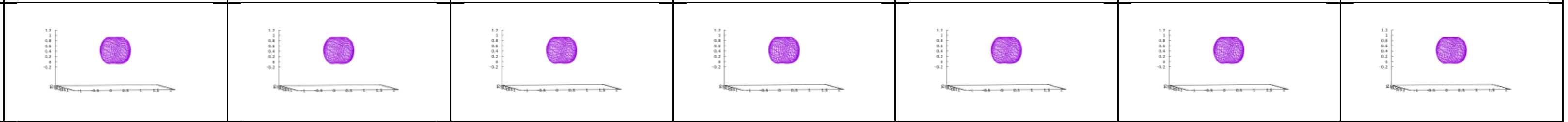
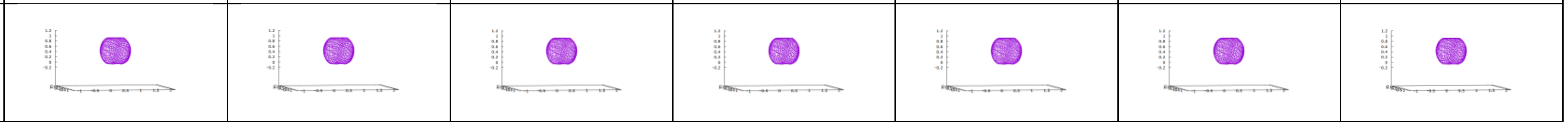
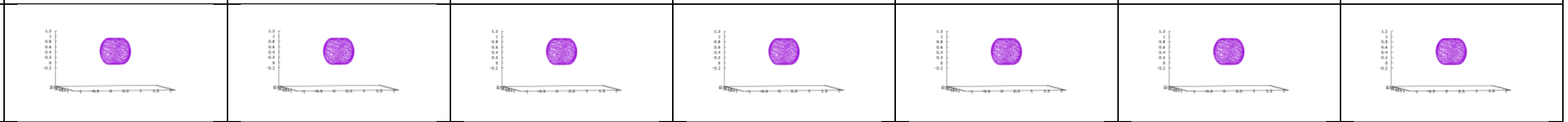
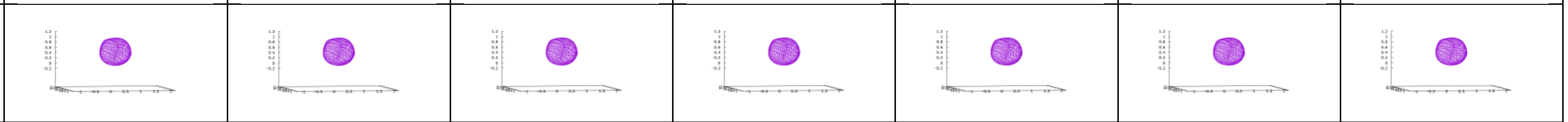
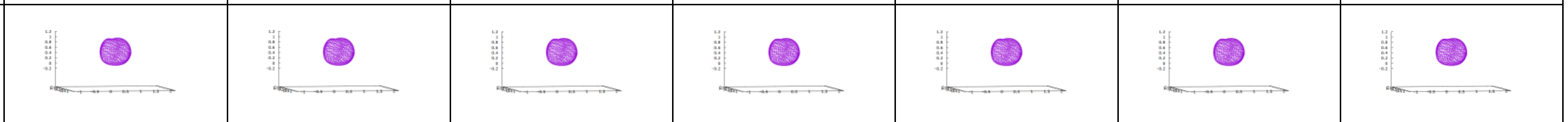
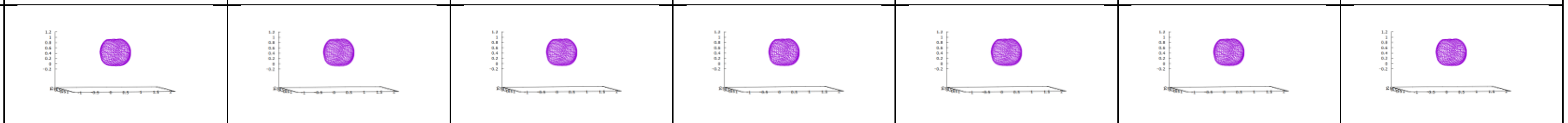
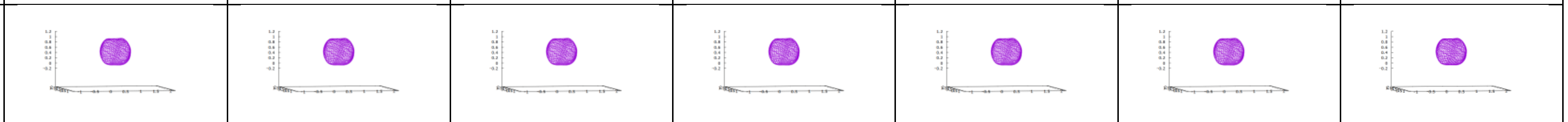
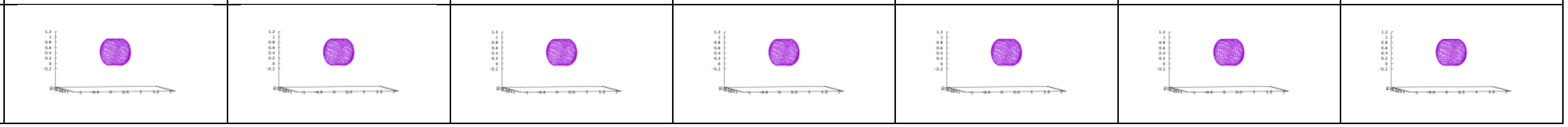
	Centripetal	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
	Exponential	6 × 6	
		8 × 8	

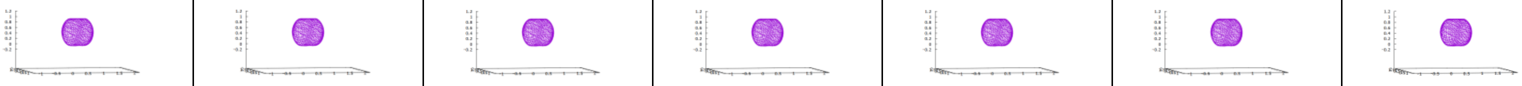
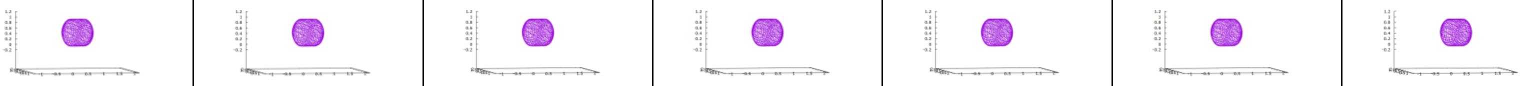
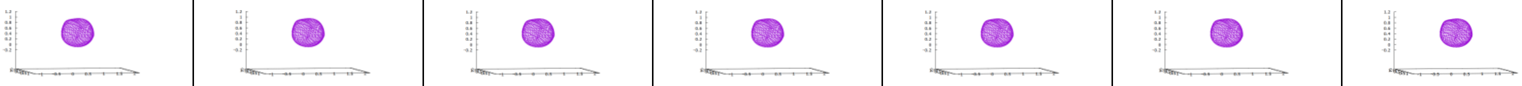
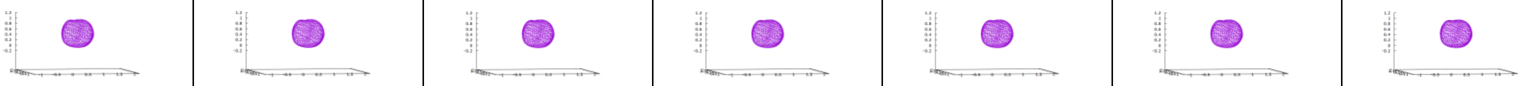
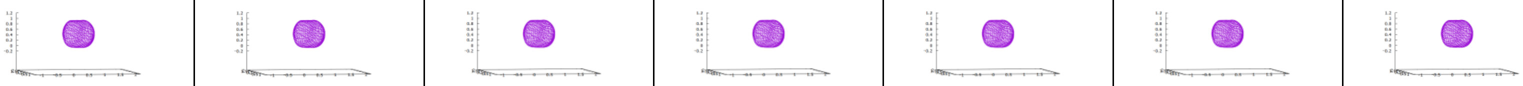
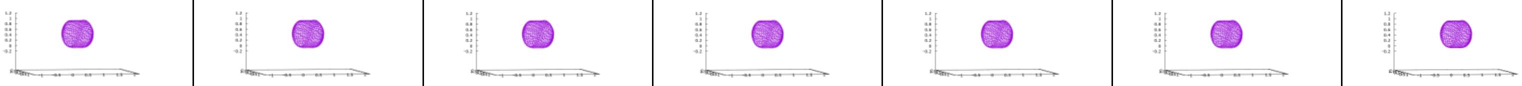
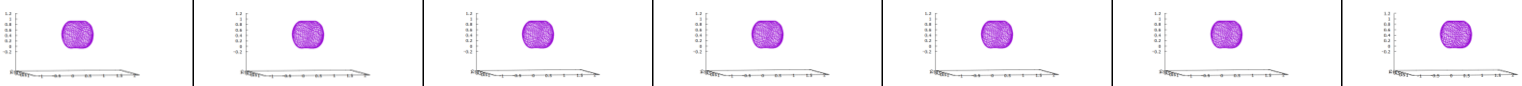
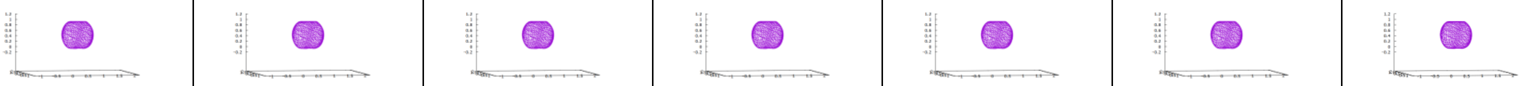
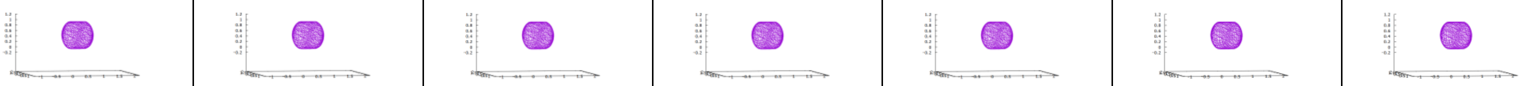
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
Spindle	Uniform	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	

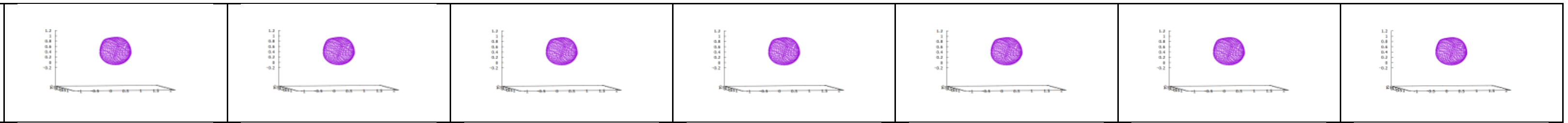
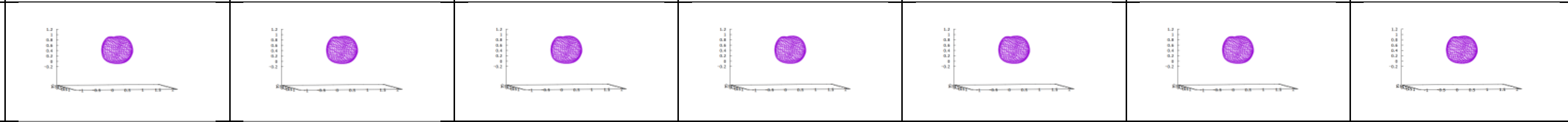
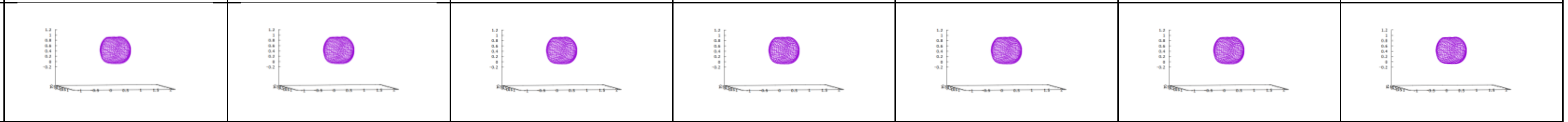
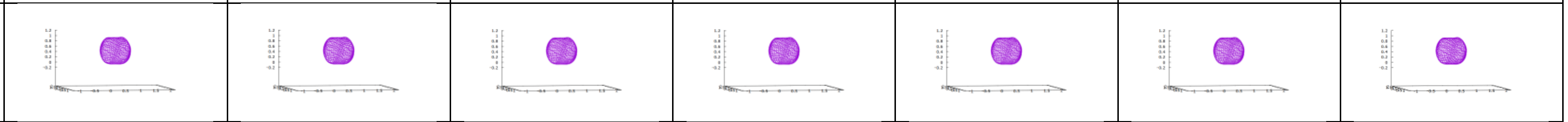
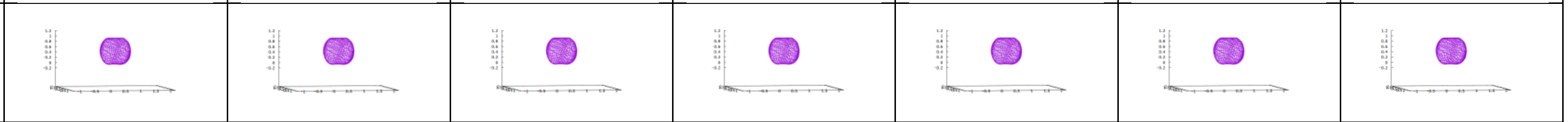
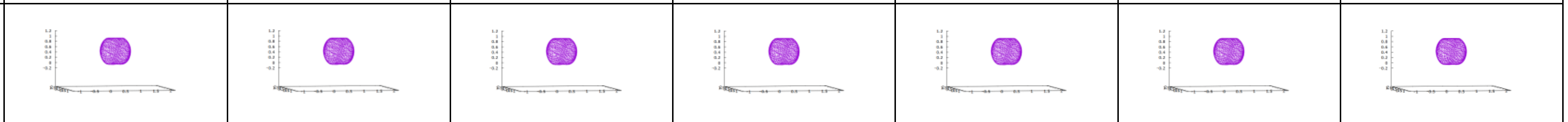
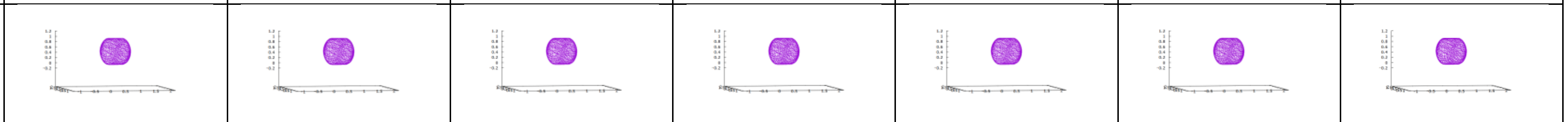
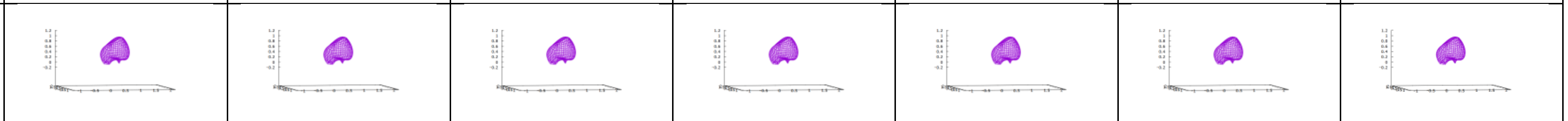
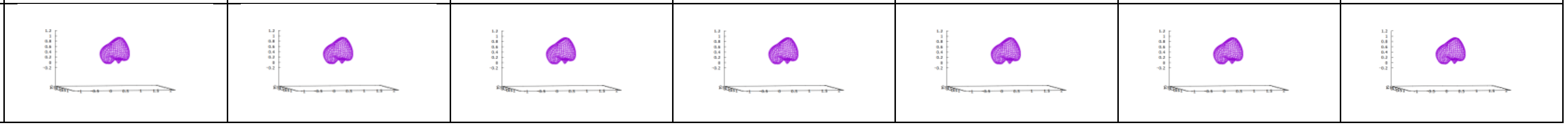
		14 × 14	
		16 × 16	
		18 × 18	
	Chord Length	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	

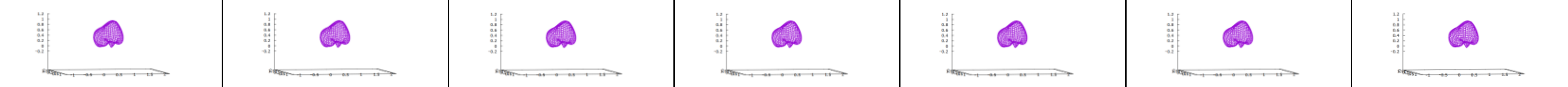
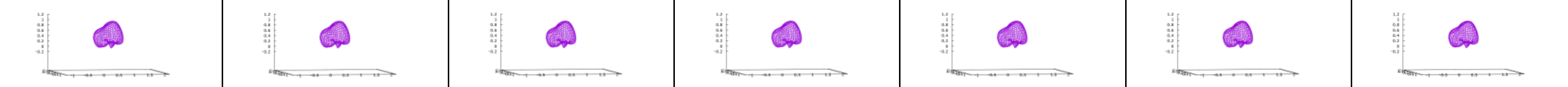
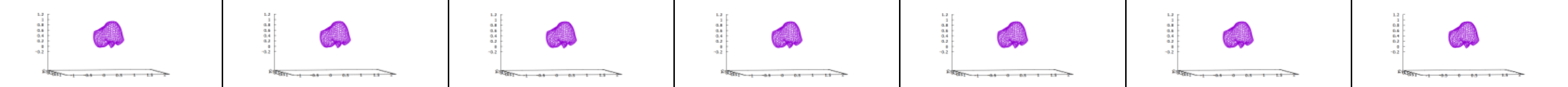
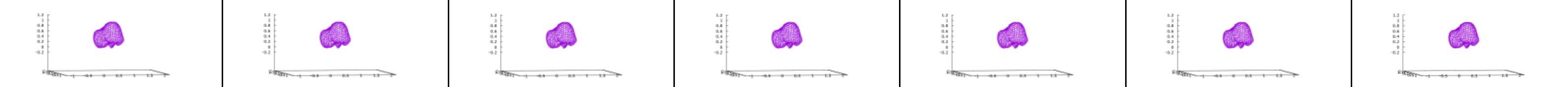
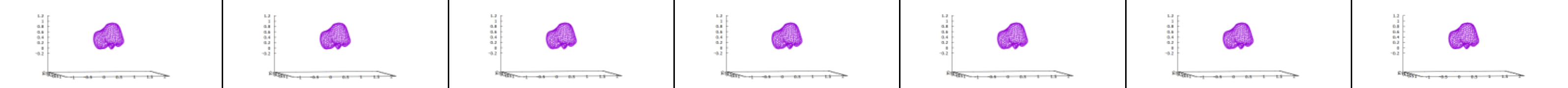
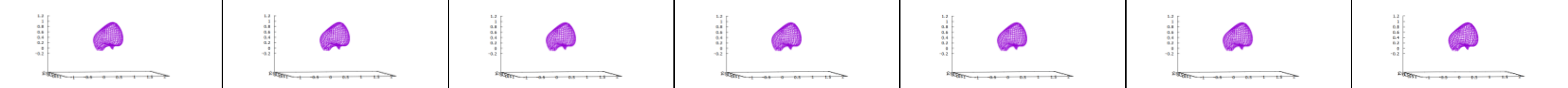
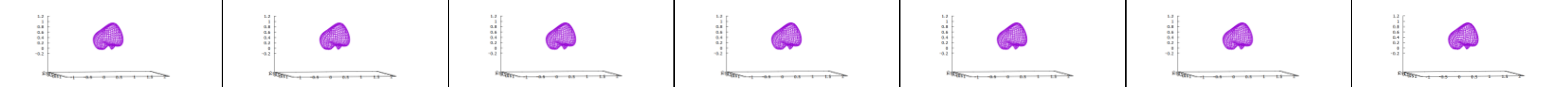
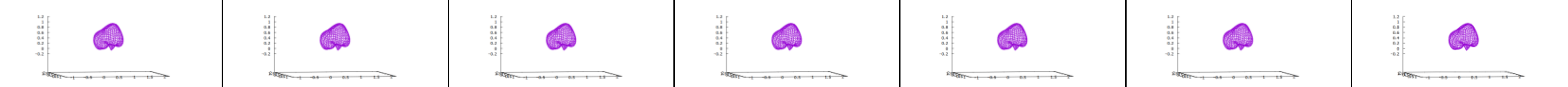
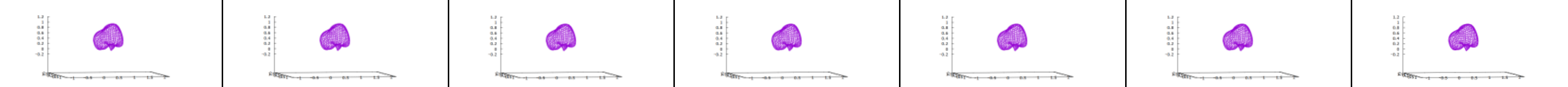
		18 × 18							
	Centripetal	6 × 6							
		8 × 8							
		10 × 10							
		12 × 12							
		14 × 14							
		16 × 16							
		18 × 18							
	Exponenti al	6 × 6							

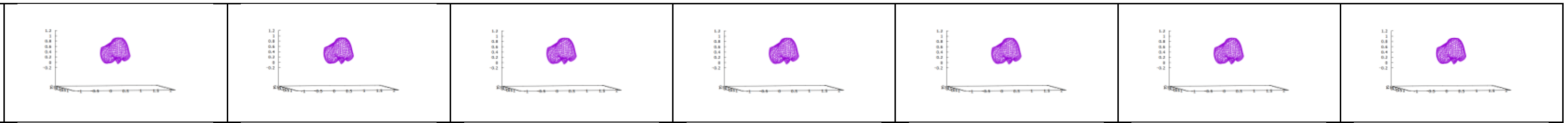
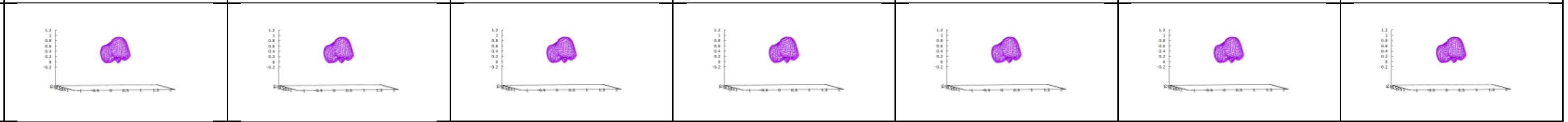
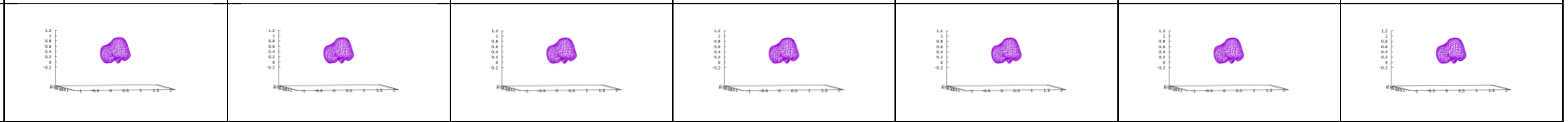
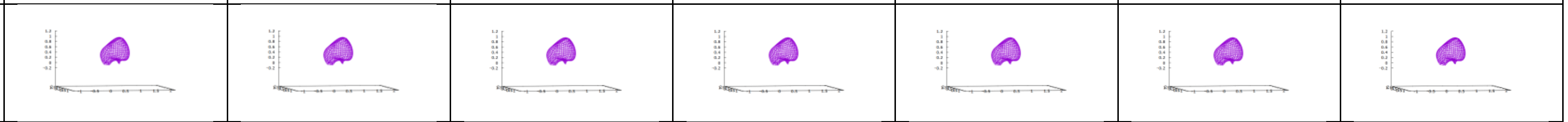
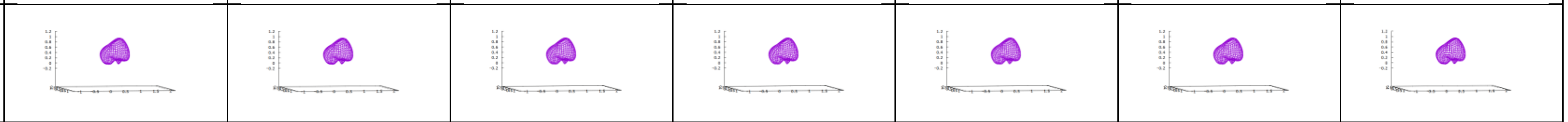
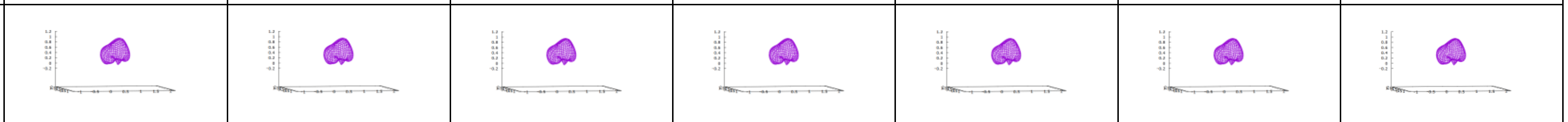
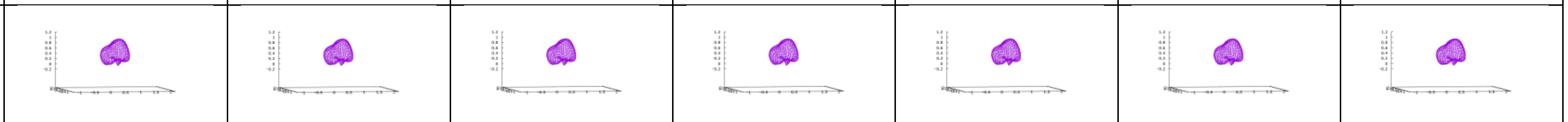
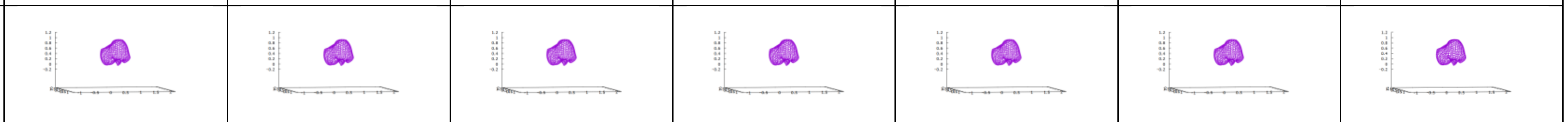
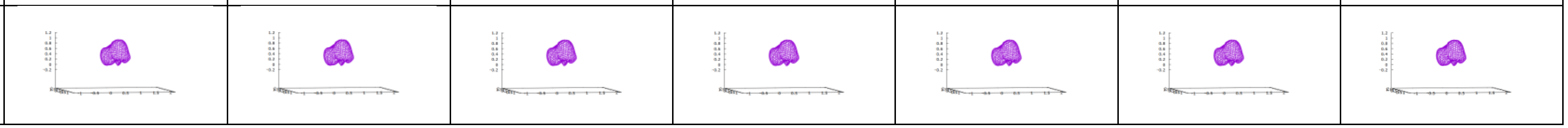
		8 × 8							
		10 × 10							
		12 × 12							
		14 × 14							
		16 × 16							
		18 × 18							
Oiltank	Uniform	6 × 6							
		8 × 8							
		10 × 10							

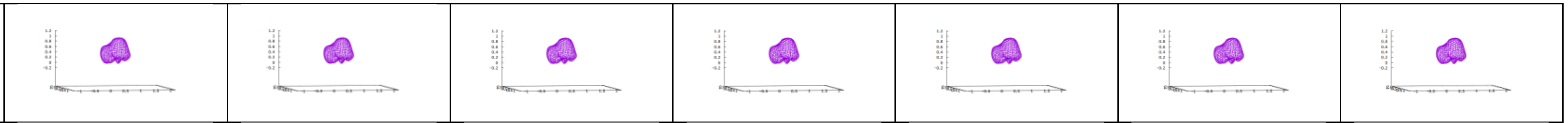
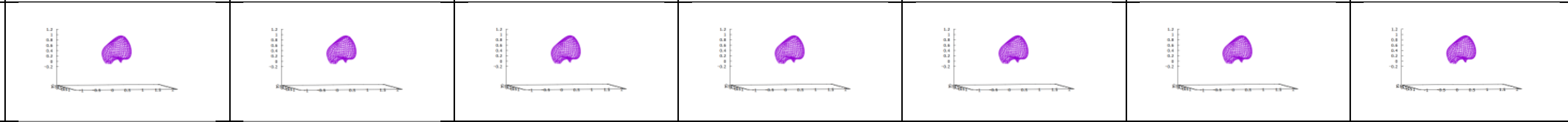
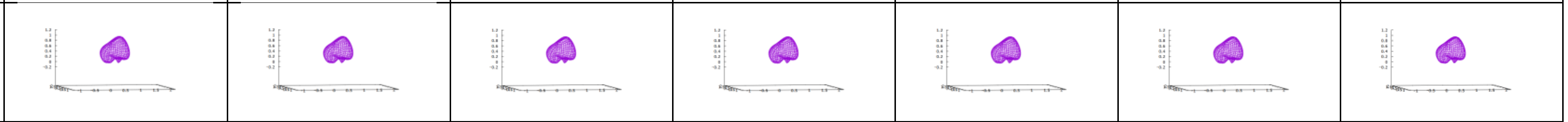
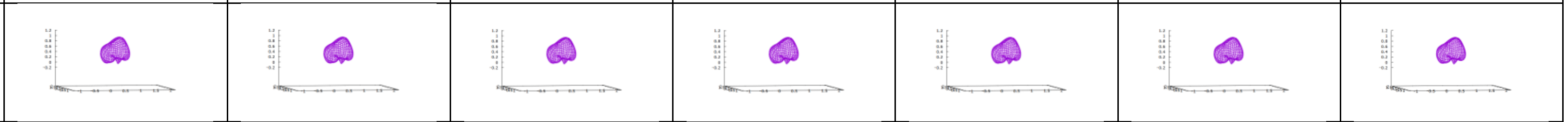
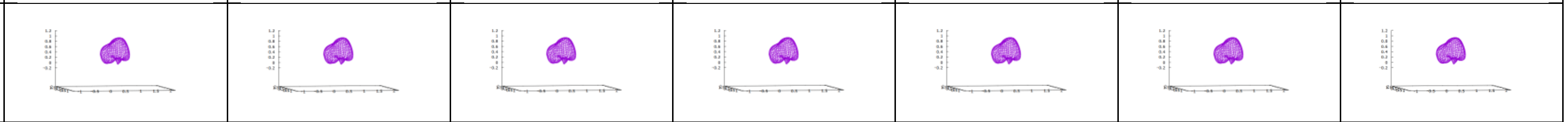
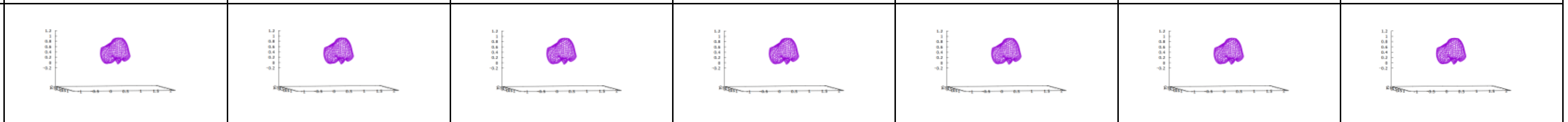
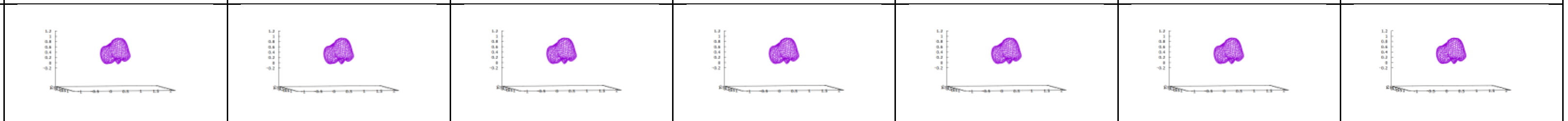
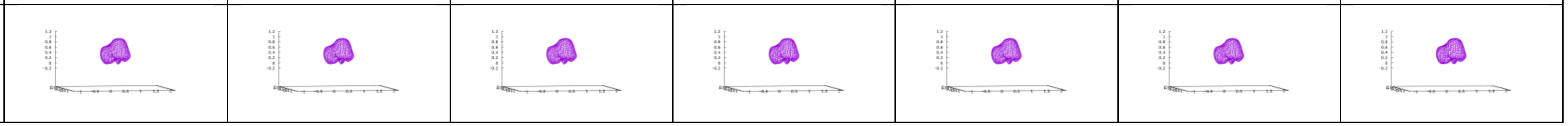
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
	Chord Length	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	

		16 × 16	
		18 × 18	
	Centripetal	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	

	Exponential	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
Talus Bone	Uniform	6 × 6	
		8 × 8	

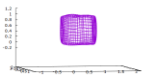
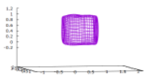
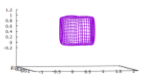
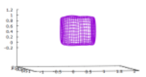
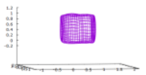
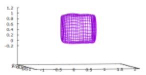
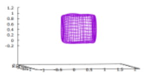
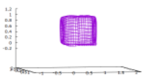
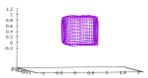
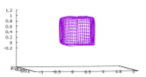
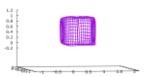
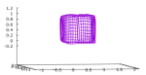
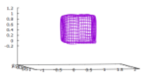
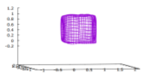
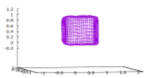
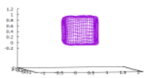
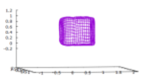
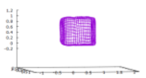
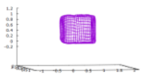
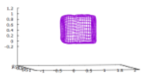
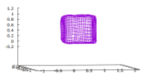
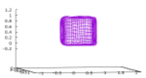
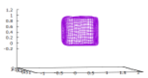
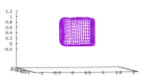
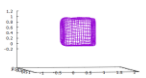
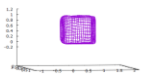
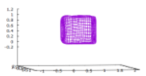
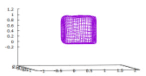
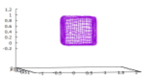
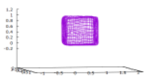
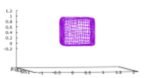
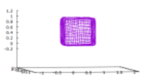
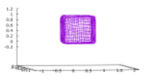
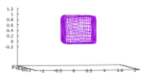
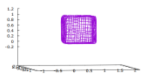
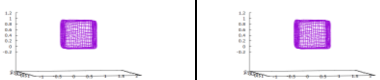
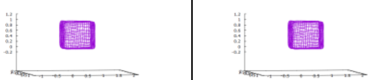
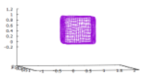
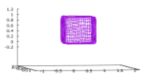
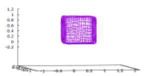
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	
	Chord Length	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	

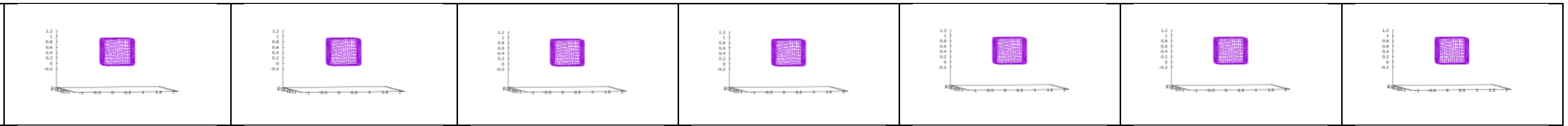
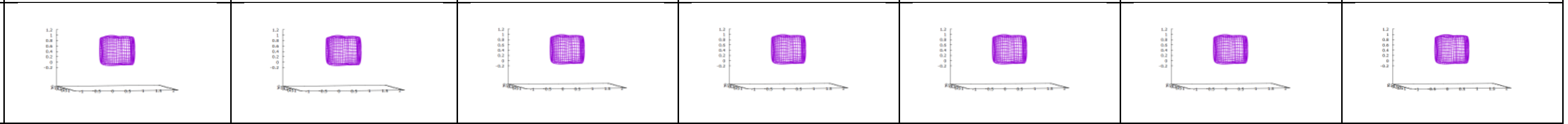
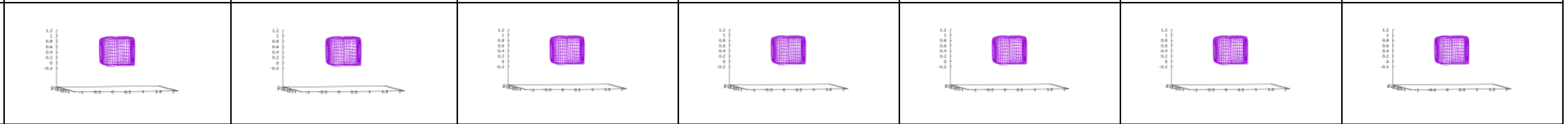
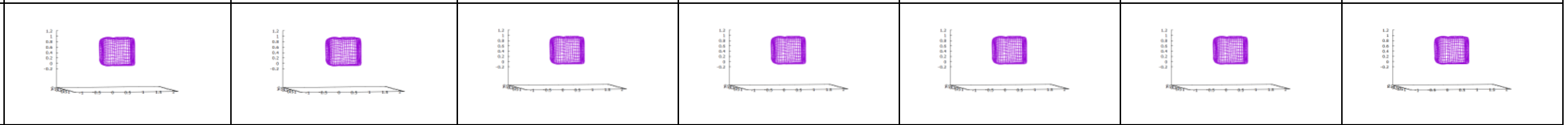
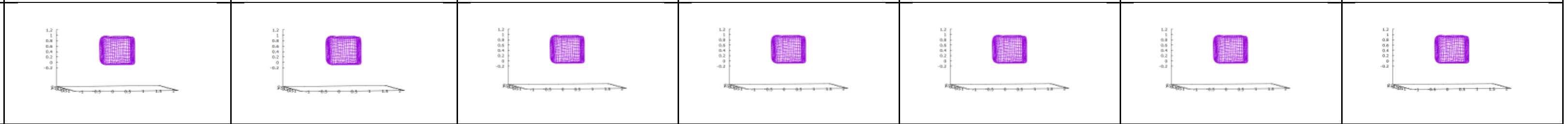
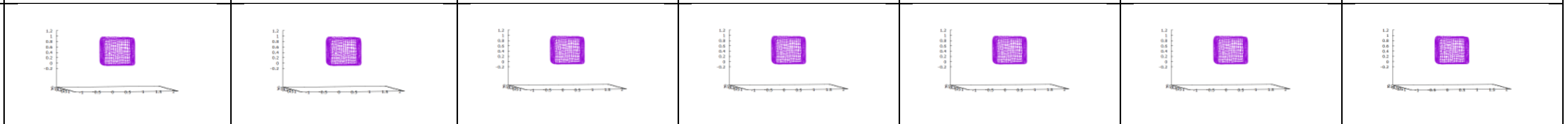
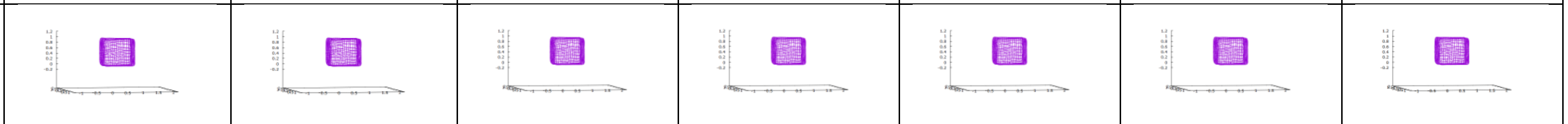
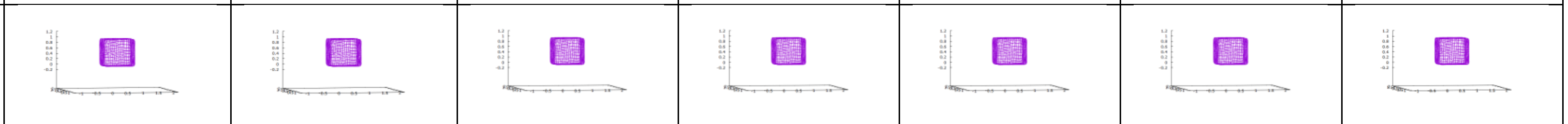
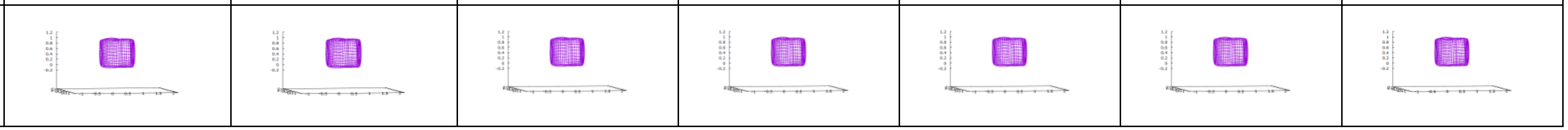
		14 × 14	
		16 × 16	
		18 × 18	
	Centripetal	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	

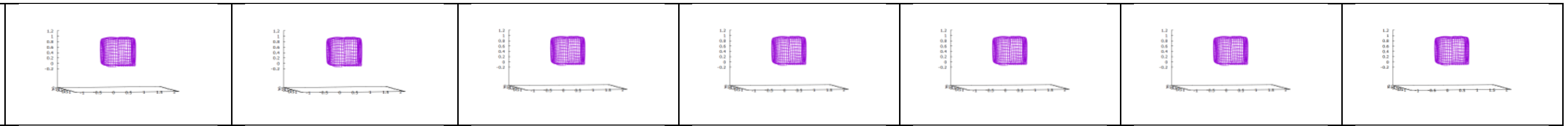
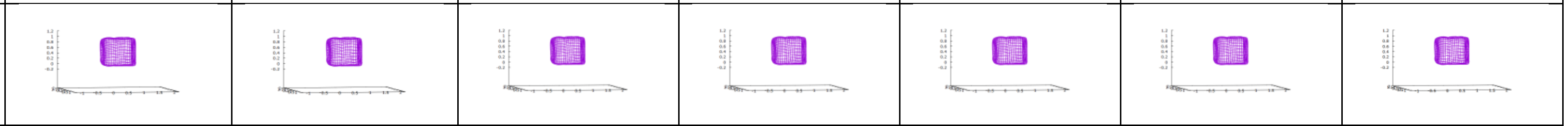
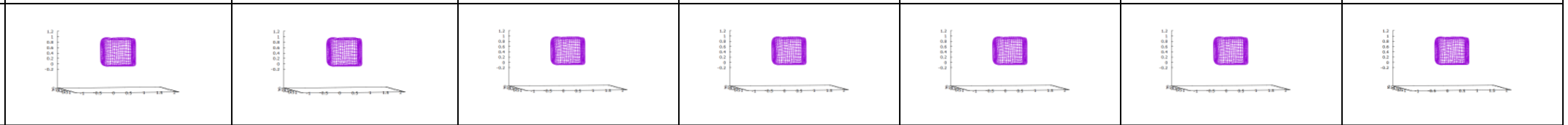
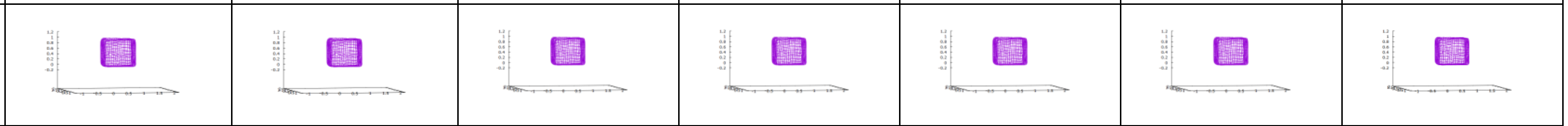
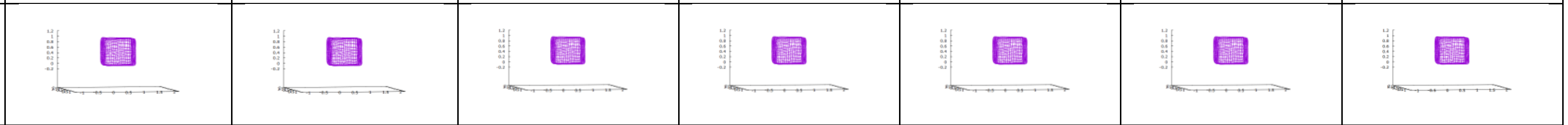
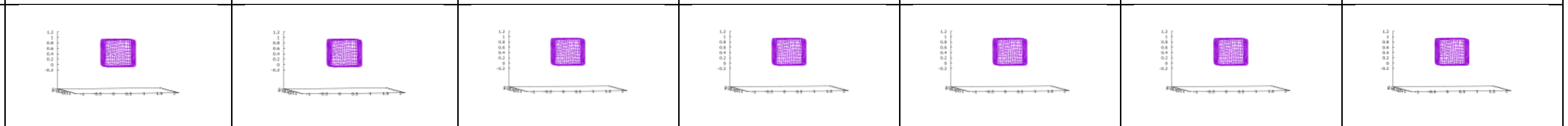
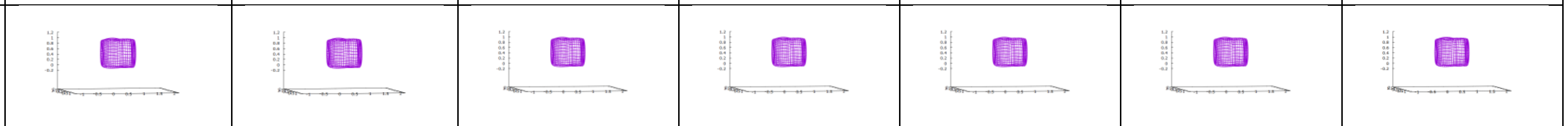
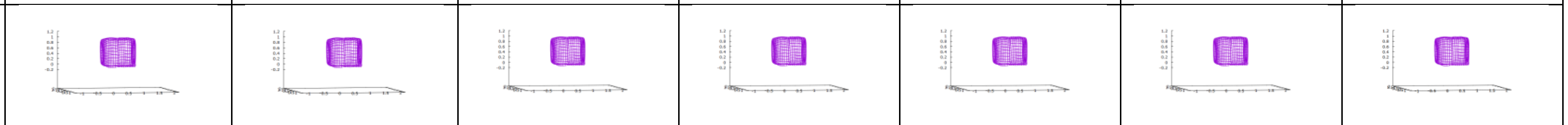
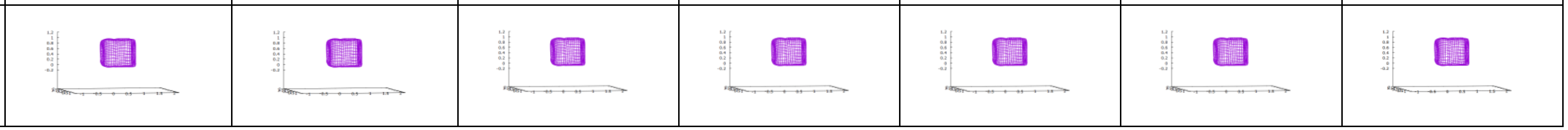
		18 × 18	
	Exponential	6 × 6	
		8 × 8	
		10 × 10	
		12 × 12	
		14 × 14	
		16 × 16	
		18 × 18	

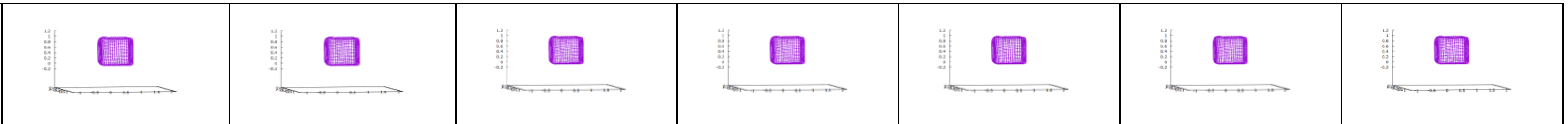
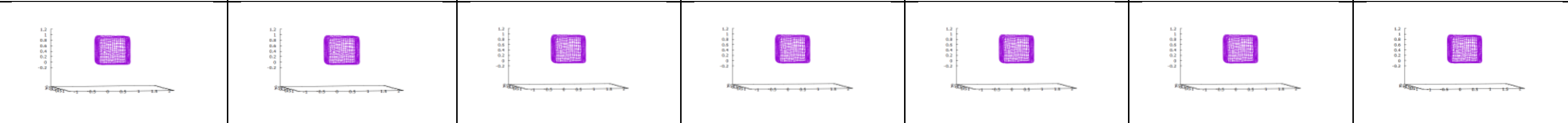
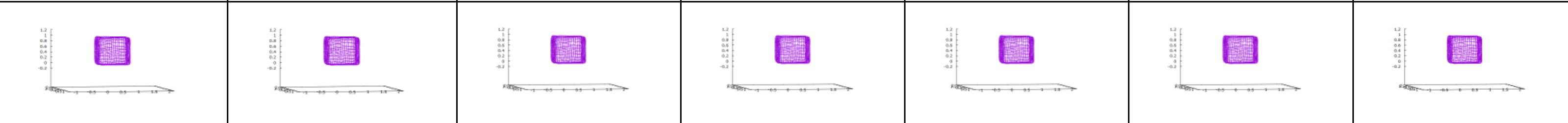
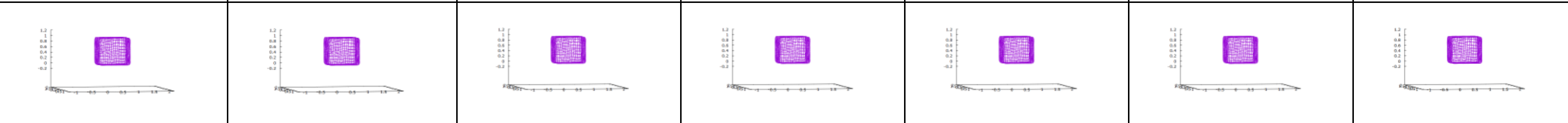
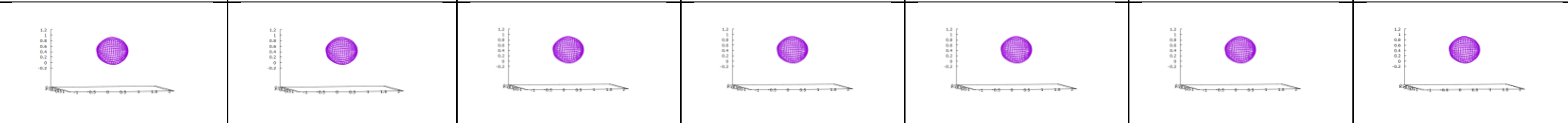
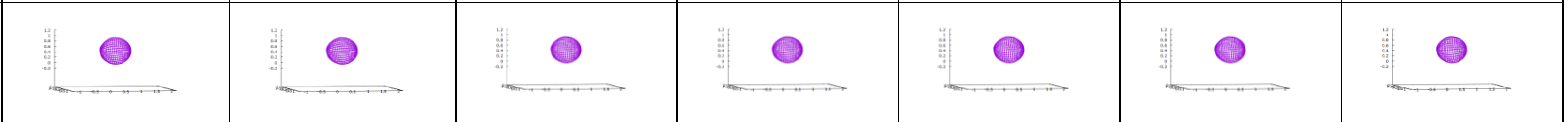
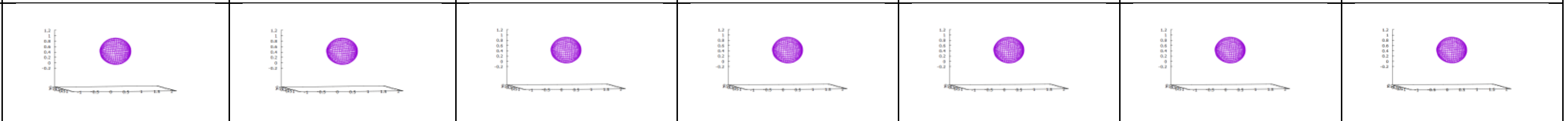
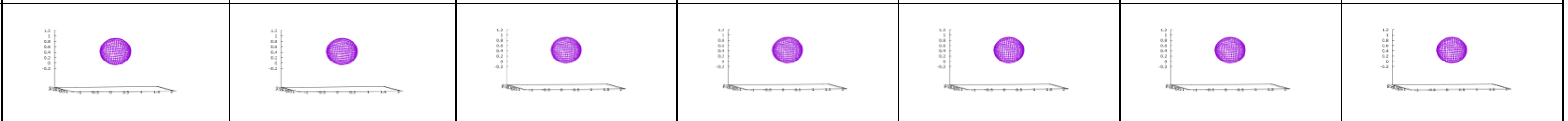
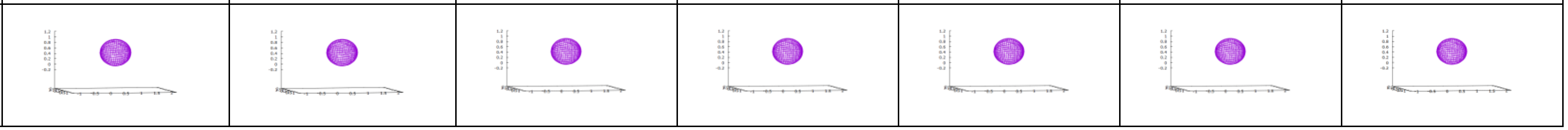
4.

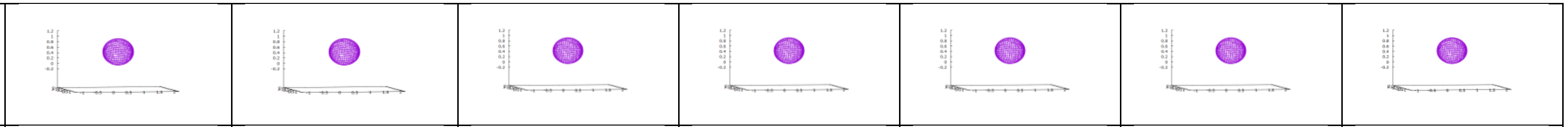
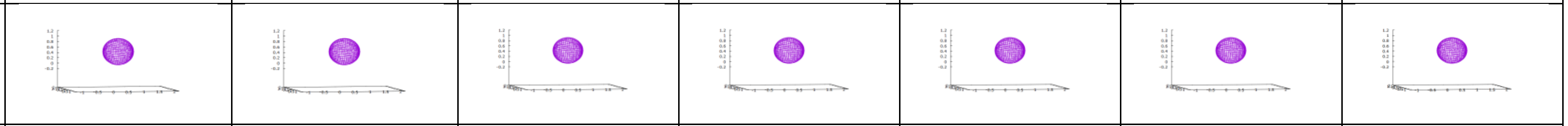
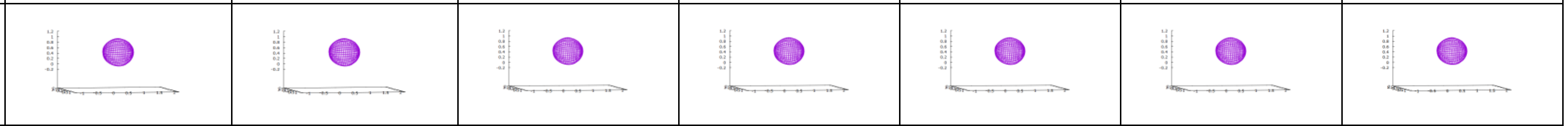
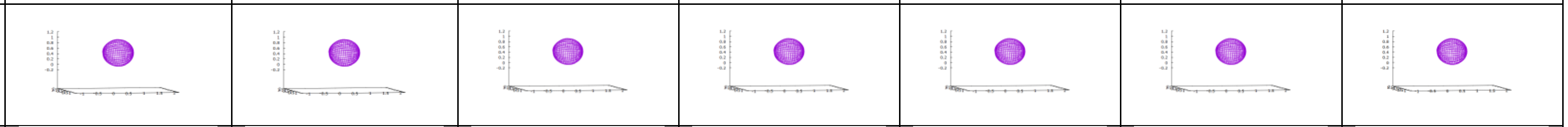
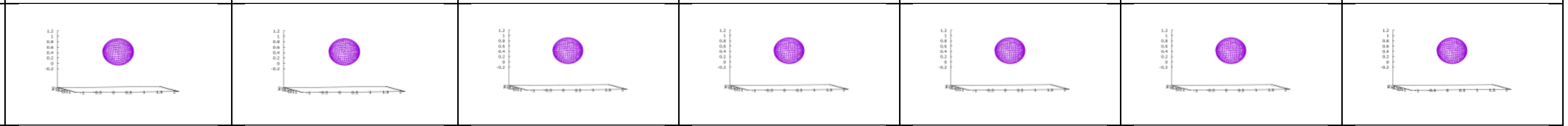
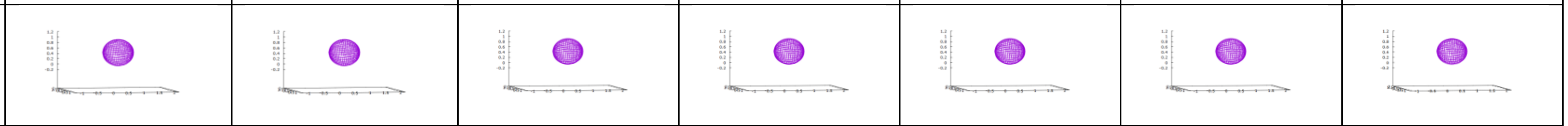
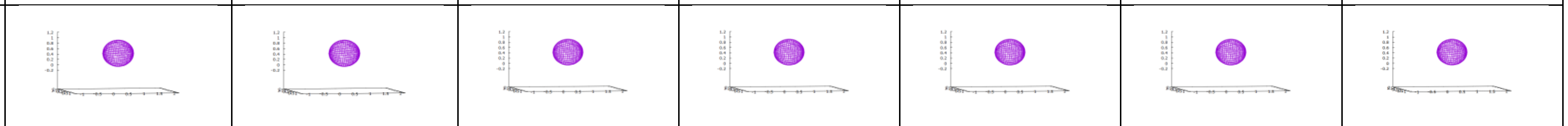
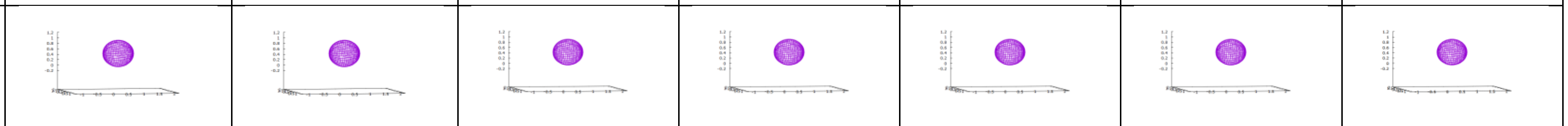
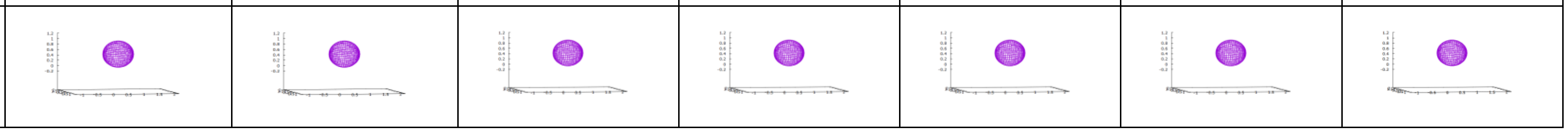
APPENDIX J: VISUALISATION OF OPTIMISED SURFACE DATA AND IMAGE RESULTS OF THE IMPROVED (*B*) SURFACE APPROXIMATION APPROACH FOR CN WITH DIFFERENT WIDTH AND LENGTH

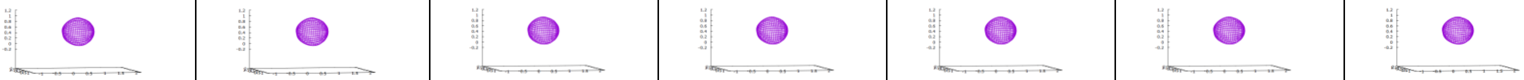
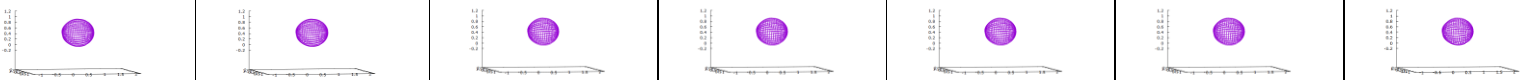
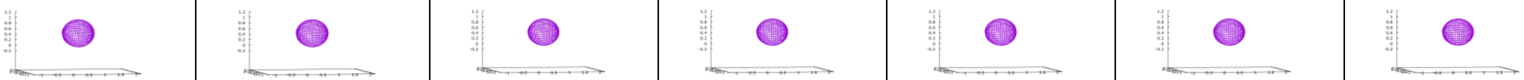
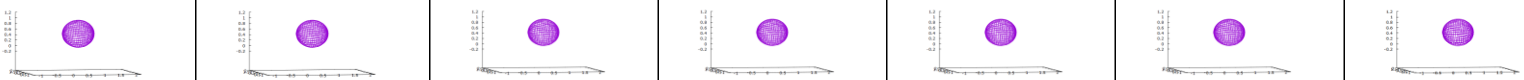
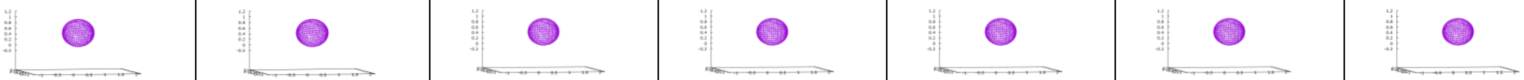
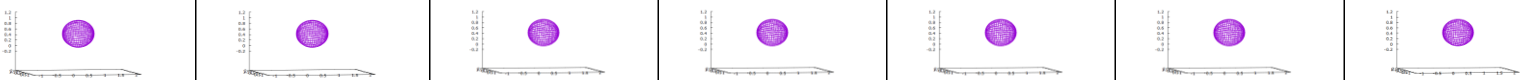
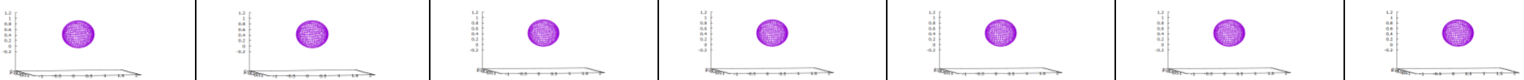
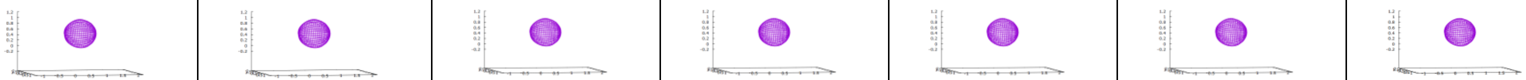
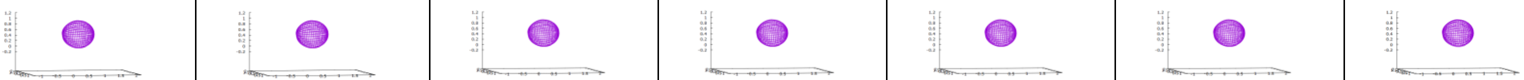
Data	Parameterisation Method	CN	GA		DE		PSO		*B*
			MIN	MAX	MIN	MAX	MIN	MAX	
Cube	Uniform	4 × 16							
		6 × 18							
		8 × 20							
		10 × 22							
		12 × 24							
		14 × 26							

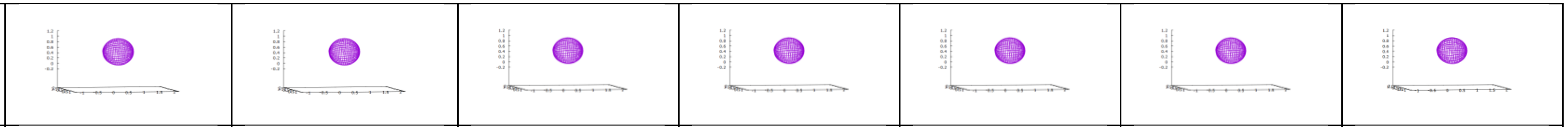
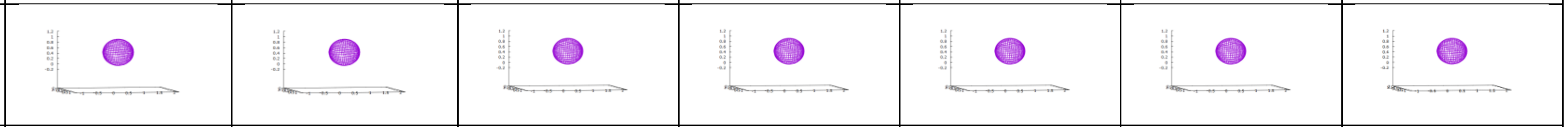
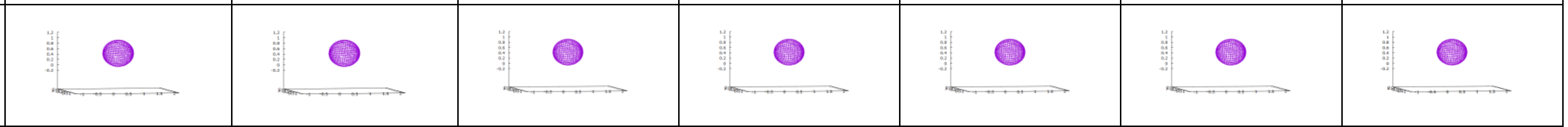
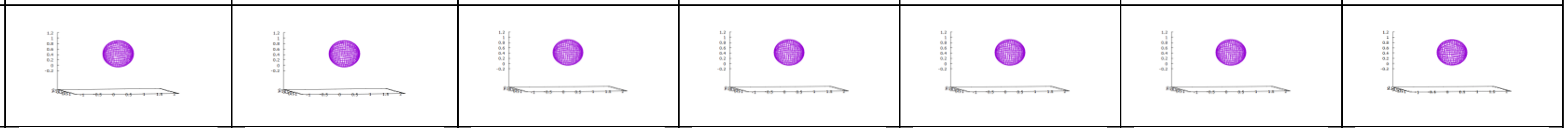
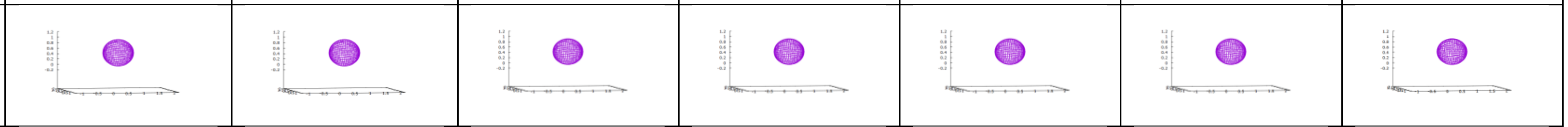
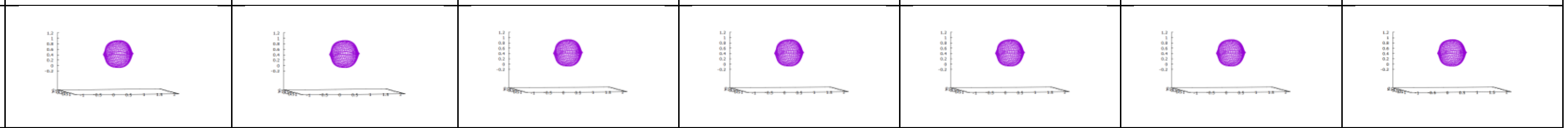
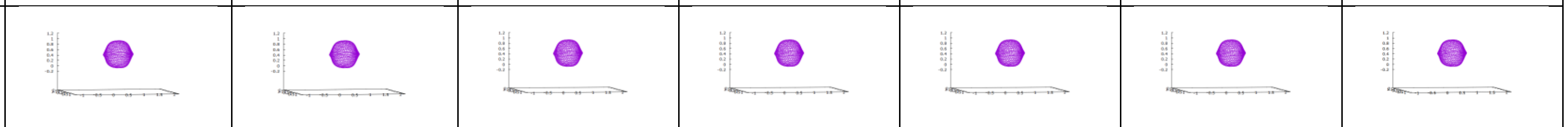
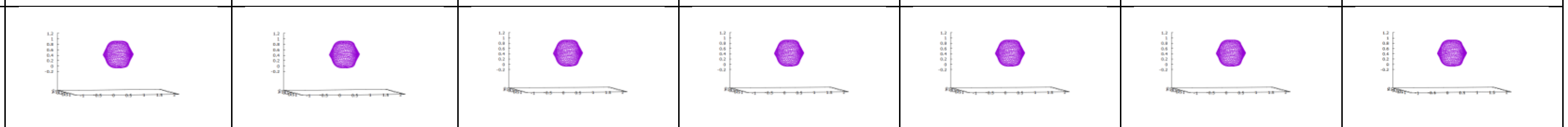
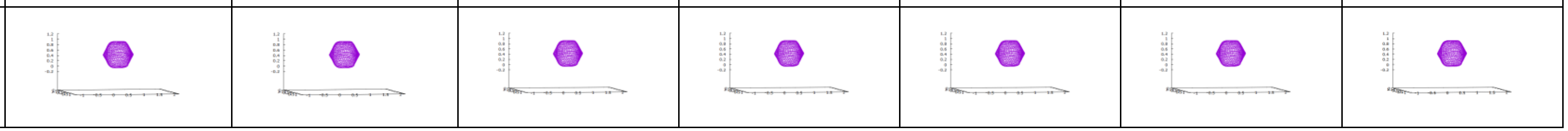
		16 × 28	
	Chord Length	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Centripeta l	4 × 16	

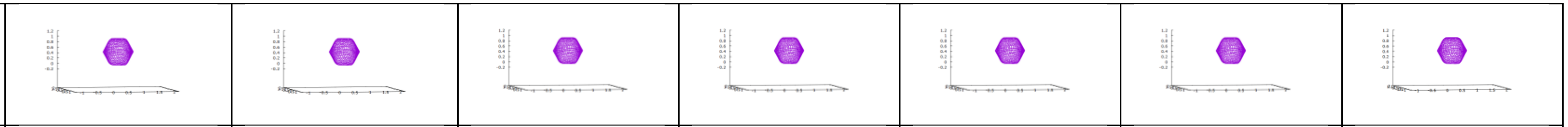
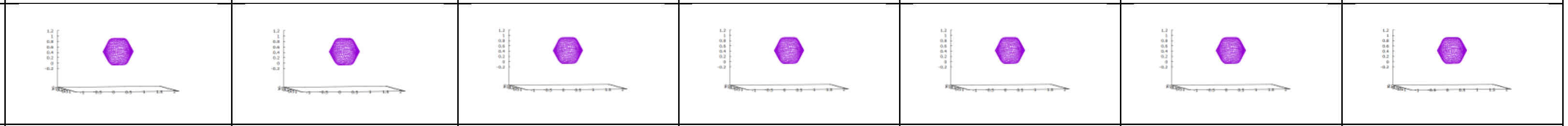
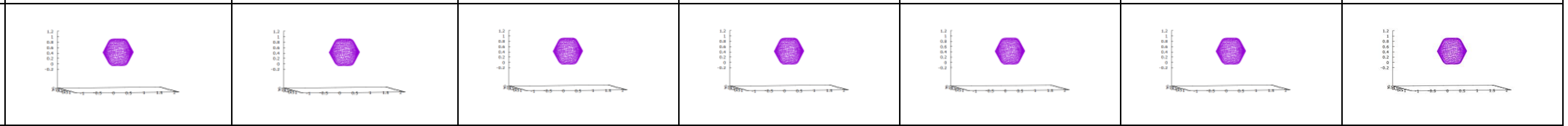
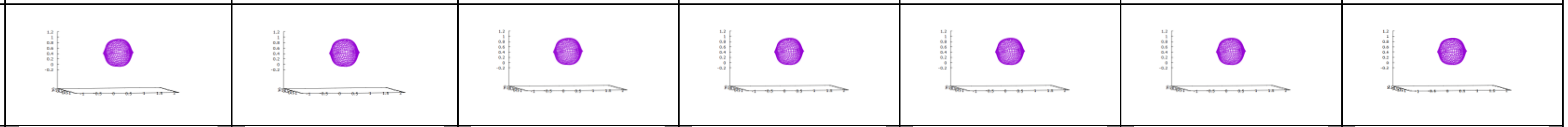
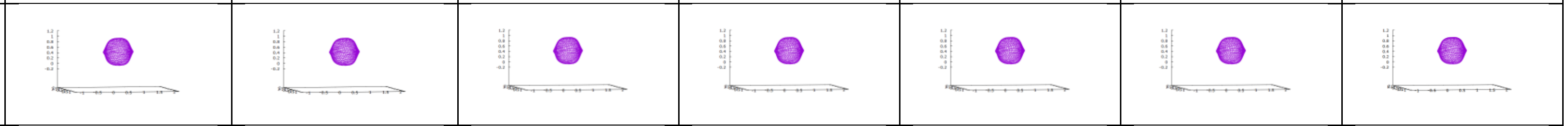
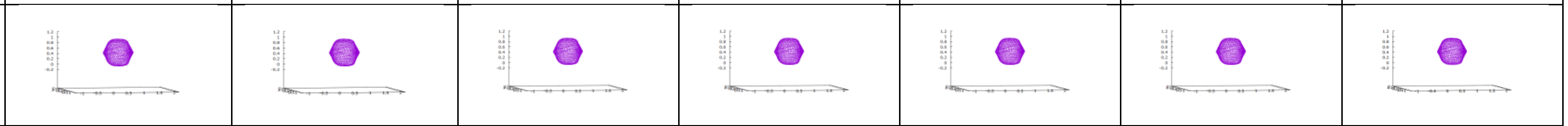
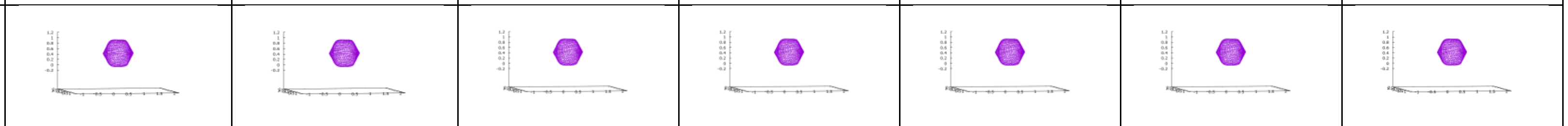
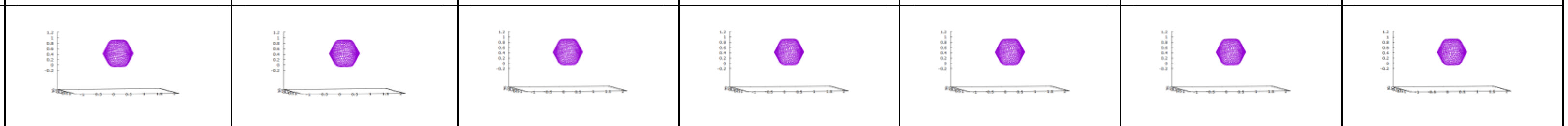
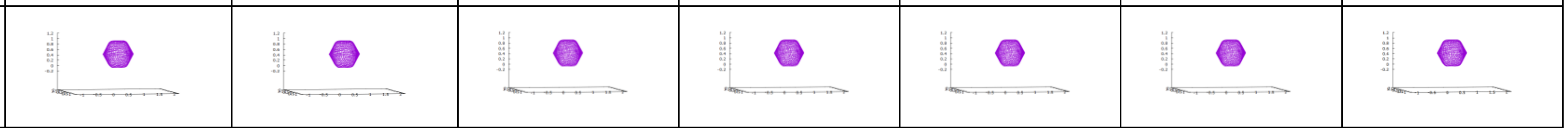
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Exponential	4 × 16	
		6 × 18	
		8 × 20	

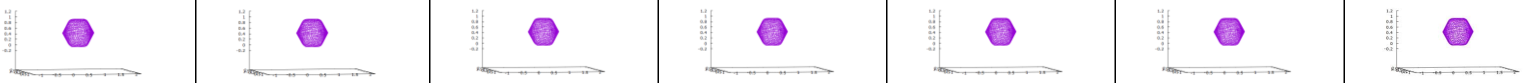
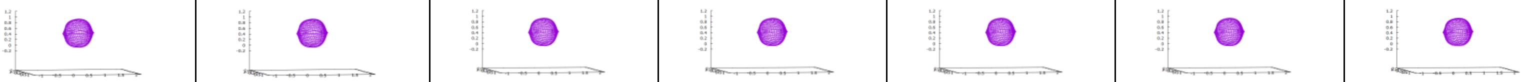
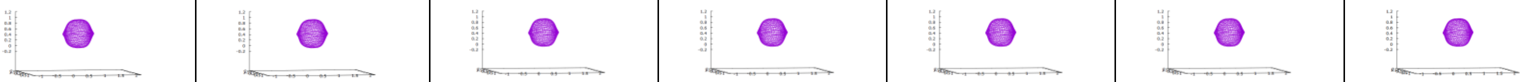
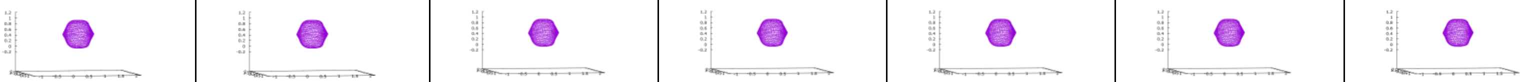
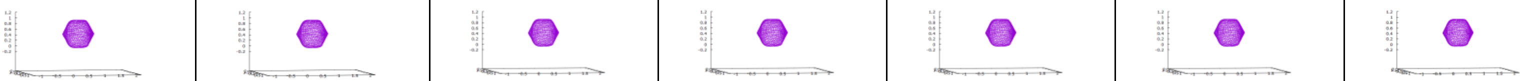
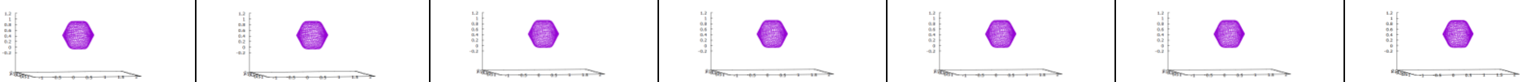
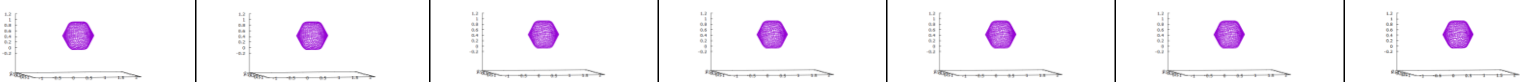
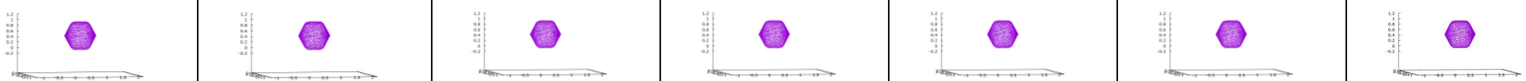
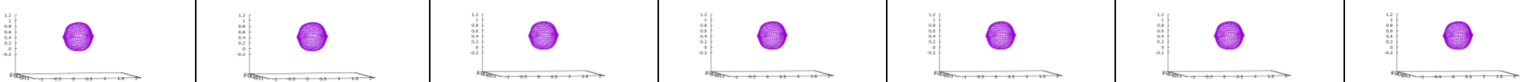
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
Sphere	Uniform	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	

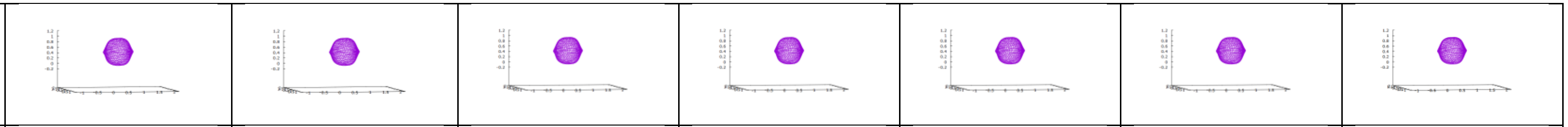
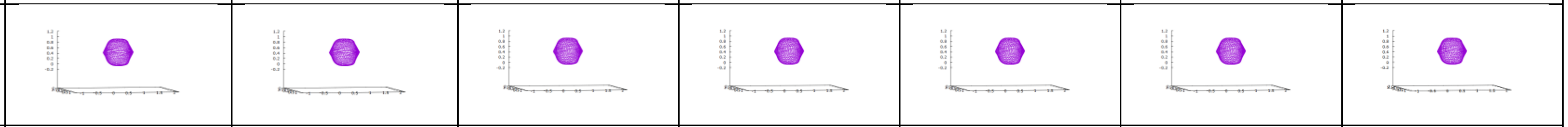
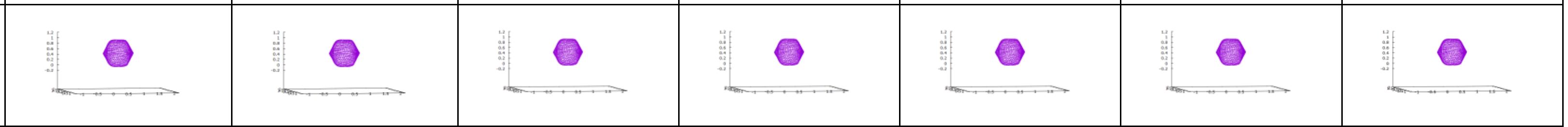
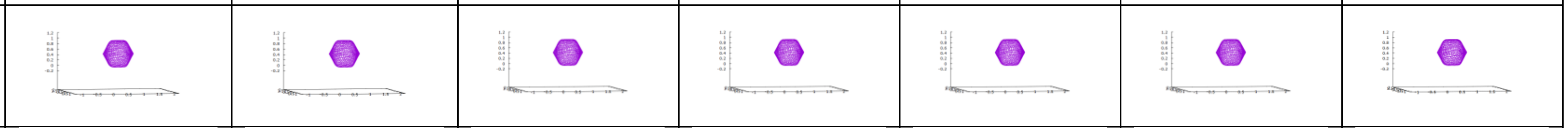
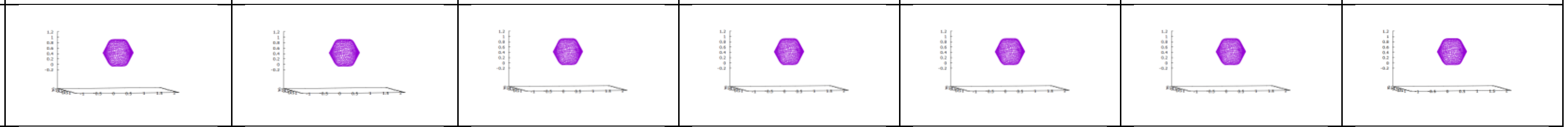
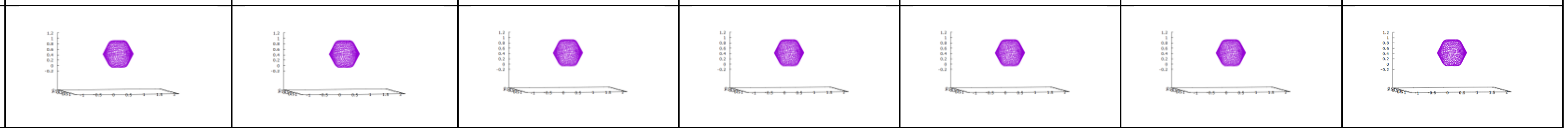
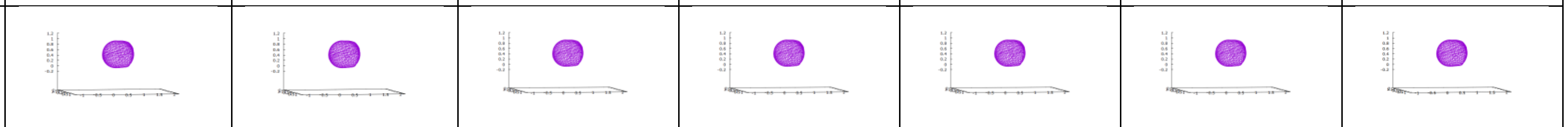
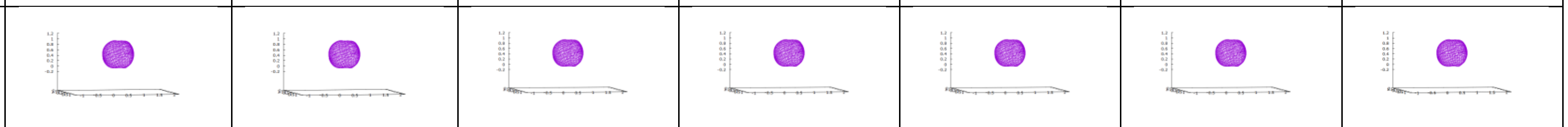
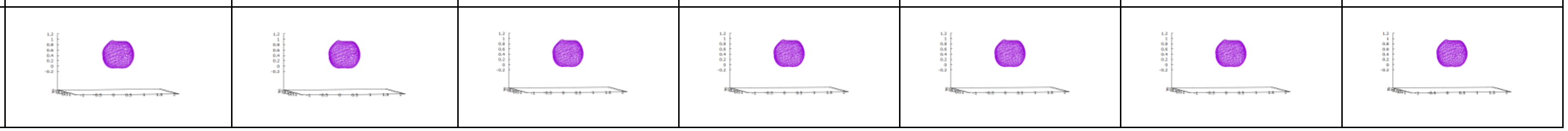
		14 × 26							
		16 × 28							
	Chord Length	4 × 16							
		6 × 18							
		8 × 20							
		10 × 22							
		12 × 24							
		14 × 26							
		16 × 28							

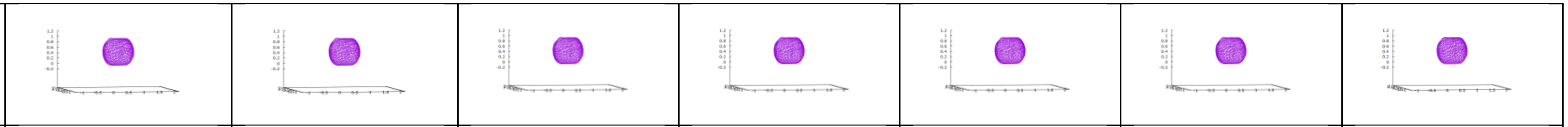
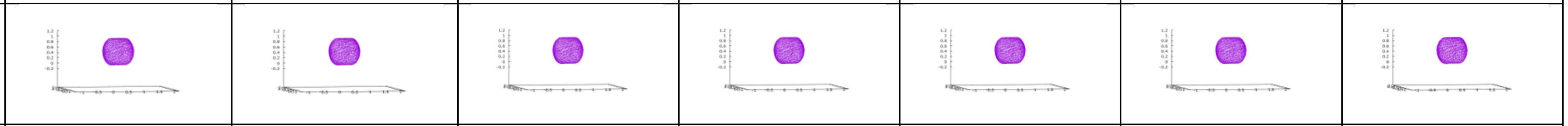
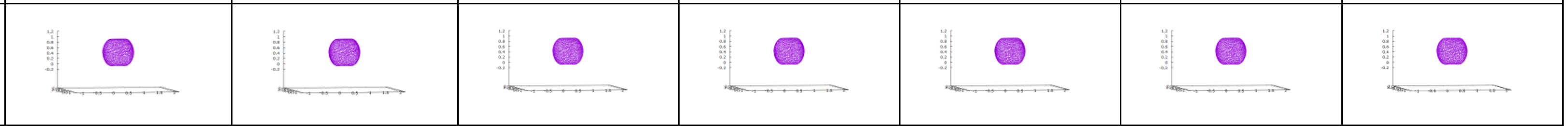
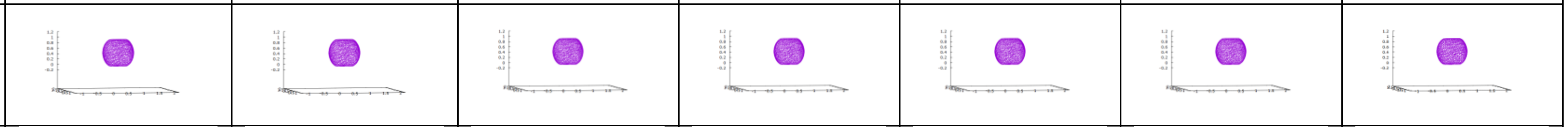
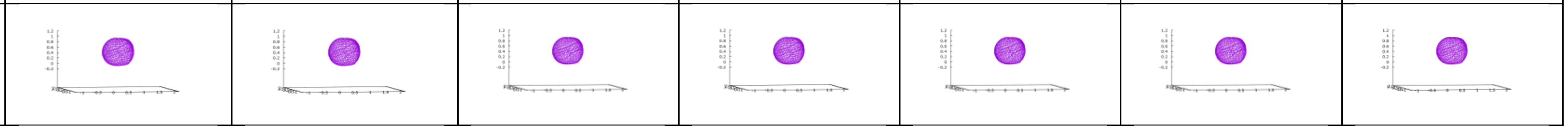
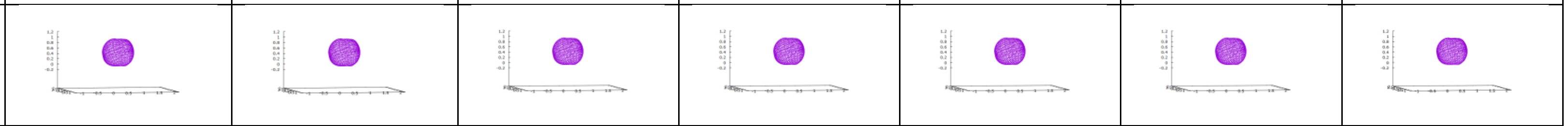
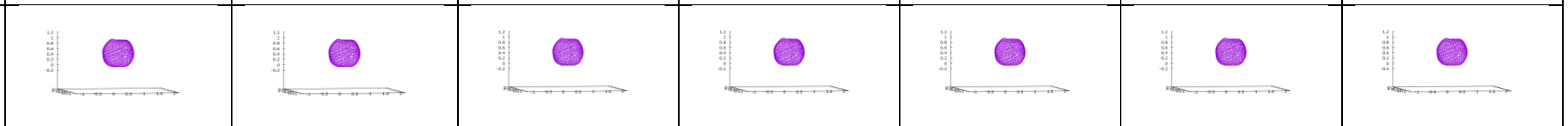
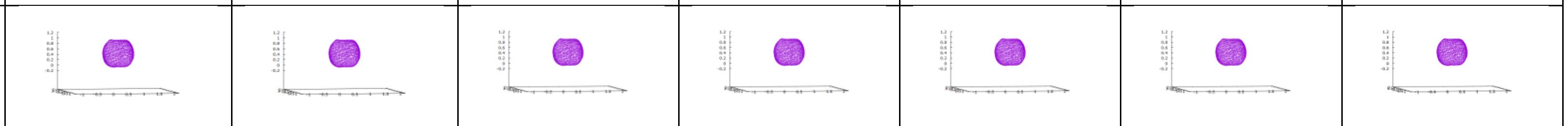
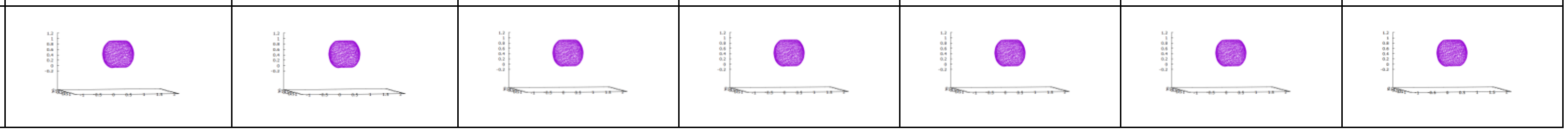
	Centripetal	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Exponential	4 × 16	
		6 × 18	

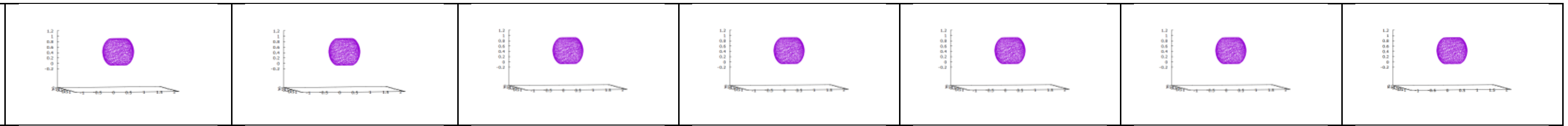
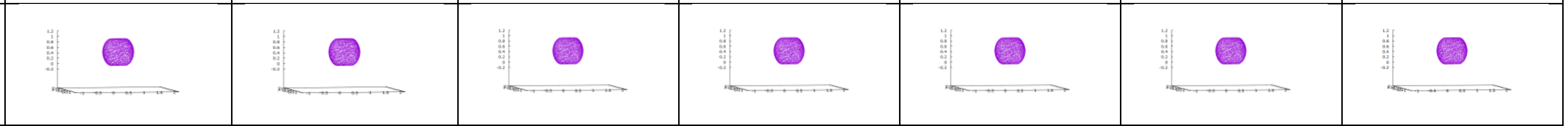
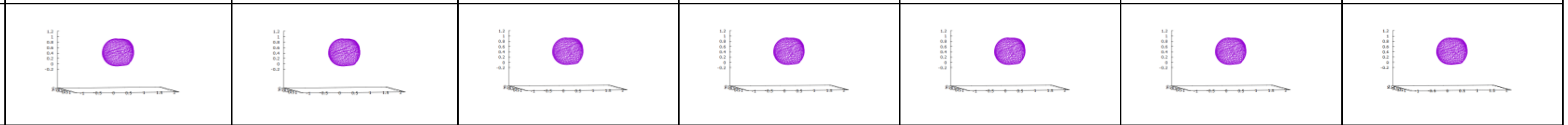
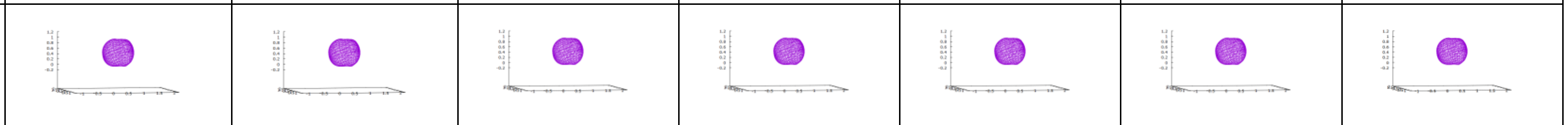
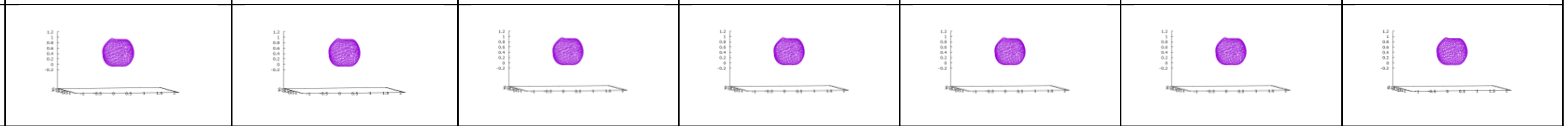
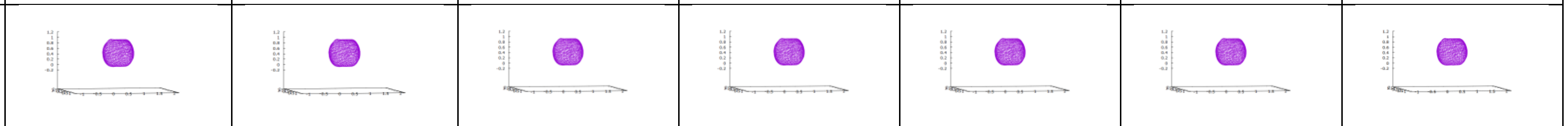
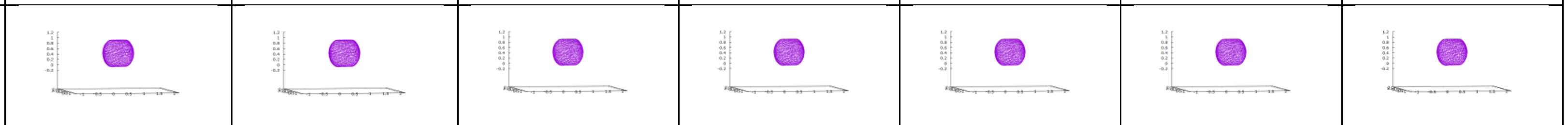
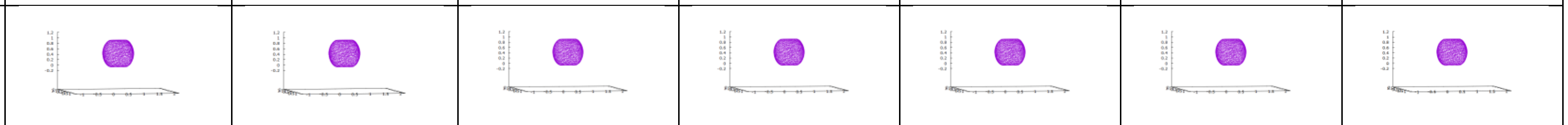
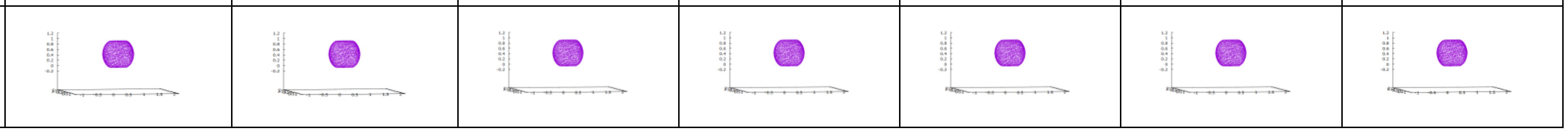
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
Spindle	Uniform	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	

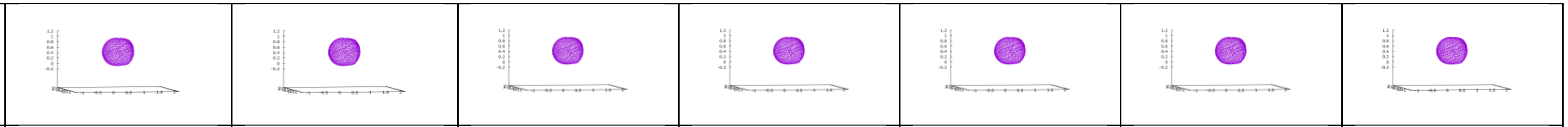
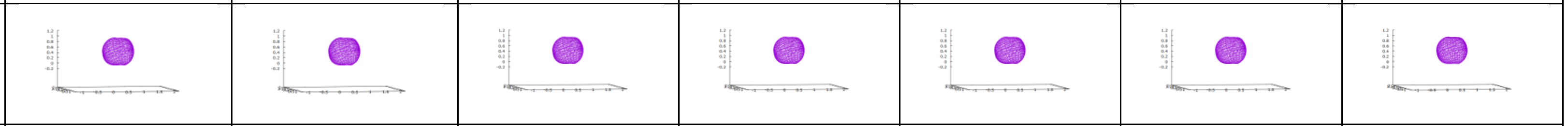
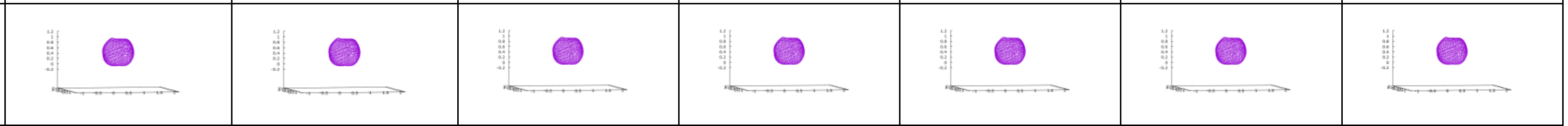
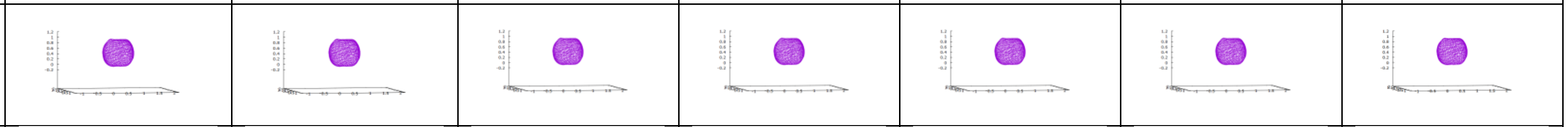
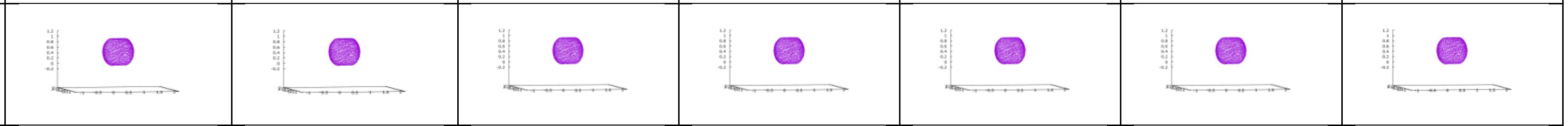
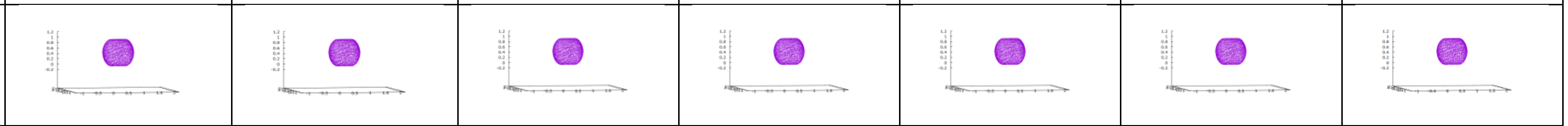
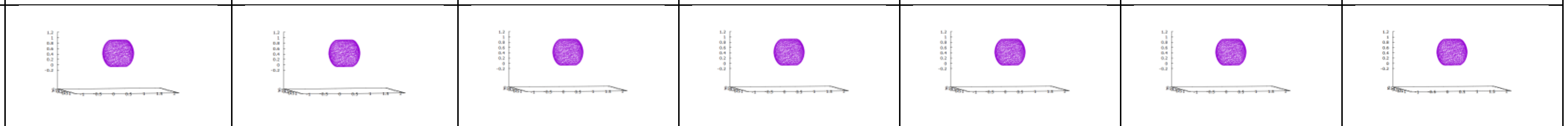
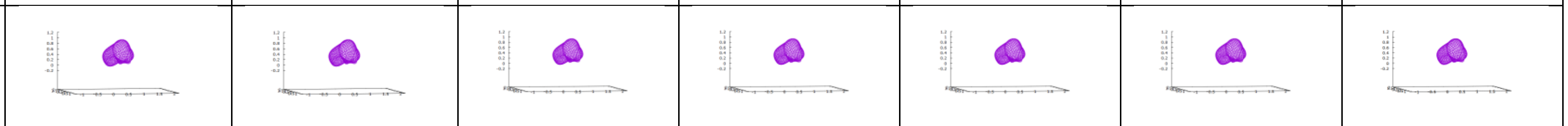
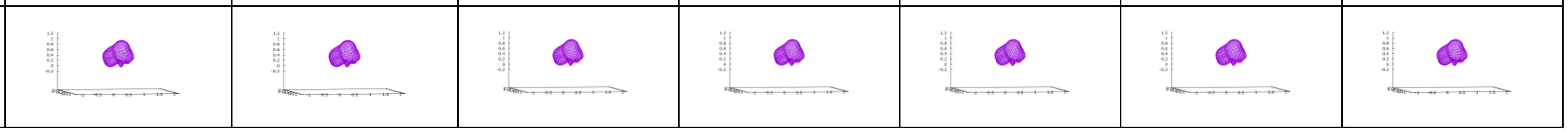
		12 × 24							
		14 × 26							
		16 × 28							
	Chord Length	4 × 16							
		6 × 18							
		8 × 20							
		10 × 22							
		12 × 24							
		14 × 26							

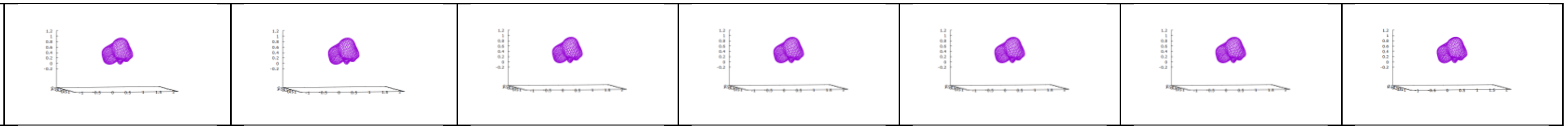
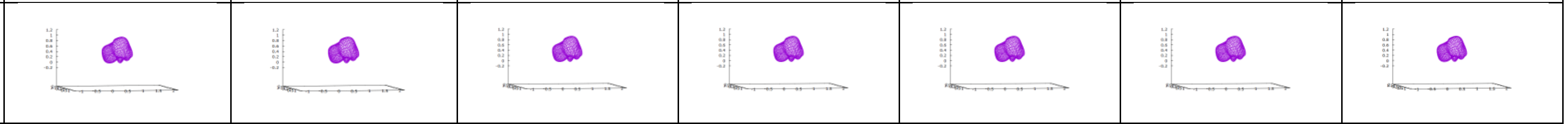
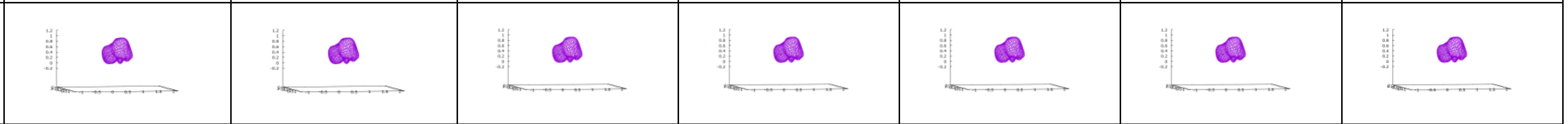
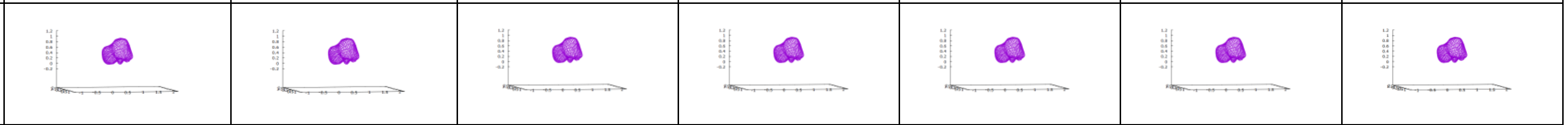
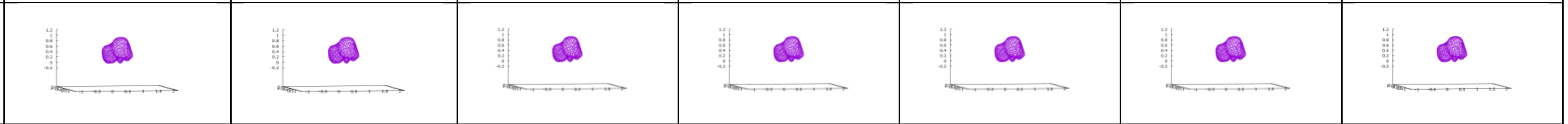
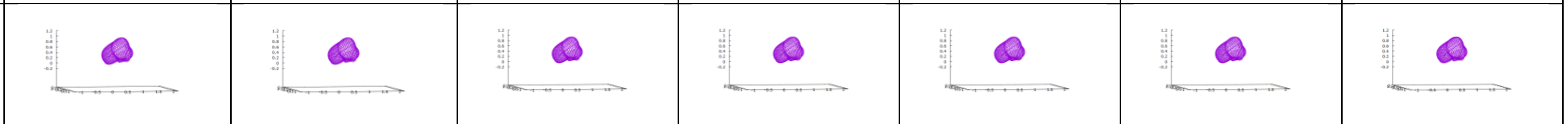
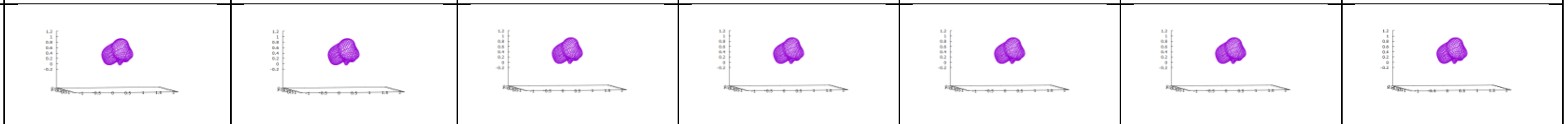
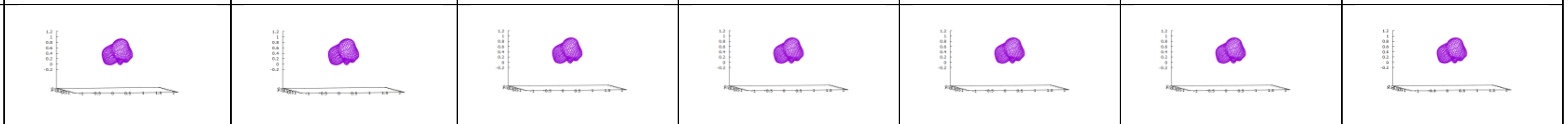
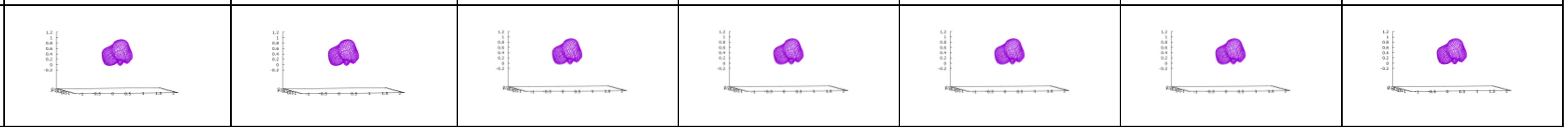
		16 × 28	
	Centripetal	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Exponenti al	4 × 16	

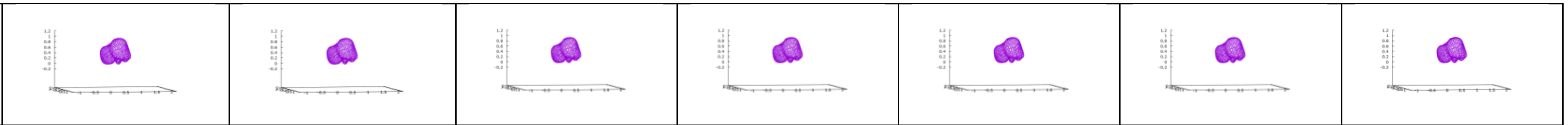
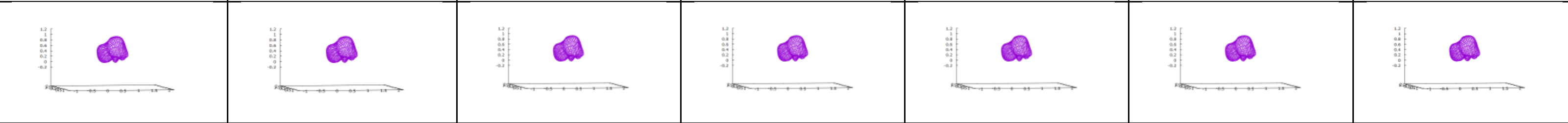
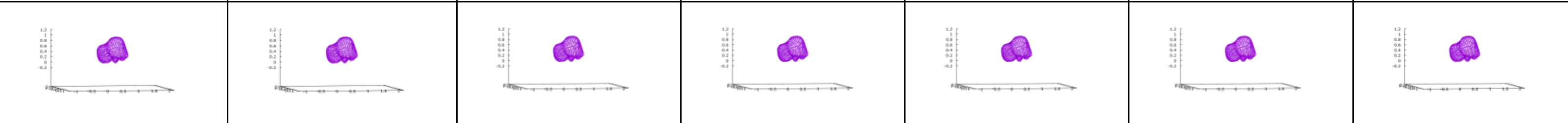
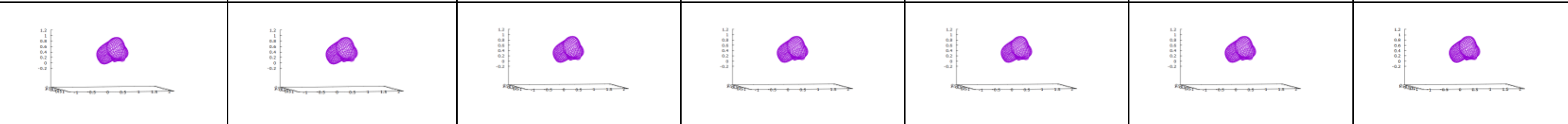
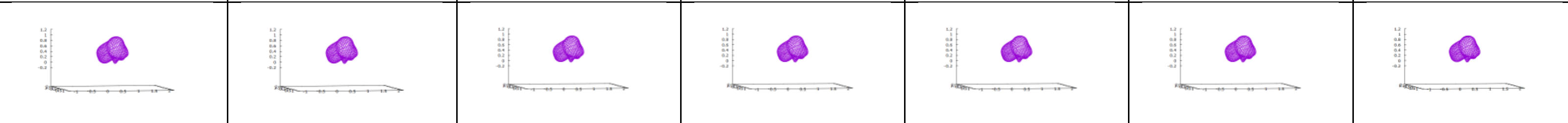
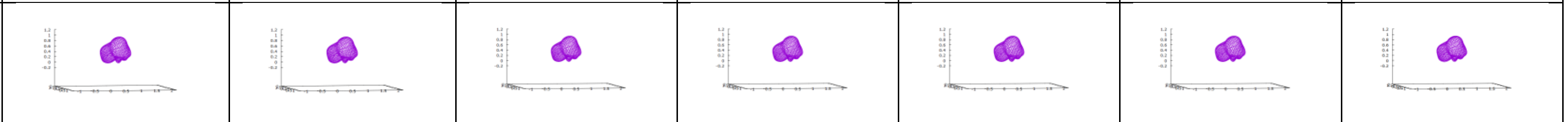
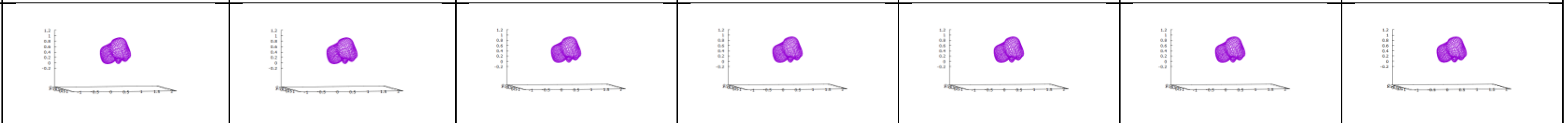
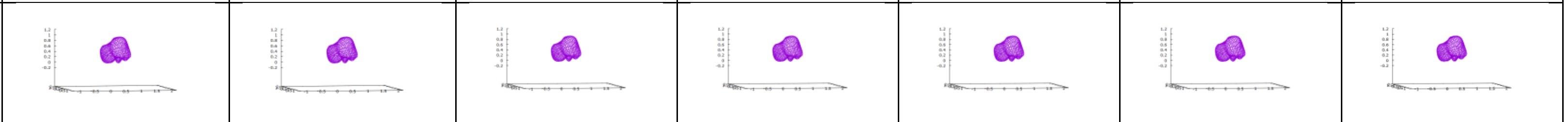
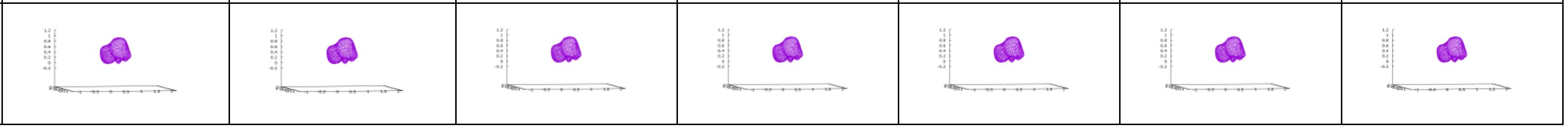
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
Oiltank	Uniform	4 × 16	
		6 × 18	
		8 × 20	

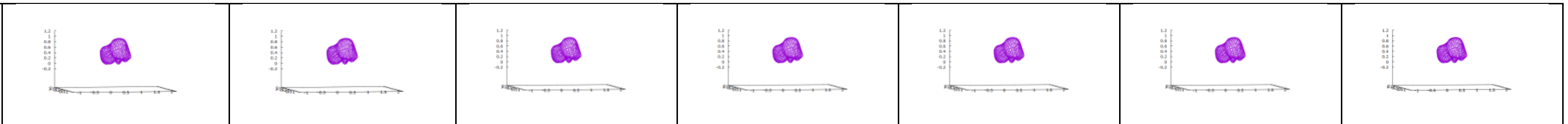
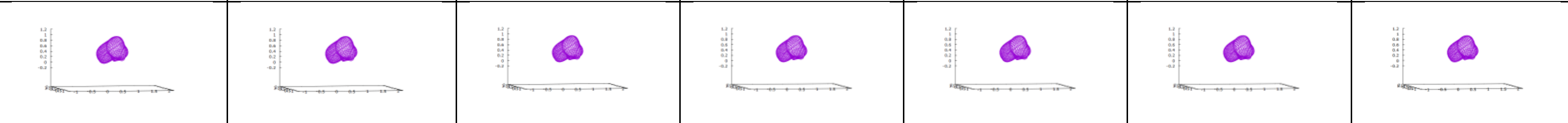
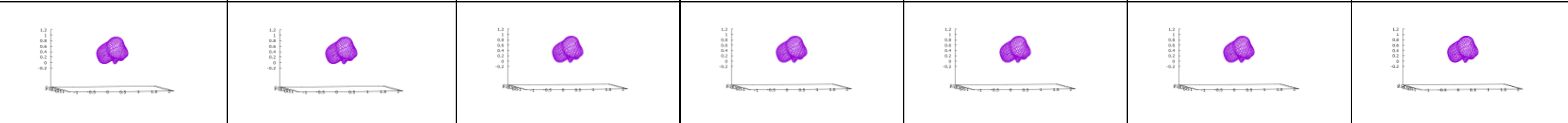
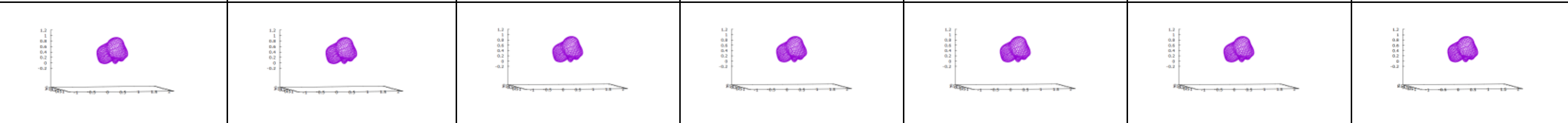
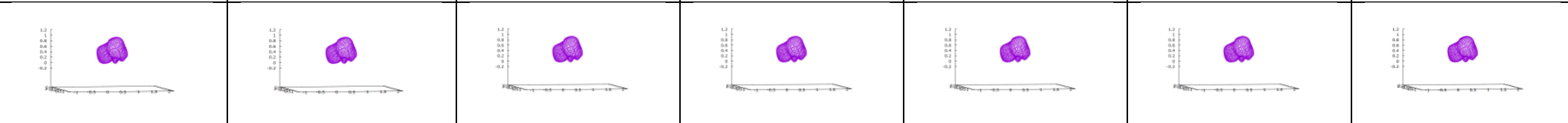
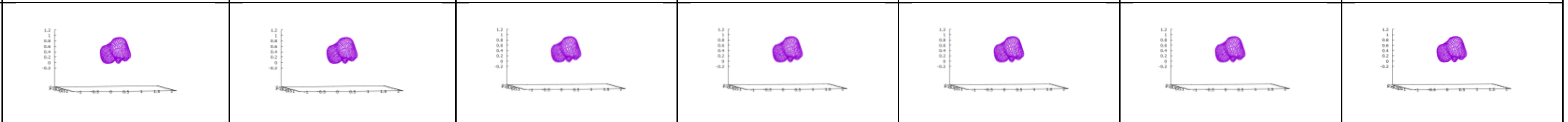
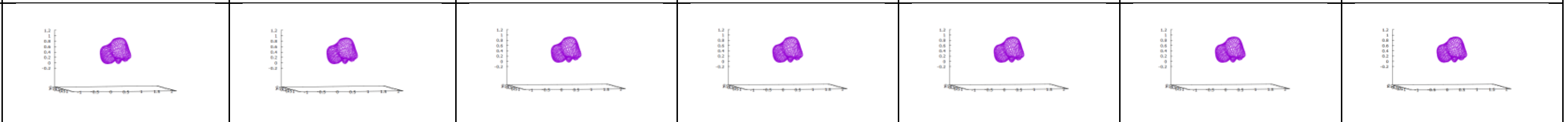
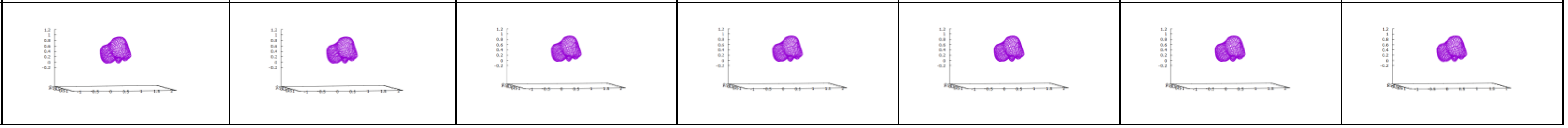
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Chord Length	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	

		14 × 26	
		16 × 28	
	Centripetal	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	

	Exponential	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
Talus Bone	Uniform	4 × 16	
		6 × 18	

		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	
	Chord Length	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	

		12 × 24	
		14 × 26	
		16 × 28	
	Centripetal	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	

		16 × 28	
	Exponential	4 × 16	
		6 × 18	
		8 × 20	
		10 × 22	
		12 × 24	
		14 × 26	
		16 × 28	

APPENDIX K: OPTIMISED SURFACE ERROR OF VARIOUS OPTIMISATION TECHNIQUES AND THE IMPROVED (*B*) SURFCE APPROXIMATION APPROACH FOR CN WITH THE SAME WIDTH AND LENGTH

Data	Parameterisation Method	CN	GA			DE			PSO			*B*
			AVG	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	
Cube	Uniform	6 × 6	32.846866	32.846843	32.846876	32.846646	32.846632	32.846663	32.846762	32.846692	32.846793	32.847252
		8 × 8	23.989220	23.989189	23.989240	23.988962	23.988949	23.988988	23.989138	23.989105	23.989168	23.989527
		10 × 10	18.808819	18.808800	18.808849	18.808504	18.808493	18.808513	18.808745	18.808727	18.808774	18.809090
		12 × 12	14.742849	14.742835	14.742869	14.742474	14.742461	14.742488	14.742781	14.742729	14.742808	14.743104
		14 × 14	10.320278	10.320251	10.320298	10.319862	10.319842	10.319885	10.320217	10.320198	10.320254	10.320493
		16 × 16	6.762083	6.762058	6.762114	6.761639	6.761612	6.761660	6.761955	6.761912	6.761991	6.762235
		18 × 18	3.489358	3.489331	3.489393	3.488944	3.488917	3.488963	3.489013	3.488982	3.489076	3.489357
	Chord Length	6 × 6	28.749832	28.749815	28.749852	28.749650	28.749642	28.749667	28.749750	28.749714	28.749800	28.750110
		8 × 8	23.294701	23.294680	23.294736	23.294443	23.294433	23.294469	23.294622	23.294551	23.294665	23.294933
		10 × 10	19.142558	19.142541	19.142576	19.142238	19.142218	19.142257	19.142487	19.142443	19.142551	19.142886
		12 × 12	14.770520	14.770500	14.770532	14.770146	14.770128	14.770156	14.770467	14.770431	14.770503	14.770796
		14 × 14	9.495319	9.495307	9.495348	9.494910	9.494886	9.494933	9.495255	9.495234	9.495289	9.495545
		16 × 16	5.790814	5.790794	5.790844	5.790399	5.790377	5.790419	5.790651	5.790562	5.790720	5.790958
		18 × 18	2.818582	2.818552	2.818609	2.818202	2.818187	2.818221	2.818221	2.818189	2.818246	2.818562
	Centripetal	6 × 6	29.934908	29.934880	29.934938	29.934694	29.934682	29.934706	29.934794	29.934758	29.934836	29.935218
		8 × 8	22.707509	22.707492	22.707525	22.707265	22.707258	22.707274	22.707426	22.707383	22.707493	22.707770
		10 × 10	18.198870	18.198861	18.198883	18.198568	18.198561	18.198585	18.198798	18.198773	18.198814	18.199131
		12 × 12	14.135104	14.135094	14.135121	14.134728	14.134702	14.134767	14.135051	14.135033	14.135068	14.135343
		14 × 14	9.428827	9.428789	9.428848	9.428442	9.428420	9.428461	9.428772	9.428743	9.428804	9.429036
		16 × 16	5.873371	5.873337	5.873393	5.872949	5.872927	5.872994	5.873227	5.873205	5.873259	5.873534
		18 × 18	2.844693	2.844662	2.844730	2.844338	2.844320	2.844359	2.844335	2.844308	2.844355	2.844669

	Exponential	6 × 6	28.952570	28.952544	28.952595	28.952373	28.952356	28.952398	28.952477	28.952431	28.952514	28.952805
		8 × 8	22.833524	22.833508	22.833540	22.833265	22.833255	22.833274	22.833433	22.833379	22.833471	22.833759
		10 × 10	18.607875	18.607847	18.607901	18.607551	18.607543	18.607565	18.607799	18.607759	18.607843	18.608152
		12 × 12	14.374870	14.374843	14.374892	14.374496	14.374481	14.374516	14.374792	14.374753	14.374834	14.375113
		14 × 14	9.372489	9.372464	9.372500	9.372089	9.372066	9.372102	9.372445	9.372427	9.372469	9.372678
		16 × 16	5.735778	5.735751	5.735795	5.735364	5.735332	5.735414	5.735610	5.735562	5.735656	5.735893
		18 × 18	2.781826	2.781800	2.781853	2.781459	2.781428	2.781497	2.781465	2.781435	2.781491	2.781822
Sphere	Uniform	6 × 6	20.203988	20.203943	20.204027	20.203678	20.203659	20.203705	20.203826	20.203726	20.203935	20.204454
		8 × 8	14.559935	14.559895	14.559971	14.559559	14.559539	14.559591	14.559808	14.559763	14.559857	14.560331
		10 × 10	11.072083	11.072046	11.072125	11.071714	11.071694	11.071739	11.071989	11.071918	11.072056	11.072429
		12 × 12	8.769416	8.769388	8.769432	8.769025	8.769009	8.769055	8.769333	8.769292	8.769385	8.769725
		14 × 14	6.826730	6.826706	6.826750	6.826313	6.826287	6.826340	6.826666	6.826640	6.826698	6.826963
		16 × 16	5.121608	5.121573	5.121627	5.121185	5.121153	5.121212	5.121449	5.121399	5.121496	5.121749
		18 × 18	2.803388	2.803359	2.803414	2.802966	2.802942	2.803006	2.803012	2.802961	2.803063	2.803381
	Chord Length	6 × 6	15.356843	15.356829	15.356857	15.356685	15.356668	15.356704	15.356771	15.356743	15.356797	15.357078
		8 × 8	11.482902	11.482895	11.482908	11.482706	11.482694	11.482714	11.482845	11.482816	11.482879	11.483114
		10 × 10	9.259928	9.259908	9.259946	9.259655	9.259641	9.259670	9.259870	9.259815	9.259929	9.260175
		12 × 12	7.401337	7.401317	7.401352	7.401014	7.400996	7.401034	7.401271	7.401231	7.401313	7.401567
		14 × 14	5.380608	5.380594	5.380625	5.380235	5.380206	5.380263	5.380531	5.380489	5.380575	5.380782
		16 × 16	3.548803	3.548779	3.548820	3.548395	3.548365	3.548433	3.548601	3.548525	3.548649	3.548917
		18 × 18	1.934292	1.934274	1.934315	1.933937	1.933910	1.933960	1.933923	1.933905	1.933932	1.934270
	Centripetal	6 × 6	16.431733	16.431711	16.431771	16.431480	16.431466	16.431499	16.431613	16.431538	16.431666	16.432150
		8 × 8	11.900710	11.900678	11.900730	11.900437	11.900425	11.900452	11.900622	11.900581	11.900678	11.901011
		10 × 10	9.326431	9.326410	9.326454	9.326139	9.326125	9.326157	9.326365	9.326337	9.326405	9.326677
		12 × 12	7.451093	7.451072	7.451107	7.450761	7.450746	7.450774	7.451030	7.451010	7.451050	7.451317
		14 × 14	5.667160	5.667133	5.667202	5.666789	5.666769	5.666801	5.667077	5.667024	5.667115	5.667360
		16 × 16	3.975629	3.975603	3.975650	3.975227	3.975209	3.975251	3.975429	3.975366	3.975501	3.975747
		18 × 18	2.126364	2.126351	2.126376	2.125997	2.125955	2.126039	2.125963	2.125915	2.126003	2.126338
	Exponential	6 × 6	15.385868	15.385846	15.385885	15.385662	15.385648	15.385684	15.385780	15.385741	15.385824	15.386218
		8 × 8	11.358126	11.358101	11.358139	11.357903	11.357892	11.357918	11.358058	11.358023	11.358080	11.358364
		10 × 10	9.097511	9.097492	9.097532	9.097234	9.097221	9.097246	9.097441	9.097414	9.097486	9.097750

		12 × 12	7.272099	7.272081	7.272120	7.271764	7.271749	7.271781	7.272038	7.271999	7.272061	7.272313
		14 × 14	5.387106	5.387088	5.387121	5.386737	5.386723	5.386756	5.387032	5.386996	5.387062	5.387289
		16 × 16	3.627674	3.627665	3.627691	3.627268	3.627247	3.627301	3.627472	3.627419	3.627541	3.627790
		18 × 18	1.966146	1.966123	1.966164	1.965785	1.965737	1.965812	1.965768	1.965735	1.965791	1.966122
Spindle	Uniform	6 × 6	16.618088	16.618063	16.618100	16.617935	16.617921	16.617952	16.618018	16.617956	16.618075	16.618302
		8 × 8	14.029460	14.029448	14.029480	14.029236	14.029220	14.029254	14.029394	14.029363	14.029419	14.029662
		10 × 10	11.426878	11.426857	11.426892	11.426593	11.426580	11.426620	11.426807	11.426790	11.426833	11.427097
		12 × 12	9.053164	9.053143	9.053186	9.052816	9.052796	9.052844	9.053110	9.053089	9.053144	9.053436
		14 × 14	6.704022	6.703993	6.704038	6.703612	6.703592	6.703628	6.703967	6.703923	6.703998	6.704238
		16 × 16	4.583129	4.583114	4.583161	4.582716	4.582688	4.582728	4.582961	4.582896	4.583003	4.583246
		18 × 18	2.514672	2.514641	2.514695	2.514279	2.514257	2.514307	2.514313	2.514295	2.514331	2.514656
	Chord Length	6 × 6	15.317080	15.317061	15.317091	15.316917	15.316907	15.316933	15.317006	15.316973	15.317042	15.317307
		8 × 8	13.190718	13.190699	13.190737	13.190518	13.190507	13.190527	13.190657	13.190622	13.190680	13.190979
		10 × 10	10.649905	10.649892	10.649932	10.649638	10.649623	10.649656	10.649837	10.649813	10.649861	10.650167
		12 × 12	8.318244	8.318226	8.318269	8.317902	8.317889	8.317915	8.318196	8.318173	8.318216	8.318483
		14 × 14	5.905771	5.905748	5.905784	5.905408	5.905392	5.905422	5.905713	5.905680	5.905739	5.905985
		16 × 16	3.768221	3.768197	3.768241	3.767844	3.767809	3.767879	3.768011	3.767966	3.768069	3.768353
		18 × 18	1.916861	1.916833	1.916884	1.916495	1.916466	1.916529	1.916481	1.916449	1.916515	1.916830
	Centripetal	6 × 6	15.585459	15.585445	15.585475	15.585299	15.585291	15.585316	15.585384	15.585354	15.585434	15.585659
		8 × 8	13.316914	13.316900	13.316931	13.316710	13.316705	13.316722	13.316860	13.316811	13.316898	13.317136
		10 × 10	10.780726	10.780708	10.780747	10.780447	10.780435	10.780464	10.780653	10.780626	10.780669	10.780897
		12 × 12	8.474587	8.474577	8.474606	8.474232	8.474214	8.474249	8.474533	8.474461	8.474572	8.474809
		14 × 14	6.126589	6.126571	6.126607	6.126192	6.126174	6.126218	6.126527	6.126470	6.126561	6.126793
		16 × 16	3.971608	3.971582	3.971624	3.971230	3.971208	3.971263	3.971416	3.971383	3.971463	3.971729
		18 × 18	2.050172	2.050156	2.050202	2.049828	2.049817	2.049847	2.049791	2.049769	2.049836	2.050128
	Exponential	6 × 6	15.344122	15.344103	15.344140	15.343960	15.343948	15.343968	15.344039	15.344024	15.344081	15.344327
		8 × 8	13.172991	13.172967	13.173009	13.172799	13.172787	13.172813	13.172939	13.172910	13.172973	13.173216
		10 × 10	10.637423	10.637403	10.637437	10.637157	10.637144	10.637179	10.637364	10.637319	10.637402	10.637621
		12 × 12	8.323099	8.323087	8.323112	8.322762	8.322748	8.322789	8.323043	8.323024	8.323069	8.323326
		14 × 14	5.946208	5.946189	5.946228	5.945827	5.945800	5.945840	5.946144	5.946111	5.946178	5.946391
		16 × 16	3.797351	3.797336	3.797372	3.796966	3.796931	3.797007	3.797138	3.797087	3.797194	3.797467

		18 × 18	1.936860	1.936827	1.936902	1.936503	1.936483	1.936517	1.936487	1.936468	1.936501	1.936817
Oiltank	Uniform	6 × 6	22.128927	22.128883	22.128966	22.128687	22.128667	22.128726	22.128816	22.128760	22.128898	22.129226
		8 × 8	17.706385	17.706363	17.706396	17.706090	17.706078	17.706102	17.706267	17.706226	17.706305	17.706700
		10 × 10	13.164475	13.164465	13.164499	13.164144	13.164127	13.164158	13.164385	13.164369	13.164404	13.164787
		12 × 12	9.669612	9.669588	9.669633	9.669230	9.669214	9.669254	9.669557	9.669528	9.669599	9.669914
		14 × 14	7.025197	7.025174	7.025224	7.024766	7.024747	7.024800	7.025123	7.025092	7.025159	7.025431
		16 × 16	4.602799	4.602779	4.602829	4.602374	4.602351	4.602402	4.602617	4.602558	4.602665	4.602955
		18 × 18	2.702855	2.702826	2.702892	2.702458	2.702431	2.702497	2.702468	2.702441	2.702489	2.702826
	Chord Length	6 × 6	19.659938	19.659923	19.659956	19.659774	19.659755	19.659799	19.659845	19.659780	19.659871	19.660172
		8 × 8	15.407697	15.407672	15.407721	15.407472	15.407455	15.407500	15.407627	15.407580	15.407661	15.407933
		10 × 10	11.206485	11.206474	11.206505	11.206199	11.206188	11.206215	11.206423	11.206386	11.206463	11.206745
		12 × 12	8.182350	8.182324	8.182364	8.182003	8.181992	8.182015	8.182281	8.182249	8.182304	8.182572
		14 × 14	5.659288	5.659266	5.659304	5.658900	5.658877	5.658914	5.659233	5.659201	5.659260	5.659485
		16 × 16	3.368184	3.368160	3.368197	3.367790	3.367768	3.367816	3.367938	3.367897	3.367984	3.368280
		18 × 18	1.892461	1.892422	1.892485	1.892096	1.892073	1.892115	1.892063	1.892046	1.892091	1.892390
	Centripetal	6 × 6	20.215643	20.215611	20.215671	20.215441	20.215429	20.215452	20.215547	20.215478	20.215596	20.215953
		8 × 8	15.906606	15.906591	15.906619	15.906355	15.906335	15.906373	15.906526	15.906489	15.906544	15.906888
		10 × 10	11.575247	11.575224	11.575264	11.574943	11.574933	11.574961	11.575178	11.575156	11.575197	11.575495
		12 × 12	8.406694	8.406676	8.406707	8.406337	8.406325	8.406350	8.406626	8.406604	8.406643	8.406920
		14 × 14	5.980795	5.980779	5.980818	5.980400	5.980384	5.980412	5.980721	5.980681	5.980764	5.980997
		16 × 16	3.673238	3.673210	3.673282	3.672846	3.672827	3.672860	3.673003	3.672974	3.673041	3.673360
		18 × 18	2.070024	2.069999	2.070050	2.069673	2.069651	2.069703	2.069643	2.069621	2.069677	2.069978
	Exponential	6 × 6	19.651202	19.651176	19.651214	19.651028	19.651009	19.651054	19.651120	19.651071	19.651189	19.651440
		8 × 8	15.392180	15.392165	15.392193	15.391956	15.391944	15.391972	15.392115	15.392057	15.392145	15.392426
		10 × 10	11.150978	11.150959	11.151001	11.150699	11.150689	11.150711	11.150917	11.150871	11.150975	11.151213
		12 × 12	8.123162	8.123134	8.123188	8.122826	8.122810	8.122856	8.123094	8.123070	8.123109	8.123378
		14 × 14	5.689606	5.689573	5.689627	5.689210	5.689189	5.689228	5.689537	5.689503	5.689563	5.689804
		16 × 16	3.412465	3.412444	3.412487	3.412073	3.412060	3.412100	3.412226	3.412174	3.412282	3.412576
		18 × 18	1.921121	1.921103	1.921135	1.920772	1.920751	1.920797	1.920740	1.920703	1.920772	1.921040
Talus Bone	Unifor m	6 × 6	27.183877	27.183848	27.183922	27.183623	27.183611	27.183648	27.183761	27.183698	27.183821	27.184267
		8 × 8	19.790979	19.790957	19.791002	19.790710	19.790696	19.790722	19.790893	19.790849	19.790946	19.791257

		10 × 10	14.465811	14.465779	14.465842	14.465488	14.465478	14.465499	14.465732	14.465697	14.465777	14.466086
		12 × 12	11.273971	11.273964	11.273979	11.273607	11.273592	11.273622	11.273913	11.273891	11.273943	11.274231
		14 × 14	7.631272	7.631250	7.631290	7.630872	7.630857	7.630888	7.631205	7.631165	7.631253	7.631480
		16 × 16	5.218803	5.218775	5.218826	5.218412	5.218382	5.218439	5.218611	5.218566	5.218660	5.218926
		18 × 18	2.833902	2.833882	2.833930	2.833502	2.833458	2.833525	2.833501	2.833437	2.833533	2.833904
	Chord Length	6 × 6	25.675380	25.675353	25.675410	25.675169	25.675153	25.675183	25.675294	25.675260	25.675355	25.675751
		8 × 8	18.592853	18.592840	18.592869	18.592625	18.592609	18.592631	18.592774	18.592728	18.592840	18.593093
		10 × 10	14.150553	14.150529	14.150573	14.150275	14.150267	14.150290	14.150490	14.150455	14.150522	14.150769
		12 × 12	10.853885	10.853869	10.853903	10.853539	10.853531	10.853549	10.853824	10.853810	10.853843	10.854089
		14 × 14	6.617837	6.617818	6.617858	6.617476	6.617462	6.617505	6.617774	6.617749	6.617802	6.618030
		16 × 16	4.252371	4.252341	4.252387	4.252002	4.251982	4.252027	4.252145	4.252113	4.252181	4.252484
		18 × 18	2.045126	2.045099	2.045172	2.044746	2.044722	2.044779	2.044726	2.044684	2.044764	2.045113
	Centripetal	6 × 6	26.202041	26.202003	26.202071	26.201808	26.201787	26.201836	26.201946	26.201891	26.202037	26.202397
		8 × 8	18.876213	18.876189	18.876243	18.875977	18.875961	18.875990	18.876145	18.876110	18.876177	18.876482
		10 × 10	13.965323	13.965310	13.965344	13.965035	13.965024	13.965044	13.965259	13.965220	13.965297	13.965549
		12 × 12	10.767965	10.767953	10.767976	10.767632	10.767612	10.767646	10.767898	10.767877	10.767917	10.768179
		14 × 14	6.920044	6.920011	6.920060	6.919671	6.919652	6.919695	6.919987	6.919954	6.920017	6.920247
		16 × 16	4.559952	4.559940	4.559968	4.559594	4.559570	4.559610	4.559721	4.559664	4.559772	4.560049
		18 × 18	2.273549	2.273525	2.273564	2.273172	2.273151	2.273201	2.273138	2.273108	2.273180	2.273533
	Exponential	6 × 6	25.831085	25.831049	25.831118	25.830859	25.830848	25.830867	25.831003	25.830969	25.831023	25.831448
		8 × 8	18.626292	18.626269	18.626316	18.626055	18.626042	18.626070	18.626217	18.626180	18.626250	18.626535
		10 × 10	13.996566	13.996549	13.996589	13.996298	13.996290	13.996308	13.996495	13.996480	13.996514	13.996782
		12 × 12	10.754648	10.754629	10.754665	10.754310	10.754300	10.754321	10.754585	10.754562	10.754626	10.754879
		14 × 14	6.676344	6.676328	6.676356	6.675982	6.675960	6.675991	6.676285	6.676260	6.676332	6.676524
		16 × 16	4.326860	4.326826	4.326892	4.326499	4.326483	4.326530	4.326641	4.326596	4.326693	4.326967
		18 × 18	2.099645	2.099619	2.099670	2.099273	2.099256	2.099290	2.099241	2.099214	2.099279	2.099644

APPENDIX L: OPTIMISED SURFACE ERROR OF VARIOUS OPTIMISATION TECHNIQUES AND THE IMPROVED (*B*) SURFCE APPROXIMATION APPROACH FOR CN WITH DIFFERENT WIDTH AND LENGTH

Data	Parameterisation Method	CN	GA			DE			PSO			*B*
			AVG	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	
Cube	Uniform	4 × 16	45.795469	45.795430	45.795512	45.794933	45.794910	45.794954	45.795303	45.795241	45.795398	45.796012
		6 × 18	29.281720	29.281704	29.281747	29.281289	29.281270	29.281306	29.281598	29.281558	29.281632	29.282095
		8 × 20	19.655804	19.655775	19.655825	19.655324	19.655311	19.655340	19.655716	19.655666	19.655767	19.656191
		10 × 22	15.254146	15.254115	15.254174	15.253625	15.253593	15.253645	15.254079	15.254026	15.254119	15.254454
		12 × 24	10.924108	10.924061	10.924143	10.923597	10.923571	10.923629	10.923998	10.923975	10.924032	10.924310
		14 × 26	7.262369	7.262345	7.262407	7.261853	7.261821	7.261887	7.261972	7.261913	7.262030	7.262420
		16 × 28	3.874813	3.874776	3.874854	3.874307	3.874267	3.874366	3.874269	3.874232	3.874342	3.874686
	Chord Length	4 × 16	38.416339	38.416308	38.416396	38.415920	38.415899	38.415947	38.416196	38.416142	38.416249	38.416722
		6 × 18	28.368302	28.368260	28.368338	28.367879	28.367853	28.367894	28.368195	28.368122	28.368251	28.368689
		8 × 20	20.598419	20.598402	20.598445	20.597969	20.597942	20.597987	20.598345	20.598295	20.598391	20.598706
		10 × 22	16.925949	16.925927	16.925987	16.925407	16.925385	16.925448	16.925881	16.925816	16.925946	16.926239
		12 × 24	11.865427	11.865377	11.865459	11.864881	11.864850	11.864908	11.865332	11.865266	11.865371	11.865611
		14 × 26	6.946973	6.946942	6.947008	6.946444	6.946399	6.946470	6.946644	6.946590	6.946708	6.947204
		16 × 28	2.906980	2.906955	2.907011	2.906523	2.906487	2.906589	2.906451	2.906394	2.906476	2.906806
	Centripetal	4 × 16	41.097548	41.097516	41.097595	41.097078	41.097043	41.097103	41.097409	41.097351	41.097502	41.098115
		6 × 18	28.054884	28.054856	28.054927	28.054462	28.054445	28.054495	28.054782	28.054731	28.054832	28.055285
		8 × 20	19.360705	19.360682	19.360734	19.360250	19.360234	19.360266	19.360626	19.360563	19.360671	19.360944
		10 × 22	15.469135	15.469099	15.469164	15.468631	15.468603	15.468650	15.469063	15.469028	15.469093	15.469367
		12 × 24	10.978080	10.978044	10.978123	10.977568	10.977549	10.977608	10.977954	10.977911	10.977998	10.978313
		14 × 26	6.827747	6.827722	6.827760	6.827236	6.827213	6.827261	6.827324	6.827295	6.827400	6.827783
		16 × 28	3.173905	3.173854	3.173940	3.173451	3.173401	3.173479	3.173389	3.173342	3.173408	3.173775

	Exponential	4 × 16	39.101872	39.101841	39.101888	39.101410	39.101387	39.101422	39.101707	39.101638	39.101794	39.102375
		6 × 18	28.037824	28.037811	28.037843	28.037397	28.037377	28.037426	28.037706	28.037665	28.037765	28.038141
		8 × 20	19.931909	19.931889	19.931946	19.931456	19.931438	19.931469	19.931814	19.931784	19.931872	19.932207
		10 × 22	16.220973	16.220957	16.220996	16.220454	16.220428	16.220489	16.220903	16.220868	16.220961	16.221275
		12 × 24	11.437603	11.437565	11.437636	11.437082	11.437056	11.437123	11.437482	11.437425	11.437543	11.437806
		14 × 26	6.843880	6.843828	6.843918	6.843357	6.843319	6.843403	6.843497	6.843451	6.843533	6.843952
		16 × 28	2.970620	2.970591	2.970671	2.970184	2.970159	2.970215	2.970091	2.970040	2.970131	2.970490
Sphere	Uniform	4 × 16	14.902661	14.902636	14.902686	14.902148	14.902131	14.902180	14.902526	14.902463	14.902578	14.903185
		6 × 18	11.262480	11.262431	11.262515	11.262002	11.261990	11.262018	11.262368	11.262328	11.262411	11.262854
		8 × 20	9.377889	9.377857	9.377920	9.377409	9.377401	9.377421	9.377801	9.377757	9.377846	9.378245
		10 × 22	7.544304	7.544282	7.544338	7.543805	7.543778	7.543835	7.544237	7.544210	7.544288	7.544603
		12 × 24	5.801236	5.801213	5.801278	5.800757	5.800712	5.800784	5.801076	5.801034	5.801122	5.801424
		14 × 26	4.142463	4.142443	4.142479	4.141992	4.141950	4.142035	4.142055	4.141984	4.142078	4.142489
		16 × 28	2.364438	2.364407	2.364463	2.363941	2.363871	2.363986	2.363915	2.363900	2.363935	2.364249
	Chord Length	4 × 16	15.468706	15.468659	15.468741	15.468244	15.468226	15.468270	15.468572	15.468502	15.468644	15.469229
		6 × 18	11.147795	11.147784	11.147819	11.147459	11.147441	11.147480	11.147716	11.147677	11.147752	11.148058
		8 × 20	9.057850	9.057832	9.057867	9.057489	9.057474	9.057500	9.057787	9.057745	9.057834	9.058099
		10 × 22	7.138467	7.138436	7.138478	7.138019	7.137994	7.138037	7.138407	7.138366	7.138475	7.138703
		12 × 24	5.243005	5.242969	5.243034	5.242563	5.242541	5.242620	5.242811	5.242760	5.242869	5.243158
		14 × 26	3.534250	3.534236	3.534279	3.533816	3.533780	3.533852	3.533811	3.533790	3.533833	3.534287
		16 × 28	1.762630	1.762580	1.762662	1.762201	1.762167	1.762224	1.762099	1.762051	1.762132	1.762445
	Centripetal	4 × 16	13.902208	13.902179	13.902262	13.901781	13.901762	13.901811	13.902073	13.902030	13.902139	13.902640
		6 × 18	10.732429	10.732413	10.732467	10.732082	10.732073	10.732096	10.732360	10.732312	10.732390	10.732737
		8 × 20	8.908633	8.908615	8.908648	8.908222	8.908216	8.908230	8.908557	8.908527	8.908595	8.908904
		10 × 22	7.138571	7.138554	7.138606	7.138106	7.138086	7.138128	7.138487	7.138434	7.138528	7.138793
		12 × 24	5.379987	5.379969	5.380006	5.379528	5.379495	5.379560	5.379814	5.379740	5.379860	5.380170
		14 × 26	3.703290	3.703273	3.703307	3.702853	3.702807	3.702888	3.702850	3.702821	3.702878	3.703292
		16 × 28	1.946766	1.946741	1.946796	1.946309	1.946284	1.946351	1.946213	1.946176	1.946243	1.946529
	Exponential	4 × 16	14.539139	14.539113	14.539175	14.538659	14.538638	14.538691	14.538962	14.538865	14.539086	14.539610
		6 × 18	10.864020	10.863997	10.864046	10.863678	10.863669	10.863683	10.863938	10.863910	10.863957	10.864263
		8 × 20	8.923226	8.923200	8.923260	8.922852	8.922842	8.922869	8.923177	8.923139	8.923212	8.923461

		10 × 22	7.089532	7.089515	7.089549	7.089083	7.089057	7.089115	7.089469	7.089432	7.089522	7.089783
		12 × 24	5.258545	5.258532	5.258569	5.258086	5.258050	5.258109	5.258363	5.258329	5.258411	5.258722
		14 × 26	3.571711	3.571670	3.571744	3.571259	3.571233	3.571280	3.571262	3.571211	3.571309	3.571697
		16 × 28	1.810042	1.810016	1.810069	1.809602	1.809558	1.809638	1.809520	1.809469	1.809550	1.809801
Spindle	Uniform	4 × 16	20.303136	20.303094	20.303178	20.302735	20.302718	20.302752	20.303027	20.302953	20.303078	20.303638
		6 × 18	15.444257	15.444216	15.444286	15.443798	15.443780	15.443814	15.444149	15.444115	15.444184	15.444657
		8 × 20	11.904988	11.904958	11.905007	11.904472	11.904460	11.904493	11.904913	11.904860	11.904950	11.905378
		10 × 22	8.973139	8.973107	8.973164	8.972601	8.972581	8.972631	8.973074	8.973038	8.973123	8.973447
		12 × 24	6.464246	6.464227	6.464286	6.463724	6.463712	6.463750	6.464101	6.464058	6.464137	6.464457
		14 × 26	4.250832	4.250798	4.250867	4.250345	4.250309	4.250403	4.250403	4.250300	4.250450	4.250889
		16 × 28	2.351893	2.351862	2.351919	2.351417	2.351372	2.351452	2.351374	2.351344	2.351421	2.351708
	Chord Length	4 × 16	19.169619	19.169556	19.169668	19.169169	19.169142	19.169189	19.169490	19.169452	19.169583	19.170128
		6 × 18	14.050294	14.050276	14.050318	14.049869	14.049854	14.049886	14.050193	14.050109	14.050259	14.050667
		8 × 20	10.725351	10.725334	10.725371	10.724896	10.724868	10.724912	10.725263	10.725226	10.725327	10.725676
		10 × 22	8.003482	8.003458	8.003506	8.002999	8.002984	8.003026	8.003394	8.003370	8.003415	8.003751
		12 × 24	5.649720	5.649682	5.649749	5.649234	5.649207	5.649261	5.649563	5.649537	5.649602	5.649897
		14 × 26	3.649062	3.649018	3.649105	3.648610	3.648560	3.648639	3.648600	3.648548	3.648637	3.649081
		16 × 28	1.955129	1.955089	1.955163	1.954713	1.954662	1.954740	1.954630	1.954605	1.954658	1.954901
	Centripetal	4 × 16	19.023646	19.023594	19.023667	19.023261	19.023236	19.023291	19.023539	19.023491	19.023585	19.024116
		6 × 18	14.433147	14.433109	14.433169	14.432691	14.432664	14.432709	14.433038	14.432965	14.433103	14.433540
		8 × 20	11.026715	11.026675	11.026744	11.026202	11.026185	11.026218	11.026613	11.026575	11.026642	11.027065
		10 × 22	8.253758	8.253738	8.253793	8.253257	8.253237	8.253272	8.253676	8.253633	8.253731	8.254017
		12 × 24	5.869515	5.869495	5.869538	5.869014	5.868990	5.869043	5.869352	5.869303	5.869416	5.869683
		14 × 26	3.776505	3.776475	3.776523	3.776039	3.776007	3.776080	3.776049	3.776012	3.776093	3.776540
		16 × 28	2.027849	2.027805	2.027900	2.027404	2.027384	2.027430	2.027321	2.027304	2.027348	2.027645
	Exponential	4 × 16	18.872991	18.872946	18.873022	18.872568	18.872540	18.872602	18.872859	18.872796	18.872953	18.873429
		6 × 18	14.087564	14.087550	14.087579	14.087121	14.087107	14.087139	14.087464	14.087445	14.087487	14.087943
		8 × 20	10.743640	10.743623	10.743659	10.743172	10.743143	10.743193	10.743554	10.743521	10.743599	10.743960
		10 × 22	8.024542	8.024519	8.024563	8.024060	8.024026	8.024085	8.024476	8.024448	8.024548	8.024803
		12 × 24	5.682923	5.682906	5.682942	5.682438	5.682424	5.682460	5.682757	5.682713	5.682817	5.683107
		14 × 26	3.665795	3.665757	3.665863	3.665348	3.665304	3.665380	3.665350	3.665321	3.665394	3.665819

		16 × 28	1.965493	1.965444	1.965529	1.965071	1.965043	1.965097	1.964985	1.964954	1.965023	1.965280
Oiltank	Uniform	4 × 16	29.149124	29.149090	29.149150	29.148657	29.148621	29.148694	29.148979	29.148903	29.149062	29.149654
		6 × 18	19.829457	19.829426	19.829490	19.829005	19.828999	19.829016	19.829354	19.829305	19.829398	19.829836
		8 × 20	14.743819	14.743786	14.743843	14.743318	14.743299	14.743349	14.743733	14.743690	14.743765	14.744166
		10 × 22	10.947890	10.947874	10.947919	10.947344	10.947324	10.947363	10.947794	10.947749	10.947851	10.948206
		12 × 24	7.936495	7.936477	7.936522	7.935954	7.935929	7.935981	7.936368	7.936346	7.936418	7.936696
		14 × 26	5.592036	5.592016	5.592063	5.591531	5.591505	5.591575	5.591636	5.591579	5.591711	5.592089
		16 × 28	3.190484	3.190456	3.190528	3.189989	3.189948	3.190042	3.189967	3.189902	3.190000	3.190289
	Chord Length	4 × 16	26.909319	26.909278	26.909353	26.908865	26.908852	26.908885	26.909177	26.909132	26.909225	26.909804
		6 × 18	17.810741	17.810718	17.810769	17.810338	17.810321	17.810349	17.810626	17.810584	17.810675	17.811067
		8 × 20	13.121151	13.121134	13.121164	13.120788	13.120771	13.120799	13.121097	13.121069	13.121111	13.121388
		10 × 22	9.595158	9.595136	9.595193	9.594721	9.594709	9.594730	9.595106	9.595056	9.595150	9.595360
		12 × 24	6.754789	6.754765	6.754809	6.754323	6.754281	6.754354	6.754647	6.754595	6.754685	6.754982
		14 × 26	4.535308	4.535280	4.535340	4.534833	4.534812	4.534870	4.534890	4.534848	4.534929	4.535360
		16 × 28	2.367793	2.367772	2.367826	2.367375	2.367358	2.367401	2.367273	2.367231	2.367302	2.367612
	Centripetal	4 × 16	26.932710	26.932658	26.932743	26.932306	26.932292	26.932333	26.932575	26.932481	26.932622	26.933113
		6 × 18	18.275082	18.275065	18.275111	18.274678	18.274658	18.274693	18.274994	18.274973	18.275021	18.275420
		8 × 20	13.397975	13.397952	13.397995	13.397558	13.397546	13.397578	13.397899	13.397865	13.397928	13.398294
		10 × 22	9.848638	9.848627	9.848658	9.848166	9.848150	9.848187	9.848576	9.848549	9.848615	9.848888
		12 × 24	6.999554	6.999526	6.999569	6.999056	6.999029	6.999092	6.999420	6.999360	6.999472	6.999715
		14 × 26	4.765031	4.765000	4.765058	4.764570	4.764543	4.764603	4.764618	4.764555	4.764688	4.765088
		16 × 28	2.555902	2.555882	2.555923	2.555478	2.555446	2.555507	2.555373	2.555330	2.555395	2.555727
	Exponential	4 × 16	26.566191	26.566171	26.566219	26.565762	26.565736	26.565779	26.566058	26.566019	26.566140	26.566622
		6 × 18	17.828909	17.828896	17.828939	17.828509	17.828490	17.828522	17.828826	17.828794	17.828871	17.829253
		8 × 20	13.070579	13.070560	13.070599	13.070196	13.070193	13.070202	13.070517	13.070473	13.070554	13.070835
		10 × 22	9.580123	9.580110	9.580136	9.579678	9.579662	9.579691	9.580075	9.580027	9.580105	9.580364
		12 × 24	6.759860	6.759844	6.759891	6.759369	6.759343	6.759387	6.759735	6.759701	6.759773	6.760065
		14 × 26	4.561900	4.561873	4.561925	4.561434	4.561403	4.561463	4.561488	4.561439	4.561547	4.561951
		16 × 28	2.400860	2.400798	2.400898	2.400406	2.400350	2.400429	2.400311	2.400273	2.400352	2.400655
Talus Bone	Uniform	4 × 16	28.027069	28.027050	28.027104	28.026621	28.026596	28.026642	28.026939	28.026880	28.027048	28.027628
		6 × 18	18.233947	18.233928	18.233973	18.233578	18.233558	18.233596	18.233870	18.233797	18.233904	18.234296

		8 × 20	14.132581	14.132557	14.132601	14.132142	14.132134	14.132154	14.132501	14.132458	14.132541	14.132923
		10 × 22	10.994991	10.994969	10.995010	10.994449	10.994430	10.994472	10.994916	10.994883	10.994947	10.995319
		12 × 24	7.964153	7.964130	7.964171	7.963597	7.963576	7.963635	7.964043	7.963946	7.964073	7.964382
		14 × 26	5.262009	5.261993	5.262035	5.261510	5.261474	5.261548	5.261624	5.261570	5.261687	5.262074
		16 × 28	3.080541	3.080520	3.080573	3.080065	3.080038	3.080105	3.080024	3.079982	3.080099	3.080396
	Chord Length	4 × 16	26.198166	26.198138	26.198206	26.197742	26.197719	26.197758	26.198010	26.197952	26.198092	26.198621
		6 × 18	18.895572	18.895543	18.895590	18.895120	18.895103	18.895143	18.895457	18.895392	18.895525	18.895910
		8 × 20	14.782232	14.782223	14.782255	14.781795	14.781776	14.781811	14.782152	14.782086	14.782193	14.782553
		10 × 22	10.601724	10.601709	10.601748	10.601234	10.601223	10.601246	10.601669	10.601648	10.601699	10.601987
		12 × 24	6.995798	6.995775	6.995834	6.995299	6.995277	6.995316	6.995696	6.995651	6.995736	6.996009
		14 × 26	4.346560	4.346527	4.346609	4.346098	4.346057	4.346140	4.346109	4.346056	4.346162	4.346587
		16 × 28	2.253989	2.253923	2.254027	2.253542	2.253510	2.253588	2.253469	2.253443	2.253504	2.253805
	Centripetal	4 × 16	25.535169	25.535117	25.535207	25.534743	25.534726	25.534768	25.535034	25.534970	25.535110	25.535604
		6 × 18	17.760337	17.760296	17.760373	17.759951	17.759939	17.759964	17.760264	17.760226	17.760299	17.760659
		8 × 20	13.874937	13.874906	13.874960	13.874509	13.874487	13.874544	13.874855	13.874804	13.874882	13.875225
		10 × 22	10.358636	10.358621	10.358661	10.358123	10.358091	10.358140	10.358560	10.358535	10.358586	10.358928
		12 × 24	7.170420	7.170398	7.170432	7.169889	7.169855	7.169922	7.170298	7.170249	7.170368	7.170618
		14 × 26	4.535505	4.535475	4.535545	4.535041	4.535022	4.535057	4.535061	4.535003	4.535085	4.535553
		16 × 28	2.450582	2.450553	2.450603	2.450147	2.450128	2.450171	2.450059	2.450016	2.450094	2.450417
	Exponential	4 × 16	25.449204	25.449181	25.449221	25.448796	25.448784	25.448813	25.449090	25.449007	25.449164	25.449586
		6 × 18	18.267931	18.267906	18.267961	18.267524	18.267490	18.267541	18.267856	18.267825	18.267883	18.268279
		8 × 20	14.299876	14.299863	14.299887	14.299442	14.299419	14.299478	14.299810	14.299759	14.299852	14.300186
		10 × 22	10.416240	10.416227	10.416264	10.415749	10.415727	10.415763	10.416172	10.416121	10.416210	10.416476
		12 × 24	6.993393	6.993369	6.993409	6.992867	6.992832	6.992897	6.993261	6.993226	6.993291	6.993589
		14 × 26	4.367806	4.367774	4.367840	4.367351	4.367302	4.367381	4.367380	4.367354	4.367434	4.367835
		16 × 28	2.295647	2.295613	2.295674	2.295196	2.295175	2.295247	2.295112	2.295080	2.295149	2.295462

APPENDIX M: CPU TIME OF VARIOUS OPTIMISATION TECHNIQUES FOR THE CN WITH THE SAME WIDTH AND LENGTH

Data	Parameterisation Method	CN	GA			DE			PSO		
			AVG	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX
Cube	Uniform	6 × 6	0.388101	0.371044	0.406877	9.784655	9.500566	10.188149	13.412222	12.878369	14.075492
		8 × 8	0.456279	0.442464	0.515516	13.370592	12.561866	15.425672	20.213740	19.911071	21.026734
		10 × 10	0.551661	0.533666	0.611869	18.032840	17.221446	19.057133	29.153866	28.656376	29.684410
		12 × 12	0.665674	0.650137	0.709199	23.796476	23.414966	24.242876	39.804323	39.059117	40.696977
		14 × 14	0.804641	0.789030	0.829599	30.745474	29.882509	31.664061	52.813786	52.200012	53.957296
		16 × 16	0.955298	0.935253	1.004452	40.282178	38.497361	42.327701	68.724385	67.832329	69.812903
		18 × 18	1.150151	1.120507	1.179987	48.646868	46.647540	50.141018	86.705628	85.693464	89.265211
	Chord Length	6 × 6	0.385994	0.376709	0.408491	10.583109	10.117027	10.888563	13.045490	12.793874	13.364637
		8 × 8	0.457684	0.442497	0.552923	14.206907	13.519582	15.094221	19.692966	19.392474	20.063678
		10 × 10	0.596087	0.554687	0.639495	19.231021	18.882349	19.664623	28.593408	28.334574	28.927130
		12 × 12	0.662576	0.638658	0.716030	25.401821	24.957085	26.199232	39.661321	39.156056	40.193459
		14 × 14	0.827267	0.797521	0.888658	32.193756	30.294066	33.778259	53.466995	52.115401	56.176835
		16 × 16	0.953373	0.924163	1.004376	39.197594	37.914932	40.954557	66.720190	65.715962	68.126311
		18 × 18	1.165943	1.131108	1.199119	48.180109	47.169969	49.553052	82.798292	82.525231	83.094313
	Centripetal	6 × 6	0.393043	0.372152	0.414902	10.051726	9.725838	10.620063	12.436757	12.362741	12.498526
		8 × 8	0.460381	0.447248	0.491504	13.601745	13.138909	14.231219	18.801053	18.761502	18.915378
		10 × 10	0.544158	0.532821	0.568167	17.918663	17.460769	18.910314	27.390550	27.311756	27.445044
		12 × 12	0.666268	0.644102	0.695273	24.226109	23.015319	25.931873	37.997220	37.918152	38.149257
		14 × 14	0.799284	0.784212	0.819481	31.696470	30.023788	34.281164	50.731034	50.531158	50.898128
		16 × 16	0.945357	0.920410	0.979257	38.559256	37.695267	39.792695	65.700416	65.569848	65.831555
		18 × 18	1.127152	1.093820	1.206800	48.310609	46.562376	50.343837	82.710592	82.553832	83.008144
	Exponential	6 × 6	0.387000	0.371558	0.417289	10.018397	9.774976	11.158660	12.401173	12.322330	12.514337
		8 × 8	0.453686	0.440665	0.487918	13.671929	13.250608	14.045888	18.761589	18.711072	18.834599
		10 × 10	0.538981	0.526420	0.565994	18.250996	17.199006	20.187655	27.394120	27.295364	27.590390

		12 × 12	0.652221	0.635979	0.710220	23.560597	22.648198	24.665452	37.975608	37.874426	38.076041
		14 × 14	0.797540	0.781307	0.830393	30.041849	29.373025	30.943339	50.743191	50.663095	50.826910
		16 × 16	0.959945	0.918628	1.049326	38.449124	36.838217	39.618502	65.699977	65.553912	65.902556
		18 × 18	1.161302	1.110376	1.271916	48.021365	46.902010	49.413085	82.633427	82.463817	82.816525
Sphere	Uniform	6 × 6	0.399241	0.371659	0.437014	9.837910	9.632968	10.080606	12.375842	12.324711	12.511008
		8 × 8	0.456525	0.443967	0.489051	13.242299	13.095354	13.568987	18.823946	18.747302	18.930457
		10 × 10	0.566311	0.535029	0.606290	18.079608	17.626956	18.519668	27.504707	27.329324	28.316209
		12 × 12	0.674988	0.646524	0.754229	24.082228	23.883318	24.240623	38.012077	37.887302	38.204602
		14 × 14	0.805602	0.789703	0.836870	30.731407	30.323911	31.352745	50.821895	50.680041	50.940250
		16 × 16	0.954310	0.927872	1.067282	39.204757	38.286835	40.277052	65.762586	65.643237	65.912855
		18 × 18	1.139046	1.099528	1.220061	47.478593	47.218318	47.792107	82.644310	82.422966	82.792910
	Chord Length	6 × 6	0.382363	0.371708	0.400797	9.779920	9.621759	10.211354	12.402161	12.358626	12.462243
		8 × 8	0.456673	0.441530	0.487623	13.280685	13.038715	13.546661	18.813936	18.752790	18.865985
		10 × 10	0.571990	0.538487	0.682671	18.050691	17.345046	19.400738	27.434574	27.378796	27.506148
		12 × 12	0.682423	0.650598	0.738676	23.949270	23.237717	25.120936	37.998155	37.908061	38.197108
		14 × 14	0.806284	0.771440	0.832470	30.797081	29.545134	31.892253	50.815077	50.658859	50.983579
		16 × 16	0.954144	0.927805	0.995089	38.331058	37.765191	39.216843	65.681214	65.543598	65.798160
		18 × 18	1.163995	1.121544	1.245875	47.180600	46.251938	48.552511	82.656766	82.503648	82.739091
	Centripetal	6 × 6	0.390231	0.372077	0.454510	9.435519	9.337962	9.601770	12.404148	12.356542	12.514893
		8 × 8	0.451730	0.439732	0.508523	12.799772	12.747817	12.887440	18.854089	18.794894	18.984032
		10 × 10	0.556134	0.535137	0.588780	17.427896	17.252213	17.736386	27.451083	27.357400	27.520412
		12 × 12	0.661353	0.641322	0.710930	22.747423	22.616844	22.833141	38.021734	37.918866	38.109729
		14 × 14	0.787827	0.767485	0.819854	29.428674	29.289432	29.549356	50.896017	50.642499	51.172291
		16 × 16	0.946622	0.923220	1.002346	36.976405	36.783689	37.181790	65.790389	65.598421	66.202952
		18 × 18	1.122970	1.097627	1.168021	48.909632	46.115113	51.908745	82.709865	82.600571	82.808898
	Exponential	6 × 6	0.377316	0.370529	0.387918	10.081694	9.563371	10.778017	12.413235	12.327561	12.511627
		8 × 8	0.450349	0.439457	0.487766	13.356826	13.063144	13.850290	18.850946	18.750676	18.926576
		10 × 10	0.545249	0.531150	0.573502	18.170022	17.991084	18.601480	27.442088	27.388248	27.502685
		12 × 12	0.657924	0.639698	0.684147	23.717066	23.039606	24.671102	38.051404	37.948127	38.141998
		14 × 14	0.790362	0.773243	0.808492	31.880794	30.916892	33.211994	50.839484	50.715683	50.919316
		16 × 16	0.940916	0.923455	0.999397	39.129381	37.471900	42.077283	65.789201	65.628422	65.970328

		18 × 18	1.118685	1.096539	1.158663	48.999313	46.574008	53.640809	82.797716	82.505856	83.243306
Spindle	Uniform	6 × 6	0.379258	0.369114	0.409242	10.111463	10.012816	10.213570	13.231119	12.717621	13.602176
		8 × 8	0.445928	0.440933	0.461389	13.755714	13.602216	14.047185	20.001718	19.738210	20.283845
		10 × 10	0.544139	0.529514	0.570193	18.746636	18.374219	19.350392	29.084728	28.762763	29.327696
		12 × 12	0.650751	0.638101	0.683812	25.210712	24.740848	25.702103	40.349682	39.717528	41.174531
		14 × 14	0.789964	0.766224	0.860277	32.000360	31.388828	32.671752	53.838631	53.332424	54.313190
		16 × 16	0.941639	0.927372	0.979413	39.927558	39.620927	40.778361	69.667273	68.928420	70.164931
		18 × 18	1.128398	1.098144	1.214000	46.812299	45.688033	48.824579	87.974729	87.065816	88.961361
	Chord Length	6 × 6	0.377691	0.368230	0.402624	9.611294	9.370712	9.818911	13.201724	12.897148	13.437537
		8 × 8	0.445688	0.437332	0.470018	12.918674	12.734442	13.183087	20.117308	19.754905	20.668642
		10 × 10	0.539381	0.531177	0.570812	17.473189	17.194199	17.903944	29.073715	28.709815	29.453660
		12 × 12	0.652302	0.643810	0.683973	23.285116	22.934653	23.499799	40.364613	40.030871	40.629624
		14 × 14	0.788496	0.767081	0.818211	30.134502	29.447079	30.912911	53.894736	53.438681	54.371645
		16 × 16	0.946962	0.917081	1.013430	37.794691	36.893959	38.950116	69.917374	68.916749	70.606390
		18 × 18	1.115344	1.089350	1.139273	47.000401	46.447177	47.568784	87.509664	86.521463	88.361061
	Centripetal	6 × 6	0.373296	0.369889	0.380376	9.589677	9.297864	9.899340	13.190936	12.904458	13.623859
		8 × 8	0.454025	0.436501	0.515193	13.223757	12.895499	13.918906	20.076197	19.617591	21.601034
		10 × 10	0.548788	0.534351	0.624160	17.626077	17.351623	17.959225	29.567346	27.872062	31.512917
		12 × 12	0.657203	0.641013	0.687204	23.320100	23.119570	23.672435	39.693426	38.565783	40.739972
		14 × 14	0.791055	0.767819	0.813855	29.835309	29.144067	31.011191	53.219940	52.212673	56.104017
		16 × 16	0.952834	0.937404	1.012233	38.026783	37.176081	38.962237	69.889700	66.312771	75.851413
		18 × 18	1.131629	1.086124	1.171234	47.074720	46.740787	47.993800	87.529694	84.112711	89.725112
	Exponential	6 × 6	0.380723	0.370353	0.423879	9.752976	9.487419	10.137441	12.997209	12.461254	13.398008
		8 × 8	0.445054	0.437207	0.474083	13.189576	12.859570	13.560253	19.784321	18.915567	20.383535
		10 × 10	0.541281	0.530026	0.592089	17.826801	17.592418	18.098436	28.859579	27.688770	29.801726
		12 × 12	0.647844	0.639278	0.656232	23.483841	22.983889	23.900709	38.496548	38.149737	39.459727
		14 × 14	0.791579	0.780431	0.828424	30.389172	29.772166	31.002731	51.895685	50.972045	56.037723
		16 × 16	0.940724	0.917528	0.964503	38.155543	37.749463	39.023225	69.715254	65.884600	72.678988
		18 × 18	1.112614	1.095986	1.158213	47.529165	46.821345	51.176938	86.719616	83.589167	88.901618
Oiltank	Uniform	6 × 6	0.383457	0.368813	0.437762	9.657258	9.307733	10.095216	13.457315	12.598560	14.486503
		8 × 8	0.448783	0.437523	0.465374	13.241409	12.899018	13.472907	20.138849	19.445407	21.304659

		10 × 10	0.541388	0.531265	0.575395	17.936720	17.296382	18.456767	29.881269	29.134411	30.853477
		12 × 12	0.650556	0.636798	0.690738	23.293606	22.961287	23.814253	39.797772	38.113750	41.605893
		14 × 14	0.798580	0.777113	0.910539	30.396819	29.893125	31.085297	53.769599	51.123556	57.005131
		16 × 16	0.927812	0.913791	0.941325	38.004596	37.318261	38.536710	71.022142	69.472487	76.433765
		18 × 18	1.120774	1.092686	1.172348	47.090571	46.303718	48.317754	83.675955	82.867531	87.849739
		6 × 6	0.382264	0.369978	0.417084	9.540042	9.355105	9.724281	12.513908	12.408529	12.619443
		8 × 8	0.445035	0.435606	0.457500	13.014391	12.564725	14.222392	18.929773	18.844583	19.035367
		10 × 10	0.542582	0.528731	0.584714	17.286791	17.076009	17.529636	27.639102	27.452717	28.035353
	Chord Length	12 × 12	0.659364	0.639205	0.701657	23.552750	22.632497	24.998487	38.312022	38.137983	38.505550
		14 × 14	0.789343	0.773945	0.817781	29.181132	29.080497	29.240823	51.026243	50.823603	51.201508
		16 × 16	0.942646	0.925224	0.968144	36.806673	36.707924	36.940324	66.079259	65.805412	66.418680
		18 × 18	1.130738	1.097409	1.179630	45.556324	45.394947	45.819603	83.165122	82.804180	83.448326
		6 × 6	0.378871	0.371697	0.391895	9.370537	9.288292	9.513158	12.977551	12.445149	13.447357
		8 × 8	0.480010	0.447468	0.599101	12.718234	12.640397	12.819807	20.129622	20.034743	20.200431
		10 × 10	0.548274	0.534557	0.597429	17.180196	17.108562	17.330511	27.962233	27.412089	29.336449
	Centripetal	12 × 12	0.650493	0.644022	0.660896	22.647536	22.525543	22.822693	38.204772	38.070603	38.381336
		14 × 14	0.798193	0.768713	0.901073	29.234019	29.128744	29.333935	51.061032	50.864291	51.477944
		16 × 16	0.947105	0.922093	1.022737	36.869832	36.640663	37.317778	65.936009	65.729993	66.099511
		18 × 18	1.123499	1.090928	1.160942	45.514452	45.339071	45.673303	82.981224	82.854433	83.137481
		6 × 6	0.388870	0.371023	0.422425	9.345657	9.288890	9.439652	12.483001	12.391823	12.600919
		8 × 8	0.441662	0.437852	0.445717	12.866340	12.668555	13.025436	18.934489	18.800341	19.107814
		10 × 10	0.541167	0.528992	0.586010	17.284900	17.094770	17.597833	27.563800	27.485670	27.646007
	Exponential	12 × 12	0.654709	0.641236	0.686330	22.654800	22.521348	22.780945	38.250206	38.101173	38.540756
		14 × 14	0.791874	0.772621	0.823968	29.291644	29.175299	29.378927	51.051257	50.828553	51.215794
		16 × 16	0.933139	0.917837	0.957118	36.779961	36.679767	36.900729	66.025633	65.897802	66.243874
		18 × 18	1.137284	1.104066	1.189842	45.530106	45.426719	45.628240	83.007036	82.920627	83.142998
		6 × 6	0.382757	0.371273	0.427274	10.980600	10.193953	12.723333	12.398625	12.330603	12.466558
		8 × 8	0.452801	0.438280	0.470623	13.962384	13.379414	15.199974	18.818016	18.761294	18.890138
Talus Bone	Uniform	10 × 10	0.552571	0.535885	0.576235	18.463178	18.360257	18.585206	27.373460	27.294143	27.467798
		12 × 12	0.653408	0.640513	0.667463	24.649889	23.690236	26.739567	38.042318	37.850333	38.213254
		14 × 14	0.786714	0.776167	0.804259	31.077642	30.561439	32.915513	50.833940	50.720695	50.998356

		16 × 16	0.955642	0.928613	1.031993	40.273041	38.756273	42.288990	65.697530	65.558422	65.867277
		18 × 18	1.122556	1.091483	1.177887	47.584336	45.760376	51.411048	82.715262	82.591350	82.876159
	Chord Length	6 × 6	0.377258	0.370326	0.400317	9.613801	9.397844	9.977116	12.466967	12.349091	12.900097
		8 × 8	0.456755	0.440500	0.504955	12.944817	12.771450	13.338056	18.845375	18.753519	19.142979
		10 × 10	0.548374	0.530609	0.574447	17.559637	17.155349	17.956261	27.402057	27.289941	27.576436
		12 × 12	0.654465	0.640790	0.680016	22.934500	22.677445	23.160848	38.003041	37.855539	38.159569
		14 × 14	0.790604	0.773964	0.822196	29.657888	29.295887	30.054841	50.776327	50.638070	51.002820
		16 × 16	0.955194	0.921739	1.031204	38.617635	37.491469	39.325440	65.692400	65.601154	65.824135
		18 × 18	1.169423	1.123157	1.392300	47.962100	46.596477	51.609079	82.650353	82.464608	82.897220
	Centripetal	6 × 6	0.390660	0.375472	0.430215	9.787143	9.592203	10.053250	12.441628	12.365017	12.564697
		8 × 8	0.447817	0.443023	0.457022	13.082653	12.989034	13.276949	18.821418	18.757341	18.923995
		10 × 10	0.553251	0.531909	0.609344	17.957685	17.362027	18.377387	27.415558	27.337321	27.484031
		12 × 12	0.680065	0.648095	0.826572	23.277543	22.999903	23.618524	40.136898	37.946488	41.851388
		14 × 14	0.787972	0.774992	0.813692	30.143077	29.837447	30.443598	56.389968	53.733926	59.938439
		16 × 16	0.949749	0.920407	1.031577	37.465811	36.763701	38.004666	69.661093	68.624276	71.684634
		18 × 18	1.122199	1.092062	1.193610	47.700055	46.854601	48.786031	88.997905	87.347960	91.912181
	Exponential	6 × 6	0.384612	0.372054	0.421802	9.787312	9.478289	10.115349	13.509589	13.242207	14.144161
		8 × 8	0.458964	0.443112	0.499785	12.915920	12.691664	13.085092	20.245841	19.841358	21.232193
		10 × 10	0.538080	0.529707	0.554080	17.614419	17.151033	18.552175	29.479141	28.502910	31.689328
		12 × 12	0.651129	0.638270	0.676715	23.509673	22.724579	25.084944	40.131778	39.508256	40.891120
		14 × 14	0.794032	0.772861	0.867894	31.334783	29.809458	33.759972	54.142484	53.624439	54.704796
		16 × 16	0.941438	0.914567	0.979636	40.274351	39.633289	42.321743	69.699028	69.094768	70.258235
		18 × 18	1.126533	1.088050	1.180825	49.573273	48.886464	50.902446	87.906294	87.431624	88.739037

APPENDIX N: CPU TIME OF VARIOUS OPTIMISATION TECHNIQUES FOR CN WITH DIFFERENT WIDTH AND LENGTH

Data	Parameterisation Method	CN	GA			DE			PSO		
			AVG	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX
Cube	Uniform	4 × 16	0.591043	0.563247	0.627139	15.651911	15.146953	15.900295	20.341968	19.905598	21.612305
		6 × 18	0.739647	0.684899	0.858069	22.608026	21.730103	23.581059	30.828541	30.543928	31.423699
		8 × 20	0.848386	0.816134	0.915082	29.809014	28.726197	30.491131	43.208668	42.934889	43.319473
		10 × 22	1.004346	0.966378	1.118221	38.118036	36.450864	41.140287	58.288716	58.178429	58.390016
		12 × 24	1.150396	1.131633	1.172620	48.299701	47.225966	51.553303	74.955542	74.589211	75.328012
		14 × 26	1.403912	1.341344	1.543191	58.074676	57.869720	58.338188	93.793985	93.232008	94.159070
		16 × 28	1.595199	1.549344	1.693442	69.684311	67.611260	75.348082	114.815637	114.708564	114.972957
	Chord Length	4 × 16	0.586852	0.562976	0.626034	15.906646	15.515463	16.248191	19.997045	19.921802	20.055761
		6 × 18	0.692003	0.675845	0.719280	22.366626	21.844897	23.193834	30.606698	30.484934	30.664336
		8 × 20	0.820347	0.802698	0.854111	30.312199	29.319936	31.614619	43.145565	42.963480	43.247215
		10 × 22	1.003053	0.984724	1.031228	38.185215	37.181548	39.149391	58.132617	58.037277	58.196552
		12 × 24	1.177196	1.144881	1.212242	48.080122	45.103756	51.840441	74.772719	74.318658	74.922771
		14 × 26	1.395062	1.346595	1.471354	58.314024	57.530856	61.822197	94.024402	93.775893	95.038272
		16 × 28	1.595161	1.541156	1.662904	69.575520	68.149731	74.097105	114.753861	114.618758	115.062855
	Centripetal	4 × 16	0.596135	0.572755	0.636654	16.050339	15.069983	17.308587	20.020958	19.973979	20.092280
		6 × 18	0.719649	0.677840	0.765636	22.548029	21.146208	23.662818	30.616195	30.503794	30.796898
		8 × 20	0.858035	0.813498	0.968608	29.368038	27.294956	31.674424	43.178927	42.993202	43.251931
		10 × 22	1.039117	0.977386	1.163726	36.147582	35.449274	37.757092	58.115463	57.810826	58.287148
		12 × 24	1.268411	1.136404	1.516420	48.439792	46.295820	53.667798	74.944777	74.505619	75.149217
		14 × 26	1.453443	1.387438	1.594899	57.788239	55.575871	61.655967	93.865229	93.305670	94.051859
		16 × 28	1.609991	1.577280	1.700811	71.637581	66.473938	80.009985	114.775157	114.595971	115.063068
	Exponential	4 × 16	0.591156	0.572007	0.639764	16.742784	16.104852	17.349766	20.003828	19.945071	20.075743
		6 × 18	0.693831	0.682983	0.718253	23.380553	22.333529	25.066284	30.628056	30.530850	30.723241
		8 × 20	0.835661	0.823974	0.858746	32.631433	29.892702	36.842591	43.276836	43.115338	43.577870

		10 × 22	0.997230	0.969420	1.035753	40.037628	38.289862	43.464525	58.151406	58.083568	58.220393
		12 × 24	1.168312	1.135082	1.207637	50.138069	48.369598	54.072068	74.843972	74.402293	75.047434
		14 × 26	1.395134	1.345207	1.566686	61.768943	58.251633	68.591728	93.909152	93.297875	94.306267
		16 × 28	1.697715	1.608249	1.867496	74.347790	70.220396	80.399789	114.565720	113.902655	114.850199
Sphere	Uniform	4 × 16	0.612037	0.574766	0.654357	17.356065	15.547736	19.395754	20.112414	20.049629	20.342152
		6 × 18	0.730030	0.700064	0.756300	23.867446	23.611099	24.107781	30.746902	30.697877	30.781601
		8 × 20	0.865364	0.838576	0.912586	28.243108	28.025792	28.620513	43.301569	43.234621	43.403717
		10 × 22	1.109512	0.999368	1.287794	36.160680	35.969323	36.317603	58.239345	58.160402	58.317952
		12 × 24	1.185662	1.143028	1.261248	45.153967	44.926380	45.462773	75.009262	74.568783	75.141129
		14 × 26	1.399523	1.332207	1.463546	55.379885	55.121846	55.561048	93.907316	93.304903	94.281968
		16 × 28	1.599161	1.526571	1.700846	66.611092	66.408165	66.840593	114.751095	113.870109	115.727930
	Chord Length	4 × 16	0.579279	0.563624	0.606861	15.503513	15.364843	15.637961	20.041783	19.993896	20.080735
		6 × 18	0.701360	0.684760	0.771361	21.325386	21.194819	21.432662	30.722097	30.619289	30.761191
		8 × 20	0.845528	0.813179	0.881447	28.097600	27.803029	28.326106	43.247603	42.883567	43.390083
		10 × 22	0.979687	0.963833	1.024433	36.259180	35.968497	36.945132	58.258856	58.175954	58.378689
		12 × 24	1.184211	1.143290	1.238936	48.288054	47.637118	48.597335	75.022456	74.953341	75.142240
		14 × 26	1.368187	1.334417	1.436768	59.702928	59.327667	60.006821	93.928757	93.508758	94.374403
		16 × 28	1.575794	1.550982	1.609686	69.348078	66.150620	71.678535	114.656833	113.952084	114.893018
	Centripetal	4 × 16	0.588594	0.562068	0.629174	15.462552	15.302797	15.679351	20.430530	20.061134	21.213174
		6 × 18	0.696334	0.682528	0.723860	21.297464	21.187621	21.419268	32.659442	31.606808	33.578256
		8 × 20	0.847657	0.822385	0.878182	27.880329	27.704803	28.068755	45.934237	43.341133	50.922151
		10 × 22	0.985310	0.968863	1.006166	36.006901	35.808787	36.211474	60.345490	58.111254	63.103667
		12 × 24	1.162694	1.139436	1.208534	44.894281	44.682565	45.114358	78.436884	75.924419	81.880155
		14 × 26	1.364125	1.328295	1.423229	55.179087	54.992523	55.472841	98.440177	94.363129	100.449915
		16 × 28	1.564942	1.541750	1.605204	66.183926	65.934240	66.381341	122.244476	118.617185	125.597890
	Exponential	4 × 16	0.578964	0.562042	0.620798	15.381274	15.233415	15.498959	21.589128	21.142736	22.731525
		6 × 18	0.698661	0.685717	0.731004	21.256341	21.106464	21.387941	33.251748	32.321225	34.333319
		8 × 20	0.831722	0.803529	0.877351	27.983573	27.831595	28.281182	47.822016	47.337418	48.647393
		10 × 22	0.985188	0.967767	1.005582	35.983820	35.862522	36.173330	62.463962	61.031141	63.797108
		12 × 24	1.152317	1.127692	1.173363	44.891201	44.609671	45.061581	79.550363	78.911531	80.015241
		14 × 26	1.362001	1.338063	1.383127	55.118841	55.045249	55.194527	99.111574	98.440600	99.659962

		16 × 28	1.574906	1.537212	1.632797	66.317884	66.136395	66.641761	121.395298	120.363804	122.418268
Spindle	Uniform	4 × 16	0.586838	0.561059	0.729726	15.320410	15.146636	15.492929	21.247223	21.011839	21.667630
		6 × 18	0.694750	0.667737	0.750800	21.136968	21.044857	21.260252	31.617925	31.001073	32.661324
		8 × 20	0.836279	0.801783	0.899233	27.954431	27.784321	28.123190	43.842669	43.611679	44.131528
		10 × 22	0.976094	0.955801	1.004767	35.983486	35.886674	36.078018	58.836626	58.724988	58.982474
		12 × 24	1.148239	1.117850	1.187160	44.802148	44.524665	44.902920	75.707250	75.484728	75.887721
		14 × 26	1.393541	1.329868	1.541141	54.884767	54.682529	55.077740	97.635846	94.666879	103.187729
		16 × 28	1.579839	1.541370	1.632134	65.915996	65.706715	66.116695	123.497214	119.669187	129.900431
	Chord Length	4 × 16	0.567494	0.553567	0.591081	15.364037	15.269317	15.422795	21.285791	21.027713	21.552254
		6 × 18	0.700759	0.677480	0.744270	21.171089	21.063722	21.375393	33.195891	31.907020	34.639082
		8 × 20	0.828756	0.801222	0.889053	27.924374	27.755257	28.013886	46.052688	45.578826	47.252158
		10 × 22	0.980832	0.964903	1.023973	35.905873	35.551847	36.343905	61.976504	60.642064	64.082573
		12 × 24	1.179041	1.134775	1.397371	44.839801	44.725211	45.042589	80.030816	78.763262	81.863866
		14 × 26	1.428508	1.336190	1.621330	54.881317	54.715436	54.977966	101.096280	99.036040	103.060893
		16 × 28	1.564394	1.535936	1.645911	65.994695	65.717291	66.320030	122.643763	121.453400	124.152609
	Centripetal	4 × 16	0.582793	0.556297	0.627398	15.353616	15.264652	15.562145	21.513359	21.038754	22.314871
		6 × 18	0.689854	0.675301	0.732897	21.121571	20.970719	21.257299	32.607861	32.171506	32.927922
		8 × 20	0.835003	0.807335	0.900578	27.855852	27.763438	27.948660	45.938474	45.409568	46.361467
		10 × 22	0.981428	0.957876	0.992468	35.887680	35.644112	36.046450	61.813699	61.307750	62.394765
		12 × 24	1.144271	1.122219	1.161745	44.849557	44.592495	45.132374	79.314098	78.557655	81.346112
		14 × 26	1.354391	1.334254	1.389894	55.072843	54.769977	55.589696	99.205564	98.054732	100.883410
		16 × 28	1.577983	1.537646	1.617448	70.740586	65.728494	74.495328	121.844101	120.240770	123.230783
	Exponential	4 × 16	0.570725	0.556716	0.636562	17.332781	15.485129	19.182170	21.152731	20.839127	21.364081
		6 × 18	0.703629	0.674907	0.819648	23.796082	22.300152	26.427910	32.346015	32.004818	33.215194
		8 × 20	0.854620	0.825496	0.884548	30.799544	29.055816	34.242564	45.461345	45.272974	45.647774
		10 × 22	1.031150	0.984631	1.076106	38.930154	37.482012	40.453657	61.016196	60.669912	61.745735
		12 × 24	1.187182	1.154661	1.261273	47.250221	45.760485	49.934564	78.622841	78.158778	79.955906
		14 × 26	1.387675	1.355736	1.442383	56.679949	55.869948	60.189400	100.047564	98.494069	102.146919
		16 × 28	1.619726	1.536456	1.741546	73.022933	69.451070	77.374287	123.066624	120.605328	130.466430
Oiltank	Uniform	4 × 16	0.597757	0.573778	0.620973	17.959995	17.152539	19.665036	21.493118	21.016233	21.862320
		6 × 18	0.722108	0.690122	0.772995	25.048991	23.823848	26.320465	32.930282	32.500726	33.900374

		8 × 20	0.843287	0.815232	0.880089	30.745225	29.111051	33.470097	45.845590	45.170128	46.595521
		10 × 22	1.007278	0.983544	1.033547	38.506784	38.012806	40.236376	61.500207	60.758329	62.268408
		12 × 24	1.173844	1.143008	1.208256	47.628593	47.358842	47.844646	78.821835	77.834549	79.866086
		14 × 26	1.384324	1.330275	1.438173	55.072192	54.344717	55.532536	99.724451	98.266908	101.531211
		16 × 28	1.649300	1.541569	1.782169	65.132682	64.004813	65.991436	122.175443	121.111021	123.345317
	Chord Length	4 × 16	0.608104	0.556610	0.817141	15.283098	15.152386	15.339689	21.528301	20.999459	22.276042
		6 × 18	0.704597	0.670145	0.785597	20.994082	20.921649	21.053370	32.862304	32.049560	33.352012
		8 × 20	0.829118	0.804201	0.905826	27.761828	27.598896	28.094516	46.208198	45.378588	46.852399
		10 × 22	0.991557	0.959747	1.081490	35.635930	35.517460	35.789401	62.590806	61.185396	64.450570
		12 × 24	1.145701	1.132211	1.186624	44.496126	44.276250	44.790530	79.872424	78.344727	82.890987
		14 × 26	1.354315	1.338341	1.393342	53.815391	53.397770	54.693792	99.541969	98.141881	101.003364
		16 × 28	1.580236	1.538629	1.626827	64.215986	64.025538	64.379131	121.600708	120.499340	122.638988
	Centripetal	4 × 16	0.579556	0.559627	0.609467	14.766338	14.623469	14.866158	21.986736	21.238878	23.322195
		6 × 18	0.695507	0.672668	0.745236	20.458448	20.302497	20.577630	32.666693	32.200807	33.186179
		8 × 20	0.829590	0.797621	0.914912	27.053456	26.874470	27.286766	44.210332	43.809581	45.592680
		10 × 22	0.989596	0.959219	1.046571	34.934592	34.784505	35.100661	59.195845	59.076195	59.325704
		12 × 24	1.156482	1.122337	1.270826	43.431872	43.217008	43.660599	76.076932	75.854023	76.370047
		14 × 26	1.358330	1.310792	1.418749	53.832041	53.224696	56.958809	95.092137	94.926030	95.283912
		16 × 28	1.578843	1.505062	1.618789	66.539634	64.719433	71.387395	116.199254	115.775157	116.647066
	Exponential	4 × 16	0.571319	0.554977	0.611887	17.954970	15.823586	20.253592	20.379396	20.263620	20.510987
		6 × 18	0.699100	0.674768	0.732843	21.971292	21.000761	22.518794	31.117579	31.039099	31.251819
		8 × 20	0.807051	0.803131	0.811122	28.420064	27.328188	29.560117	43.781133	43.696046	43.950700
		10 × 22	0.974598	0.956432	1.007020	37.262492	35.855965	42.670554	58.951059	58.802402	59.116571
		12 × 24	1.152988	1.138987	1.190482	44.442686	43.284482	47.148838	75.922674	75.689655	76.200879
		14 × 26	1.348849	1.317229	1.376976	53.428822	53.265831	53.733119	95.167119	94.921512	95.366325
		16 × 28	1.579280	1.540623	1.632245	64.308708	63.990682	64.608528	116.153811	115.892720	116.509148
Talus Bone	Uniform	4 × 16	0.582972	0.560772	0.627148	15.338758	15.227124	15.476213	21.392736	21.253802	21.615722
		6 × 18	0.699524	0.670006	0.740356	21.246188	21.082751	21.414455	32.555960	32.235261	32.997860
		8 × 20	0.823445	0.806806	0.869064	27.997091	27.822928	28.123913	45.757175	45.303353	46.268205
		10 × 22	1.002877	0.967773	1.043573	35.976624	35.849144	36.109206	61.331010	60.627592	62.013174
		12 × 24	1.198770	1.131366	1.276745	44.797103	44.578205	44.936692	79.161395	78.119295	80.007334

		14 × 26	1.432079	1.380499	1.541253	55.058486	54.907971	55.193927	99.696830	98.581196	101.022965
		16 × 28	1.618976	1.576506	1.643556	66.024178	65.903845	66.251617	119.872332	117.911322	123.433480
	Chord Length	4 × 16	0.601641	0.574760	0.635899	15.363030	15.278962	15.426382	21.457830	20.668099	23.542191
		6 × 18	0.736153	0.710934	0.755390	21.152336	21.034670	21.230449	32.107965	31.271108	33.896088
		8 × 20	0.850623	0.804370	0.943007	27.996059	27.771412	28.186760	44.658469	43.533643	46.625292
		10 × 22	1.035432	0.986297	1.133769	35.992872	35.795570	36.128133	64.239653	61.969577	67.136808
		12 × 24	1.212042	1.169196	1.266342	44.924694	44.579401	45.306874	80.541610	78.959892	82.613791
		14 × 26	1.415929	1.360527	1.539558	55.194733	54.883534	55.601465	106.600190	98.847061	119.660066
		16 × 28	1.629846	1.570080	1.688119	66.299630	66.133131	66.548294	121.249795	117.152039	123.912160
	Centripetal	4 × 16	0.595157	0.572452	0.640051	15.333088	15.240493	15.462058	20.932335	20.324772	21.786376
		6 × 18	0.694591	0.677487	0.725880	21.243836	21.131520	21.408165	32.768159	32.129229	33.448146
		8 × 20	0.877759	0.852318	0.933134	27.954149	27.858363	28.080114	46.331572	44.455403	47.235624
		10 × 22	1.005028	0.978860	1.049733	35.991403	35.709762	36.193022	62.665989	61.702224	65.399604
		12 × 24	1.200296	1.172342	1.244569	44.863516	44.650974	45.043227	80.052226	78.913571	81.258296
		14 × 26	1.409535	1.369410	1.481505	55.896879	55.417512	56.489742	101.040905	98.668089	102.391556
		16 × 28	1.634736	1.590654	1.727737	66.285483	65.861819	67.089882	123.162866	121.714159	124.864274
	Exponential	4 × 16	0.600442	0.571914	0.648361	15.327364	15.219504	15.457988	21.760800	21.479171	22.117842
		6 × 18	0.718811	0.690909	0.785943	21.121236	20.949237	21.253455	32.856950	32.493056	33.372366
		8 × 20	0.870529	0.810426	0.944757	27.873086	27.761087	28.027101	46.564739	45.853313	47.291339
		10 × 22	0.985965	0.964950	1.020940	35.887284	35.731813	36.032178	61.605738	59.040130	64.246078
		12 × 24	1.162945	1.137208	1.219805	44.724974	44.565575	44.929640	79.052903	75.605387	81.131413
		14 × 26	1.361896	1.333274	1.404147	55.073223	54.851671	55.305032	98.211603	97.507992	99.238201
		16 × 28	1.604932	1.549650	1.694426	66.024043	65.807445	66.346289	120.713160	118.810603	124.736262