A Discrete Differential Geometric Approach for Simulation of Coupled Multiphase Mesoporous Systems

Stefan Christian Endres, M. Eng.

November 30, 2023

A Discrete Differential Geometric Approach for Simulation of Coupled Multiphase Mesoporous Systems

Dem Fachbereich der Produktionstechnik
der
UNIVERSITÄT BREMEN

zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte

Dissertation

von

Stefan Christian Endres, M. Eng.

Gutachter: Prof. Dr.-Ing. habil. Lutz Mädler (Universität Bremen)
Prof. Dr.-Ing. Sergiy Antonyuk (Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU))

Tag der mündlichen Prüfung: 17.06.2024

Stefan Christian Endres, M. Eng.

A Discrete Differential Geometric Approach for Simulation of Coupled Multiphase Mesoporous Systems

PhD Thesis

Reviewers: Prof. Dr.-Ing. habil. Lutz Mädler and Prof. Dr.-Ing. Sergiy Antonyuk

University of Bremen
Faculty of Production Engineering
Chemical and Process Engineering
Badgasteiner Str. 3
28359 Bremen, Germany

Abstrakt

Eine Vielzahl von Prozessen und Systemen in der chemischen- und anderen Bereichen der Ingenieurwissenschaft sind mehrphasig. Anwendungen, die Prozesse mit mehreren Phasen und zunehmender geometrischer Komplexität beinhalten, sind in den letzten Jahren immer häufiger geworden. Versuche zur Modellierung und Simulation von Systemen mit einer großen Anzahl von Grenzflächen mit komplexen Geometrien werden häufig durch die Grenzflächennachführung und die Genauigkeit der Krümmungsberechnung eingeschränkt, die eine große Fehlerquelle der numerischen Methode darstellt. Diese Arbeit löst die seit langem bestehende Schwierigkeit, die Krümmung von Grenzflächen genau und effizient zu berechnen.

In dieser Arbeit wurde ein neues praktisches Werkzeug entwickelt, das in eine breite Palette von State-of-the-Art-Methoden implementiert werden kann. Es wird gezeigt, dass die rigorose Ableitung der Krümmung von Grenzflächen eine genaue oder nahezu genaue Berechnung der Krümmung für die Simulation von Dreiphasensystemen ermöglicht. Darüber hinaus wird gezeigt, dass ein spezieller trapezförmiger Integrationsfehler verwendet werden kann, um die Kontinuität der Grenzfläche abzuschätzen, was für die Gewährleistung der dynamischen Genauigkeit bei Dreiphasensimulationen nützlich ist, indem die Verfeinerung einer numerisch erfassten Fluid-Fluid-Grenzfläche, die vielfältig ist, kontrolliert wird.

Es wird weiter gezeigt, dass die hier entwickelte Methode zur Berechnung von Krümmungen für beliebige Fluid-Fluid-Grenzflächen, die vielfältig sind, genau berechnet werden kann. Das Ausmaß, in dem eine Grenzfläche vielfältig ist, kann durch Berechnung der Topologie des vereinfachten Komplexes mit Hilfe der hier entwickelten rechnerisch effizienten Methoden verfolgt werden.

Abstract (English)

A myriad of processes and systems in Chemical- and other fields of Engineering Science are multiphase. Applications that involve processes containing multiple phases of increasing geometric complexity have become increasingly common in recent years. Attempts to model and simulate systems with a large number of interfaces with complex geometries is often limited by the interface tracking and accuracy of the curvature computation which is a large source of error of the numerical method. This work resolves the long standing difficulty in computing the curvature of interfaces accurately and efficiently.

In this work a new practical tool was developed that can be implemented in a wide array of state-of-the art methods. It is shown that the rigorous derivation of the curvature of interfaces allows for an accurate, or near-exact computation of curvature for simulation of three-phase systems. Additionally, it is demonstrated that a special trapezoidal integration error can be used to provide an estimate of continuity of the interface, which is useful for ensuring dynamic accuracy in three-phase simulations by controlling the refinement of a numerically captured fluid-fluid interface that is manifold.

It is further shown that the method for computing curvatures developed herein can be computed exactly for arbitrary fluid-fluid interfaces that are manifold. The extent to which an interface is manifold can be tracked by computing the topology of the simplicial complex using computationally efficient methods developed especially for this application. Since the method for computing curvatures are near-exact and mesh independent, a very large tolerance can be used in large scale simulation of complex three-phase materials. In principle, it is only necessary to track that the interface is continuous. Therefore, the complex can be refined in such a way that the error matches the desired tracking of non-solid volume due to porosity and asperities in three-phase systems. The result of this work can be implemented in any formulation where the underlying geometry of the model is manifold, and is especially useful in the fields of multiphase flow, thermodynamics, materials engineering and surface energy minimisation.

What I have today mentioned as to the origin of the moving forces which are at our disposal, directs us to something beyond the narrow confines of our laboratories and our manufactories, to the great operations at work in the life of the earth and of the universe.

Hermann von Helmholtz

Preface

The core results of this thesis were published in the following works:

- **Endres, S. C.**, Ciacchi, L. C., Mädler, L. (2021) A review of contact force models between nanoparticles in agglomerates, aggregates, and films, *Journal of Aerosol Science*, 153, 105719, doi: 10.1016/j.jaerosci.2020.105719.

- **Endres, S. C.**, Avila, M., Mädler, L. (2022) A discrete differential geometric formulation of multiphase surface interfaces for scalable multiphysics equilibrium simulations", *Chemical Engineering Science*, Vol 257, 117681, doi: 10.1016/j.ces.2022.117681.

- **Endres, S. C.**, Mädler, L. (2022) Asymmetric Three-Phase Surface Tension Forces in Agglomerating Particulate Systems, *Chemie Ingenieur Technik*, 95, 278-283, doi: 10.1002/cite.202200128.

The main software libraries developed for this project were publicly released under open source MIT compatible licenses. All data used for test cases are also available which can be used the reproduced all the results of the test cases shown in this work:

- **Endres, S. C.**, Mädler, L. (2022) Leibniz-IWT/ddgclib: library for "Discrete differential geometry curvature in fluid simulations." First release (v0.4.9), Leibniz Institute for Materials Engineering IWT.
  https://github.com/Leibniz-IWT/ddgclib

- https://github.com/Stefan-Endres/hyperct

Acknowledgements

No work is done in isolation and this dissertation is certainly no exception. Above all the greatest debt for this work is owed to my advisor Professor Lutz Mädler who provided valuable input, feedback, insights and no small amount of patience throughout the research and writing process. I would also like to thank Professor Sergiy Antonyuk for many interesting discussions on the topic of agglomerating particles as well as for agreeing to review this dissertation.

Many fruitful discussion throughout the years have contributed directly or indirectly to refining ideas formulated in this work. Most significantly from our collaborators Professor Marc Avila and Professor Lucio Colombi Ciacchi, who's expertise and suggestions have significantly improved the quality of this work. Professor Michael Dreyer from who's lab course inspired the use of edge reconstruction in Sessile microdroplet test case. Detailed independent discussions with Doctor Holger Marschall and Doctor Tomislav Maric inspired the investigation of DDG errors on asymmetric geometric structures and the use of integrated quantities in simulation. A special thanks is owed my friends and mentors: Carl Sandrock, who taught me programming and inspired the interleaved vertex-vertex data structure eventually used in the $hyperct$ library, and Professor Walter Focke who taught me thermodynamics that proved essential for building the energy minimisation schemes used in this work. Additionally, thank you to my colleagues with whom I have had numerous fruitful discussions on dynamic three-phase contact angles and multiphase flow over the years: Patrick Giefer, Apostolos Kyrloglou as well as my collaborator Alexander Bußmann who developed the method for implementing the DDG formulation in a Level-set method framework.

Financial support by the German Research Foundation (DFG) through the Research Training Group GRK 1860 "Micro-, meso- and macroporous non-metallic Materials: Fundamentals and Applications" (MIMENIMA) for this project is gratefully acknowledged.

Finally, I would like to thank my family for their support: Peter and Micheala Endres for their hospitality and help with navigating the intricacies of the German system, my parents Christian and Vivienne Endres and my wife Esté Endres for her unwavering support and motivation throughout my doctoral journey without whom this dissertation would not have been possible.

Contents

List of Latin Symbols

Symbol	Description	Unit
A_i	Area of $st(i)$	m^2
A_{ijk}	Area of triangle ijk (discrete area)	m^2
A_m	Cross-section of the capillary bridge meniscus neck	m
a	Radius of a tube in the capillary rise system	m
C	Curve (smooth function)	-
C_i	Dual area around f_i (discrete dual area)	m^2
C_{ijk}	Area of the dual of the triangle *ijk* (discrete dual area)	m^2
$\mathbf{e_{ij}}$	Edge (discrete vector from vertex *i* to *j)*	m
$\mathbf{F}$	Force vector (physical quantity)	N
f	Surface function (smooth geometric property)	-
$\mathbf{f_i}$	Position of vertex (discrete position vector in $\mathbb{R}^3$)	m
H	Mean normal curvature of a surface embedded in in $\mathbb{R}^3$ (smooth geometric property)	m^{-1}
$\widehat{H}_i$	Mean normal curvature at vertex *i* (discrete point-wise integrated average)	m^{-1}
$\widehat{HNdA}_i$	Mean normal curvature at vertex *i* (discrete integrated one-form)	m
h_J	Height of a capillary film at equilibrium (Jurin's law)	m
$i,\ j,\ k,\ l$	Used to denote a vertex in the plane (discrete position vector in $\mathbb{R}^2$)	-
K	Gaussian curvature of a surface embedded in $\mathbb{R}^3$ (smooth geometric property)	m^{-2}
$\widehat{K}_i$	Gaussian curvature at vertex *i* (discrete point-wise integrated average)	m^{-2}
k_g	Geodesic curvature (smooth geometric property)	rad
$\widehat{k}_{g,i}$	Geodesic curvature at vertex *i* (discrete integrated quantity)	rad
Δp	Laplacian pressure difference (physical quantity)	Pa
$\updownarrow_{ij}$	Length of edge e_{ij}	m
$\mathcal{M}$	Manifold surface embedded in $\mathbb{R}^3$ (smooth, locally Euclidean in every neighbourhood on the surface)	-

Symbol	Description	Unit
$\mathbf{N}$	Unit normal (smooth normal vector)	-
$\mathbf{N_i}$	Unit normal at vertex *i* (discrete approximated normal vector)	-
$\mathbf{N_{ijk}}$	Unit normal of triangle *ijk* (discrete piece-wise linear normal vector)	-
n	Number of discretised points	-
R	Osculating sphere radius (smooth geometric quantity)	m
R_1, R_2	Osculating sphere radius corresponding to principal curvatures (smooth geometric quantity)	m
S	Surface segment (smooth)	m^2
$\widehat{S}$	Simplex (basic unit on a discretised surface)	-
st	Star operator	-
$\mathbf{T}$	Unit tangent vector (smooth tangent vector)	m
U_m	Cross-section of the capillary bridge meniscus neck	m

List of Greek Symbols

Symbol	**Description**	**Unit**
$\alpha_{\mathbf{i}}$	Smooth 1-form	-
$\widehat{\alpha_{\mathbf{i}}}$	Discrete 1-form (discrete integrated quantity)	m
$\alpha_{\mathbf{i}}$	Smooth 1-form	-
$\widehat{\alpha_{\mathbf{i}}}$	Discrete 1-form (discrete integrated quantity)	m
α, β	Interior angles defined in a simplicial complex	rad
∂	Boundary operator	-
γ	Surface tension (physical property)	N · m^{-1}
Θ_C	Three-phase contact angle (physical property)	rad
θ	Geometric angle (geometric quantity)	rad
$\widehat{\theta}$	Used to represent an integrated geometric angle (discrete integrated quantity)	rad
θ_i^{jk}	Interior angle at vertex f_i of triangle ijk (geometric quantity)	rad
κ	Scalar curvature (smooth)	m^{-1}
λ_c	Capillary length	m
ξ	Interior domain scalar quantity between two numbers in an interval in the trapezoidal rule	m
$\Delta\rho$	Density difference between two fluid phases	kg · m $^{-3}$
κ_1 , κ_2	Principle curvatures (smooth)	m^{-1}
Π	Disjoining pressure (physical property)	Pa
φ_{ij}	Dihedral angle at oriented edge e_{ij}(discrete)	rad
χ	Euler characteristic (topological property of a surface)	-

Abbreviations

ADF	Annular dark-field imaging
AFM	Atomic force microscopy
CFD	Computational fluid dynamics
DDG	Discrete- differntial geometry
DEM	Discrete-element models

DFT	Density functional theory
DPI	Dry powder inhaler
FTC	Fundamental Theorem of Calculus
GB	Gauss-Bonnet Theorem
MD	Molecular dynamics
JKR	Johnson, Kendall, & Roberts
PBE	Population balance equation
PP	Primary particle
TEM	Transmission electron microscopy
vdW	Van der Waals force

Introduction

1

Multiphase systems applied in engineering practice have become increasingly complex over the last century especially due to the advent of functional nanoscale materials. Systems of complex hierarchical mesoporous structures exhibit a rich array of phenomenological physics and chemistry that manifest as forces which drive the restructuring of materials and their properties. Due to the differences in scales of interest, with pores on the order of 2-50 nm to films on the order of several μm, a fundamental understanding necessitates the development models at the interface of surface science and continuum mechanics in order to understand and predict the material properties. In addition to modelling challenges, the large surface-area-to-volumes involved presents a unique challenges in computing the geometric data structures needed for the direct numerical simulations of structures. Of vital importance to such systems with large surface areas, and to multiphase systems in general, are the resulting surface tension forces which must be computed using curvature estimation; the central topic of this thesis.

1.1 Introduction to forces in mesoporous structures

Much of the progress in the scientific and engineering modelling and simulation of mesoporous materials stems from a desire to understand and optimise their production, structural integrity and performance. Due to the myriad of potentially useful applications in catalysis, battery materials, sensors and solar cells, there are many multi-physics and multi-scale problems that arise where nanoscopic properties affect bulk material properties on the order of several micrometres. In particular, the processes of fluid imbibition and drying. Fluid-solid interaction at this scale is largely driven by surface tension forces. The rich body of literature on this topic is reviewed in Section 2. In these systems nanoscopic primary nanoparticles (3-50 nm) aggregate to form chemically sintered aggregates (200-300 nm) which in turn agglomerate due to physical forces such as surface tension. As discussed in Section 2 restructuring can occur in these materials that is primarily driven by surface tension forces. Other important forces at this scale are van der Waals-, electrostatic-, repulsive-, friction forces and forces due to disjoining pressure. The simulation of surface tension in mesoporous systems has historically been challenging due to the inherent difficulty in computing the curvature of interfaces between phases, especially at the nanoscale which produce very large floating point errors and other more fundamental issues that have limited the application of direct numerical simulation in both engineering applications and advancing scientific understanding of mesoporous materials.

Therefore, the main focus of this work resolves the long standing difficulty in computing the curvature of interfaces and the applications of this in mesoporous materials such as nanoparticle films. Nanoparticle films are arguably the most challenging systems to model due to the restructuring of aggregates and primary nanoparticles which necessitate multi-phase simulation with moving solid bodies. The main applications are to understand the restructuring that occurs due to fluid imbibition and drying.

1.2 Restructuring in mesoporous systems and the need for direct numerical simulation

Restructuring in mesoporous systems is an important phenomenon that leads to changing characteristic macroscopic materials properties due to restructuring at the meso- and microscale. This phenomenon is especially common in multiphase processes such as coating, drying, imbitition and catalytic systems involving fluid-solid interaction as elaborated on in Section 2. While common in particulate systems with high shear gradients, it is also very common in nano- and microscale systems where the surface tension forces can dominate over viscuous forces. In such systems where surface tension forces dominate interface capturing becomes important for accurate simulation of the process. The difficulties in state of the art curvature computation methods for fluid-fluid interfaces and three-phase contact lines is explored in Section 2 and Section 3.

In order to derive salient quantitative information there is a need to track both the solid phases as well as the fluid phase in coupled multi-phase simulation using methods such coupled computational fluid dynamics (CFD) and discrete element method (DEM) simulation (called CFD-DEM). The large surface area present in nanoparticulate systems relative to the spatial volumes of interests presents a challenge interface tracking due to the large number of discretised points required to simulate the interface accurately. Therefore new approaches were needed in order track the interface as well as compute curvature accurately which is the origin of the surface tension forces that cause the restructuring. A new collection of methods and data structures based on discrete differential geometry is discussed in Section 3 which has direct applications in energy minimisation computations, and coupled multiphase CFD.

1.3 Interface curvature computation using discrete differential geometry

Energy minimisation computations are useful for finding equilibrium states of complex materials and parameterisation in coarse grained methods such as DEM. Meanwhile accuracy in coupled multiphase CFD is essential for modelling and understanding processes especially at scales that lack dynamic experimental data to validate the solver. While there are wide families of methods that can be classified within surface energy minimisation and multiphase CFD, all of them require an accurate computation of curvatures on the interface. Therefore, this work will focus on a general method that is applicable to all three-phase systems. Such a method is general because in 3 dimension there are only two geometric possibilities: contact between two phases (interface *surfaces*) and contact between three phases (three-phase contact *lines*, which in terms of differential geometry are more accurately described as *curves*).

Because of the importance of surface tension (and therefore curvature) in nanoparticle films and other multiphase processes where surface forces tension are dominant, it is worth developing a rigorous formulation built on the powerful theorems available in Differential Geometry, and the related field of Discrete Differential Geometry (DDG) in particular. Therefore, the focus of this work lies in gaining a deeper understanding of the curvature and surface tension of interfaces. The central hypotheses of this work are as follows:

Hypothesis H_1: A rigorous derivation of the curvature of interfaces will allow for an accurate simulation of three-phase systems by controlling the refinement of a numerically captured fluid-fluid interface that is manifold.

Hypothesis H_2: Curvature can be computed exactly for arbitrary fluid-fluid interfaces that are manifold, which allows for a very large tolerance of Hypothesis H_1 to be used in large scale simulation of complex three-phase materials.

Hypothesis H_1 and H_2 will be referred back to multiple times throughout this work. Section 3 reviews basic concepts from the mathematical theory of Discrete Differential Geometry which is fundamental to developing a new formulation that proves these hypotheses. Because of the generality of the formulation it is applicable to any data structure that is triangulable, which covers most families of CFD and energy minimisation surface tracking methods and can therefore be implemented common CFD solvers as will be discussed in Section 2.3.1.

1.4 Overview of Dissertation

An overview of this Dissertation is provided in Figure 1.1. The development of a new method based on discrete differential geometry for computing curvatures in three-phase systems as well as topological methods useful for the modelling and simulation of large interface areas is conducted in Section 3. The applications and validations on analytical solutions, experimental data and particle liquid bridges is bridges is shown in Section 4. The last two chapters 5 and 6 discuss the future of this work as well as the conclusions of the current dissertation.

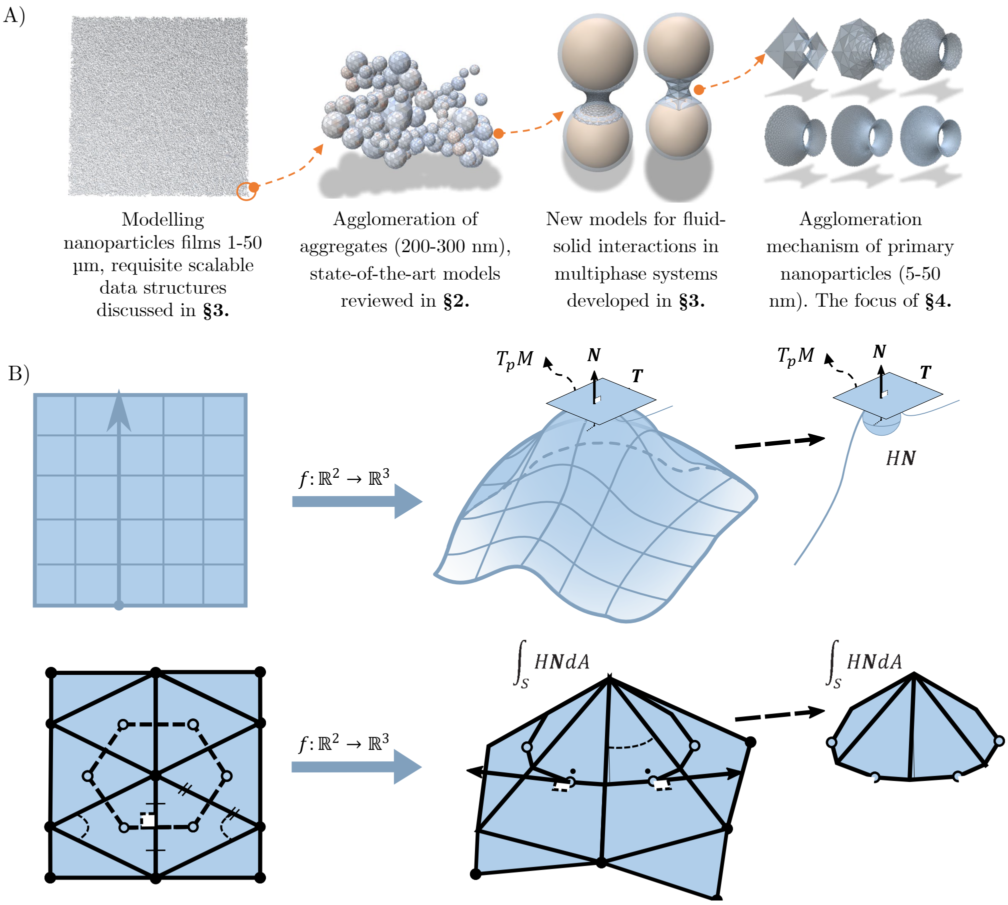
A)
Modelling nanoparticles films 1-50 µm, requisite scalable data structures discussed in §3.
Agglomeration of aggregates (200-300 nm), state-of-the-art models reviewed in §2.
New models for fluid-solid interactions in multiphase systems developed in §3.
Agglomeration mechanism of primary nanoparticles (5-50 nm). The focus of §4.
B)
$f: \mathbb{R}^2 \to \mathbb{R}^3$
T_pM
N
T
T_pM
N
T
HN
$f: \mathbb{R}^2 \to \mathbb{R}^3$
$\int_S HNdA$
$\int_S HNdA$

Figure 1.1.: A) An overview of this thesis. Mesoporous materials such as nanoparticle films are built up from a hierarchical assembly of nanoparticles, the complexity involved in capture the direct numerical simulation of fluid-fluid-solid interfaces is discussed in Section 3. In Section 2, the models for numerical simulation and how they are connected to multi-scale models is discussed. At the primary particle level the fluid-solid interaction is modeled at an aggregate level with new models and methods developed in Section 4. Finally, advances in particle-particle liquid bridges for differently sized particles are discussed in Section 4.2. B) The overarching idea in this work is the use of Discrete Differential Geometry to solve the problem of computing accurate curvatures on interfaces efficiently. The key constructions and theorems involved in discrete surfaces and their associated properties are discussed in Section 3

.

2 Forces in multiphase mesoporous systems

Literature review of force models and simulation methods in nanoparticulate systems, CFD-DEM methods and the discrete differential geometry of interfaces

In this Chapter the state-of-art models and methods available in literature for simulating mesoporous films are reviewed. A particular focus is placed on the contact forces between nanoparticles and their interaction with fluids in Section 2.1. The argument is made that the surface tension forces are primarily responsible for the structure of materials at this scale, which in turn necessitates the development of rigorous methods for direct numerical simulation and scientific understanding of agglomerating systems. In Section 2.3 CFD-DEM models used in engineering applications of agglomerating particulate systems as well as the latest developments in the field of contact between nanoparticles in mesoporous films in particular are reviewed, ending the discussion on phase interfaces and the current understanding of them. The field of differential geometry and curvature in particular is reviewed in Section 2.4 which becomes the fundamental building block for the methods developed in this work. As demonstrated in Section 2.3 a major limitation of multi-phase simulation is inaccurate curvature computations used to compute surface tension forces.

2.1 Contact Forces in Nanoparticulate Systems

The most general multiphase, mesoporous systems involve cases where all phases including the solid phases move. In this section the important contact forces involved in such nanoparticle systems are reviewed. This system is summarised in Figure 2.1 for mesoporous films, but similar concepts are applicable in all agglomerating systems where surface tension plays an important role. A detailed review of contact forces in agglomerating nano- and microparticle systems can be found in Endres, Ciacchi, and Mädler (2021)

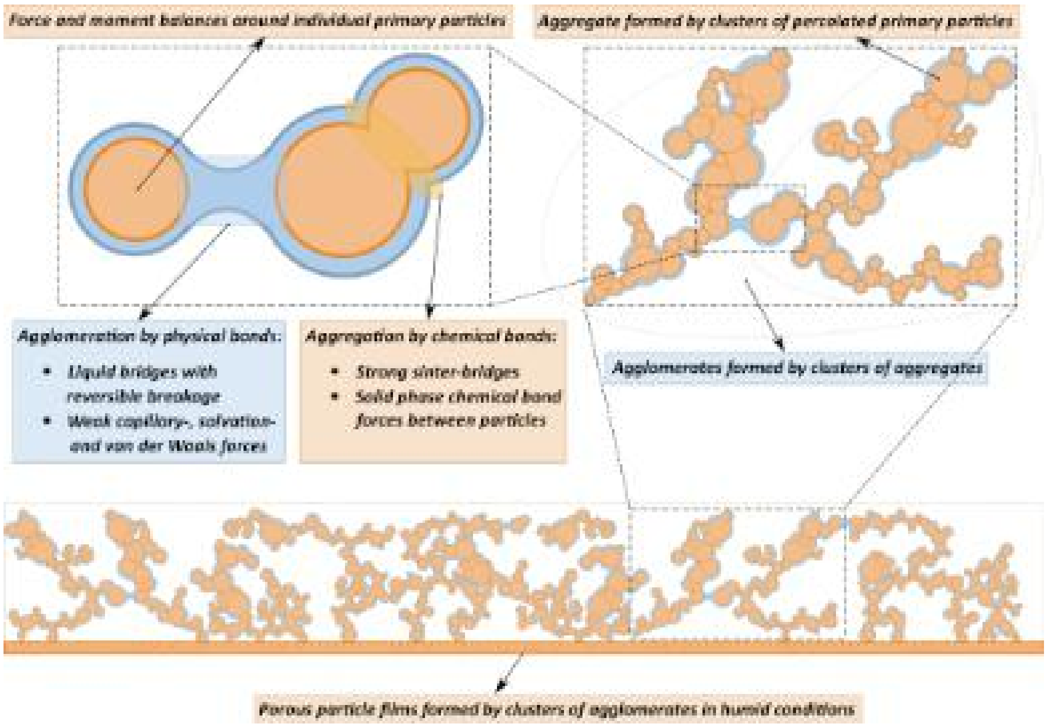
Force and moment balances around individual primary particles
Aggregate formed by clusters of percolated primary particles
Agglomeration by physical bonds:
Liquid bridges with reversible breakage
Weak capillary-, solvation- and van der Waals forces
Aggregation by chemical bonds:
Strong sinter-bridges
Solid phase chemical bond forces between particles
Agglomerates formed by clusters of aggregates
Porous particle films formed by clusters of agglomerates in humid conditions

Figure 2.1.: Hierarchical assemblies of nanoparticle films with a thickness on the order of 1–50 m is formed by agglomerates with typical size ranges bewteen 200–300 nm which are held together by weaker forces arising from electrostatics, van der Waals, solvation or capillary effects. These in turn are made up of chemically bonded aggregates which typically consist of a distribution of primary particles in the nanoscale range (5–50 nm)

and tables summarising all models is shown in Appendix A. As discussed in Endres, Ciacchi, and Mädler (2021) particle-particle contact forces F can be broadly categorised as follows:

1. Particle-particle repulsive forces F_{nr}.
2. Particle-particle attractive forces F_{na}:
   a) *Short range:*
      i. Van der Waals (JKR, Hamaker) F_{vdw}.
      ii. Coulombic forces F_e.
   b) *Long range:*
      i. Capillary forces F_{cap}.
      ii. Solvation forces F_{sol}.
3. Dissipative forces.
4. Sintering bridge forces.

While all these forces play a role in the final structure (and importantly, the macroscopic performance) of the material it should be noted that at

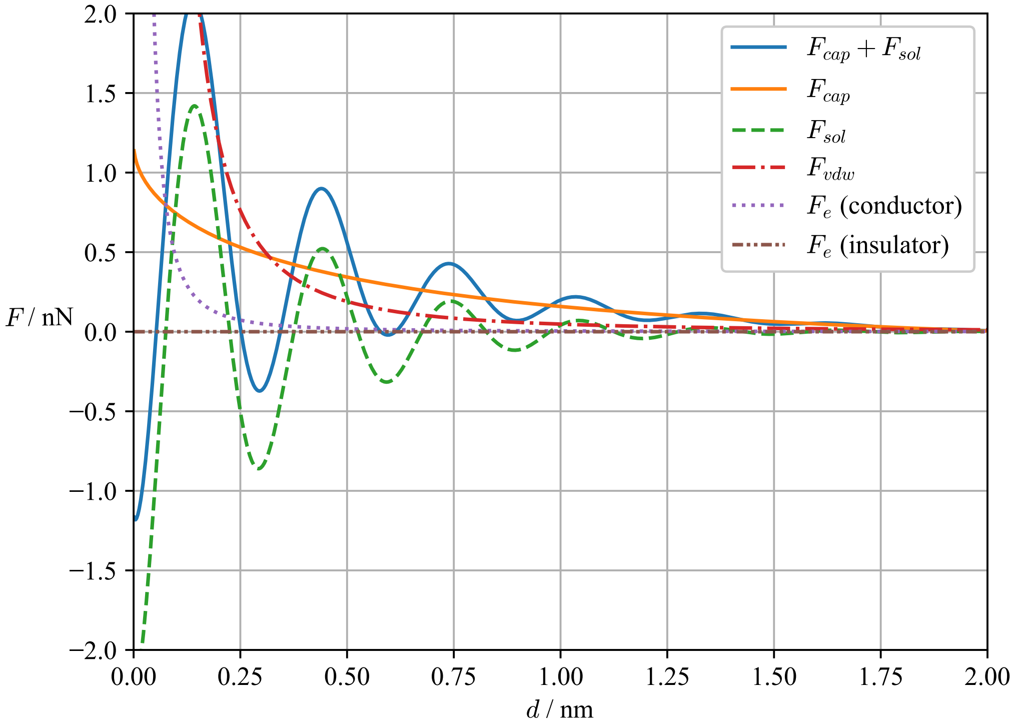

Figure 2.2.: Capillary and solvation forces can dominate the interaction at a critical distance which lead to various agglomerated particle-particle structures due to local minima in the force-distance curves.

small scales the agglomeration mechanism dominates over the other forces when liquid is present or if the even if the material is just exposed to humid conditions. This is primarily as a result of the fact that local energy minima of the system will converge to *long range* force minima as opposed to *short range* forces as demonstrated by the presence of local minima in Figure 2.2 for two unit charged 4 nm particles. Interactions in microscopic particles, where solvation is less prominent are dominated by the competition between capillary and repulsive forces -which occurs in the fluid phases as well as solid particle collisions- (Endres, Ciacchi, and Mädler, 2021).

Capillary forces occur due surface tension which arises due asymmetries in fluid-fluid interfaces and adhesion forces due to fluid-solid or three-phase fluid-fluid-solid interaction. While exact analytical solutions are known for particle-particle liquid bridges with particles of the same diameter (Schubert, 1984; Schubert, 2012; Dörmann and Schmid, 2014; Herminghaus, Semprebon, and Brinkmann, 2019; Wittel et al., 2019) it is less well established how arbitrary particle configurations should be discretised as shown in Figure 2.3 for applications in CFD-DEM and parameterisation, especially when complex topologies such as those shown in Figure 2.1 are present.

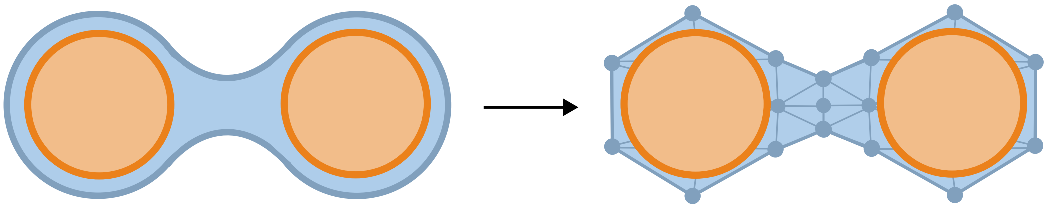

Figure 2.3.: The discretisation of liquid-liquid particle bridges is needed in order to predict the surface tension forces F_{cap} between the particles by integrating the surface tension at the interface. A central topic of investigation in this thesis is determining precisely how much refinement is needed in order to precisely predict the correct forces for arbitrary configurations and sizes of particles

The key conclusion of this Section is that while there are many forces at play that determine the final structure of agglomerating materials the Capillary force (due to surface tension) tends to dominate the energy minima in the mesoporous scale. In the next Section it will elaborated on the precisely what the surface tension forces are and how these arise as asymmetries in surface energies of phases in contact.

2.2 Description of gas-liquid-solid interfaces

In order to avoid confusion with the three-phase solid-fluid-fluid contact angle and the two contact angles of bridge models for describing wetted particle-particle liquid bridges it is useful to first consider investigation on a more fundamental case. The refinement and discretisation of the prototypical capillary rise phenomenon is shown in Figure 2.4. The capillary rise phenomenon has been studied quantitatively by Thomas Young as early as 1805 (Young, 1805).

Surface tension is the primary determinant of the equilibrium shape and mechanical driving force of gas-liquid-solid interfaces. It can be understood as a surface energy minimisation of inhomogeneous phase interfaces (e.g. gas-liquid-solid). This can be intuitively understood as the tendency of interfaces to shrink to the minimum surface area possible under volume and boundary constraints. The minimisation of surface energy is the main driving force in dynamic processes such as the formation of metallurgical microstructures, porous filters, and the restructuring of nanoparticulate systems (Endres, Ciacchi, and Mädler, 2021). In the Young-Laplace equation, which applies to macroscopic systems and extends down to the nanoscale (Liu and Cao, 2016; Laube et al., 2017), changes in mechanical pressure due to surface tension are driven by the mean curvature of the interface.

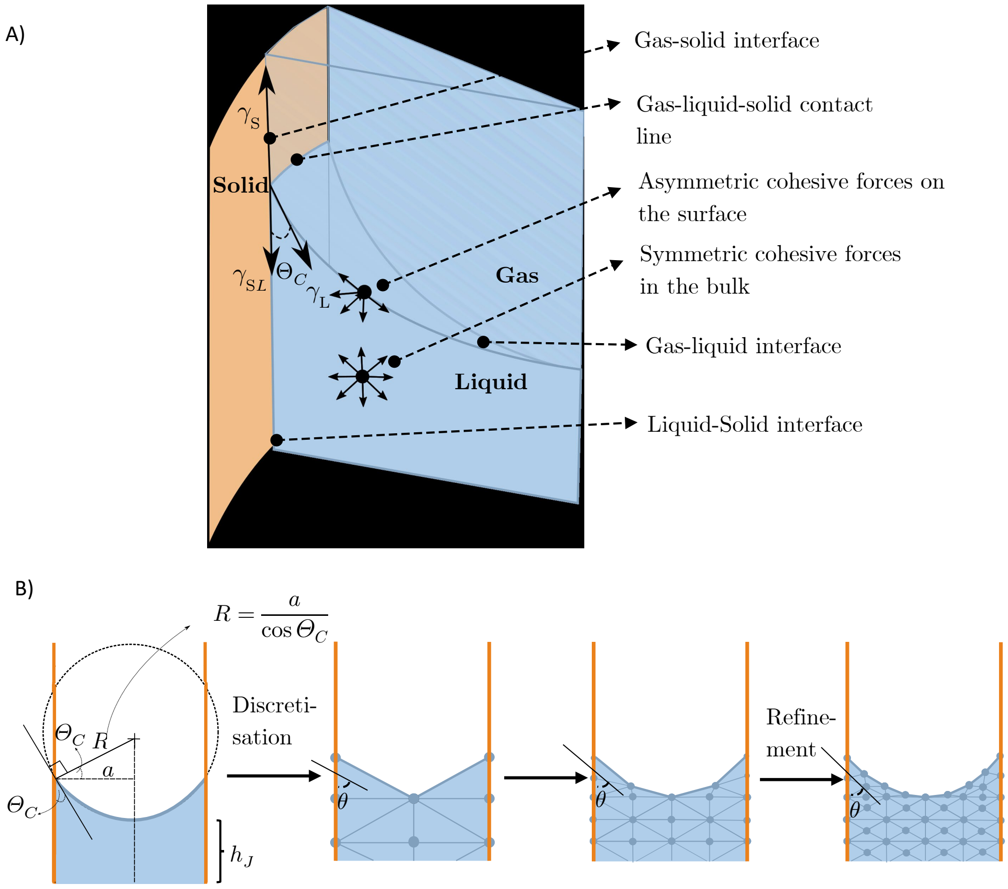

Figure 2.4.: A) The three-phase contact angle Θ_C is determined by the balance of three surface energies: $\gamma_S - \gamma_{SL} - \gamma_L \cos\Theta_C = 0$ Surface tension arises due to the asymmetric cohesive force on the surface of a fluid interface. The energy of the system is minimised when the local surface area is minimised subject to constraints. B) Idealised capillary rise geometry with a symmetric discretisation shown as an example. The angle Θ_C is the three-phase contact line, a is the radius of the tube, R is the radius of the spherical cap or osculating sphere, and $h_J = \frac{2\gamma\ \cos\Theta_C}{\rho g a}$ is the Jurin height. The discretised contact angle θ only approaches the true smooth contact angle Θ_C for very fine discretisations, which are computationally expensive and exhibit low convergence.

A generalisation of the Young-Laplace equation that accounts for deviations due to non-continuous effects, providing a direct link between the number density profiles of surface interfaces (found using methods such as MD simulations as a function of height above the solid interface h) and the surface tension $\gamma(h)$), is given by the Young-Laplace-Derjaguin equation (MacDowell, 2017; MacDowell et al., 2018):

$$\Delta p(h) = \gamma(h)H - \Pi(h). \tag{2.1}$$

The mechanical pressure drop between fluid phase surface interfaces is denoted by Δp. The surface tension is γ, and H is twice the mean curvature. The liquid interface film height is h and the disjoining pressure at a given height is $\Pi(h)$. The surface tension can be expressed as $\gamma(h) = \gamma_{1v} - \frac{\xi^2 d\Pi}{dh}$, where the macroscopic surface tension is denoted by γ_{1v} and it is perturbed by the disjoining pressure gradient $\frac{\xi^2 d\Pi}{dh}$ (MacDowell, 2017; MacDowell et al., 2018). The term $\Pi(h)$ allows the model to account for non-continuum effects near an interface, such as solvation forces (Endres, Ciacchi, and Mädler, 2021). These effects become important in nano- and mesoscale systems, where the capillary and solvation forces are dominant.

The prototypical example of surface tension modelling is the idealised capillary rise experiment, which is modelled using the Young-Laplace-Gauss equation (Gauss, 1877). In a capillary rise system, liquid adhesion to the walls of the tube causes the fluid to rise or fall. If the tube is sufficiently narrow, the surface interface can be approximated as a spherical cap. In this case, the Young-Laplace equation has an exact analytical solution for Δp, which is related to the radius of the osculating sphere, R:

$$\Delta p = \gamma H = \frac{2\gamma}{R}. \tag{2.2}$$

This idealised solution is useful for the numerical comparison of errors between any developed framework and the physical solution.

The equilibrium of the interfacial energies at gas-liquid-solid interfaces can be parameterised by the contact angle Θ_C (see Figure 2.4A). In any numerical discretisation of gas-liquid-solid interfaces, the discretised contact angle θ (or the measured geometric angle) only approaches the true smooth contact angle Θ_C for a very fine discretisation and will never theoretically result in the true value, as shown in Figure 2.4B. In many formulations, the handling of the contact angle is based on heuristics developed from numerical studies (Chen, 2013).

The large number of discretised elements required to approximate interfaces (such as mesh elements or sampling points) is a challenge for most simulation methods. The reader is referred to De Vaucorbeil et al. (2020) for an extensive review of surface tension formulations in material point methods. Similar issues are present in other computational fluid dynamics (CFD) methods, where three-phase contact angles can be modelled as a boundary condition. In some cases, it is impossible to simulate systems with large representative volume elements (i.e., large enough to compute the bulk properties of a system) without extensive computational resources over the desired real time. Some examples of numerical CFD methods used in multiphase fluid simulation include volume of fluid (VoF), constraint interpolation (CIP), and level set methods. In a dynamic system with fluid phases coupled to moving solid phases, these methods require iterated calculations of fluid fields with the coupled interfaces at each timestep.

The remeshing step is computationally expensive and lacks a formal method for computing the accuracy of the discrete approximation scheme. For these reasons, a new formulation for surface interface modelling and their three-phase contact lines was developed based on recent advances in discrete differential geometry (DDG). Currently, there are no definitions of a physical boundary condition for the three-phase contact angle Θ_C (as opposed to a geometric contact angle θ) available in DDG literature. The main contribution of this work is to provide a discretised formulation of Θ_C (Chapter 3) to a DDG framework, which accurately represents the true smooth three-phase contact angle Θ_C, as demonstrated later.

This formulation allows for the use of minimal computational resources to capture interfaces in three-phase processes to a specified degree of accuracy. In a dynamic framework, this formulation would allow the solver to refine the mesh in domains with large momentum gradients and otherwise sparsely sampled interfaces at a specified accuracy. The formulation described here is most suited for coupling with mesh free methods such as the Material Point Method (MPM), recently reviewed by De Vaucorbeil et al. (2020). In the present work, only the energy of the surface itself is minimised (described in more detail in Section 3), and the possibility of coupling the formulation to a momentum solver is briefly discussed. In order to provide the broader context of this work within the context of dynamic multiphase CFD the next subsection will provide a brief overview of the state of the art multiphase CFD applications involving CFD with wet particulate systems.

2.3 Computational Fluid Dynamics - Discrete Element Methods

2.3.1 Overview of multiphase CFD-DEM methods

The field of multiphase CFD is still an active field of development with a large number of methods available suitable to different use cases. These can often be broadly categorised as pure Eulerian as well has many hybrid Euler-Lagrange approaches, some popular approaches include:

1. The volume of fluid (VOF) method.
2. Smoothed-particle hydrodynamics (SPH).
3. Level-set method (LSM).
4. Lattice Boltzmann methods (LBM).
5. Material point methods (MPM).
6. Particle-in-cell (PIC) methods.

One of more of these methods is further combined with an interface capturing method such as the popular scheme by Roenby, Bredmose, and Jasak (2019) in combination with VOF method (Noh and Woodward, 1976; Hirt and Nichols, 1981) that can be used to compute the curvature and subsequently the surface tension forces. In order to place the hypotheses of this work into greater context some of the most prominent studies using CFD-DEM methods in field of Chemical Engineering are reviewed. Many of these studies are based on the popular CFDEM© software (Hager, Kloss, and Goniva, 2018). The model behind this software, as well as an array of friction models (describing the fluid-solid component of the momentum exchange) is reviewed in Section B. In general, the surface tension force F_s can be linearly added to the momentum balance equation. Surface tension in turn is computed from the local curvature of the interface or estimated from coarse grained DEM models.

2.3.2 CFD-DEM simulation applications

In simulations it is of interest to understand the change in structure of the particulate systems, either because the performance of the final material depends on that structure or because the drag force coefficient C_D is changed by the geometry of the particulate structure which in turn affects the fluid velocity field and flow properties of the process as discussed in Section B.

An important class of materials is aerosol-synthesised materials which are naturally formed as aggregates and agglomerates of primary nanoparticles. The porous, fractal, and complex hierarchical structure of these materials results in transport properties in gases that differ from those of spheres. In addition to particle size, the complex geometry of nanoparticle structures plays a key role in nanoparticle transport parameters such as the friction coefficient (Mädler and Friedlander, 2007).

The coupling of fluid mechanics to DEM is widely employed in simulating processes such as the production of nanoparticle powders in fluidised bed reactors (Anantharaman, Ommen, and Chew, 2015; Martín et al., 2014; Diez et al., 2019; Fabre et al., 2016; Fabre et al., 2017; Jin, Kleijn, and Ommen, 2017; Liu et al., 2016; Mogre et al., 2017; Salikov et al., 2015; Sutkar et al., 2016; Salameh et al., 2017) and particle separation in cyclones (Misiulia, Lidén, and Antonyuk, 2021; Misiulia, Nedumaran, and Antonyuk, 2023). Additionally, this coupling approach is used for modeling the agglomeration and fragmentation of dry powder inhalers (Pei et al., 2013; Pei et al., 2015; Tong et al., 2013; Tong et al., 2015; Yang, Wu, and Adams, 2014; Yang, Wu, and Adams, 2015).

These selected use cases and more are reviewed in detail in Section B. An important remark is that most of these applications are of of the unresolved and two-way coupled flavour suitable for the scale of the processes studied in those work. However, more recently there has been an increasing focus on direct four-way coupled numerical simulation where the details of the interface affect the fluid flow and it is desired to understand how much refinement is needed. The problem with such processes is the poor scalability of current methods. The confirmation of Hypothesis H_1 is essential in this context for accurate simulation using the minimal amount of computational resources. In particular, direct numerical simulation is essential in order to understand that the restructuring that occurs in nanoparticle films produced by the FSP process (Schopf, Salameh, and Mädler, 2013a; Laube et al., 2018;

Baric, Ciacchi, and Mädler, 2019; Schopf et al., 2017) where no dynamic validation is available to understand the changes in mesoporous sturctures during processes such as fluid imbibition in catalytic, sensor and battery systems.

For most numerical schemes, the accuracy of the simulation is dependent both on the refinement of the discretisation as well as the magnitude of the local field gradients and curvature. For example, for the VOF method (Jasak, 1996) found that the number of computational points required is large where the solution of the gradient is large and depends on the local curvature. In multiphase flow an interface capturing method is required in order to approximate the geometry of the interface so that curvature can be computed locally. These methods in turn rely on finite difference methods such as second order Crank-Nicholson or the adapted from Arbitrary-Lagrangian-Eulerian (ALE) techniques (Hirt, Amsden, and Cook, 1974) and marker and cell (MAC)-like schemes (Harlow and Welch, 1965) developed for continuum surface force (CSF) model developed by Brackbill, Kothe, and Zemach (1992).

What all these schemes have in common is that they compute the gradient of the normal using finite difference approximation. In developing Hypothesis H_2, it is important to highlight loss of geometric knowledge in the integrated approximations of gradients to find the surface tension force in contrast to exact computation of integrated forces. In particular, when computing surface tension forces it is important to note the distinction between *integrated* and *point-wise* forces. For example, Brackbill, Kothe, and Zemach (1992) distinguish between the volumetric surface tension force $\boldsymbol{F}_{\mathrm{sv}}$ and a surface tension force per unit interfacial area $\boldsymbol{F}_{\mathrm{sa}}$ related through:

$$\lim_{h\to 0}\int_{\Delta V}\boldsymbol{F}_{\mathrm{sv}}d^3x=\int_{\Delta A}\boldsymbol{F}_{\mathrm{sa}}dA, \tag{2.3}$$

where h is the thickness of arbitrary edges that are normal to the surface. The term $\boldsymbol{F}_{\mathrm{sv}}$ can be used directly in the momentum balance, for example in the Navier-Stokes equation, by bilinear addition to the right hand side:

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t}+(\mathbf{v}\cdot\nabla)\mathbf{v}\right)=-\nabla p+\mu\left(\nabla^2\mathbf{v}\right)+\rho\boldsymbol{F}_{\mathrm{sa}}. \tag{2.4}$$

In this context both $\boldsymbol{F}_{\mathrm{sv}}$ and $\boldsymbol{F}_{\mathrm{sa}}$ can be refered to as point-wise forces that act on the fluid at a point in the field. This is equal to the force when the Young-Laplace equation Equation 2.1 is integrated over some arbitrary patch of area of the surface. By integrating both sides of Equation 2.4 over some arbitrary volume element Ω (substituting the surface force Equation 2.3 that only acts on the interface):

$$\int_{\Omega} \rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} \right) dV = \int_{\Omega} -\nabla p dV + \int_{\Omega} \mu \left(\nabla^2 \mathbf{v} \right) dV + \int_{S} \rho \boldsymbol{F}_{\mathrm{sa}} dA. \quad (2.5)$$

The entirety of the term $\int_S \rho \boldsymbol{F}_{\mathrm{sv}} dA$ is an integrated force. It is important to note that most CFD methods actually solve the integrated version of the momentum balance equation as written in Equation 2.5 through the various discretisation methods outlined at the beginning of this chapter. However, as will be discussed in Section 3 any discrete data structure embedded in $\mathbb{R}^3$ can be triangulated and therefore its integrated gradients can be computed. In order to preserve consistent notation this work will only refer to the area surface tension force as F_s with the understanding that this force acts on a local volume element due to the presence of a curved interface. The surface tension force is defined as

$$\boldsymbol{F}_{\mathrm{s}} = \gamma H \mathbf{N}, \quad (2.6)$$

where γ is the surface tension, H is the mean curvature and $\mathbf{N}$ is the unit vector normal to the surface. Many interface capturing methods, such as the method by Brackbill, Kothe, and Zemach (1992) compute the mean curvature H at a point on the surface calculated using

$$H = -(\nabla \cdot \mathbf{N}), \quad (2.7)$$

Further in the CSF model, the normals of the gradients are computed from a mollified color function $\tilde{c}$ at a point on the surface:

$$\mathbf{N} = \frac{\nabla \tilde{c}}{|\nabla \tilde{c}|}, \quad (2.8)$$

so

$$H = -(\nabla \cdot \frac{\nabla \tilde{c}}{|\nabla \tilde{c}|}), \quad (2.9)$$

Other definitions can include estimation from volume fractions, but are similar in form to Equation 2.9 in the sense that it is required to compute the Laplace-Beltrami operator $\Delta = \nabla \cdot \nabla$ of the surface. Because this involves a second order approximation, this can result in large errors especially in regions where curvature is high which additionally results in other numerical artifacts such as spurious currents. The main focus of this work is to eliminate that error entirely and develop integrated methods that can be applied towards more rigorous interface formulations. In order develop a general formulation of curvature in discrete interfaces, key concepts from the field of smooth differential geometry are first reviewed, this background will be essential for development of a discrete formulation that follows in Section 3.

2.4 Smooth differential geometry of general surfaces

Before a general formulation of a discrete differential geometric framework can be developed, that is computable and applicable to a CFD framework, it is necessary to first review key concepts from the field of smooth differential geometry and then develop a discrete formulation based on the theorems discussed in this section.

2.4.1 Smooth differential geometry of general surfaces embedded in $\mathbb{R}^3$

The mean curvature normal $H\mathbf{N}$ of an interface point has a general geometric interpretation on an arbitrary smooth mathematical surface, as illustrated in Figure 2.5. It represents the deviation of the local gradient of the surface and is given by:

$$H\mathbf{N} = \nabla \cdot \mathbf{N} \ = \nabla \cdot \nabla f, \tag{2.10}$$

where $\mathbf{N}$ is the normal vector of the surface f at a point. It is important to understand unambiguously what is meant by surface in order to understand the limitations of the formulation developed in this work. More rigorous definitions adapted from literature (Lee, 2003, p. 3) are:

Definition 1. *A **manifold** $\mathcal{M}$ is a topological space that is locally Euclidean, that is every point has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$.*

Additionally adopting the definition of a surface (Henle, 1979) with or without a boundary:

Definition 2. *A **surface** $\mathcal{S}$ is a manifold of dimension two embedded in $\mathbb{R}^3$.*

These definitions apply to the all physically realisable phase surface interfaces, but notably excludes certain important abstracted models that do not fit within these definitions. For example, a 1D string model attached to a 2D surface is not manifold at the point of intersection. In order to use the methods developed in this work the 1D model must be mollified (given an extruded finite volume). These definitions will be referred back to in Section 3 when developing the discrete equivalents, but first curvature on smooth surfaces is reviewed. Mathematical surfaces can often be represented by a parametric function that maps a $\mathbb{R}^2$ domain to its embedding in $\mathbb{R}^3$ $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ as shown in Figure 2.5. An important example of such a function for validating numerical solutions of liquid bridges with analytical solutions is the Catenoid function (explained later in Equation 4.3).

In order to fully understand the mean curvature H, the smooth Gaussian curvature and the principal curvatures are first introduced. At a given point s on a 1D curve $C(s)$ embedded in a 3D surface (Figure 2.6), there exists a unique tangent vector $\mathbf{T}(s)$, normal vector $\mathbf{N}(s)$, and the (scalar) radius of an osculating circle $R(s)$. The scalar curvature $\kappa(s)$ is simply the inverse radius of this osculating circle $\kappa(s) = \frac{1}{R(s)}$. The vector and scalar quantities are related by the equations $\kappa = |\kappa \mathbf{N}| = |d\mathbf{T}(\mathbf{X})| = 2\pi/(2\pi R) = 1/R$, where $\mathbf{X}$ is any vector on the tangent plane projecting $\mathbf{T}$. At any given point on a curve $C(s)$, a tangent vector can be rotated to any direction θ as shown in Figure 2.6). The mean normal point curvature $H(s)$ can be defined as an integral over the curvature $\kappa(s, \theta)$, omitting the parameter s for clarity:

$$2H = \frac{1}{2\pi} \int_0^{2\pi} \kappa(\theta) d\theta \tag{2.11}$$

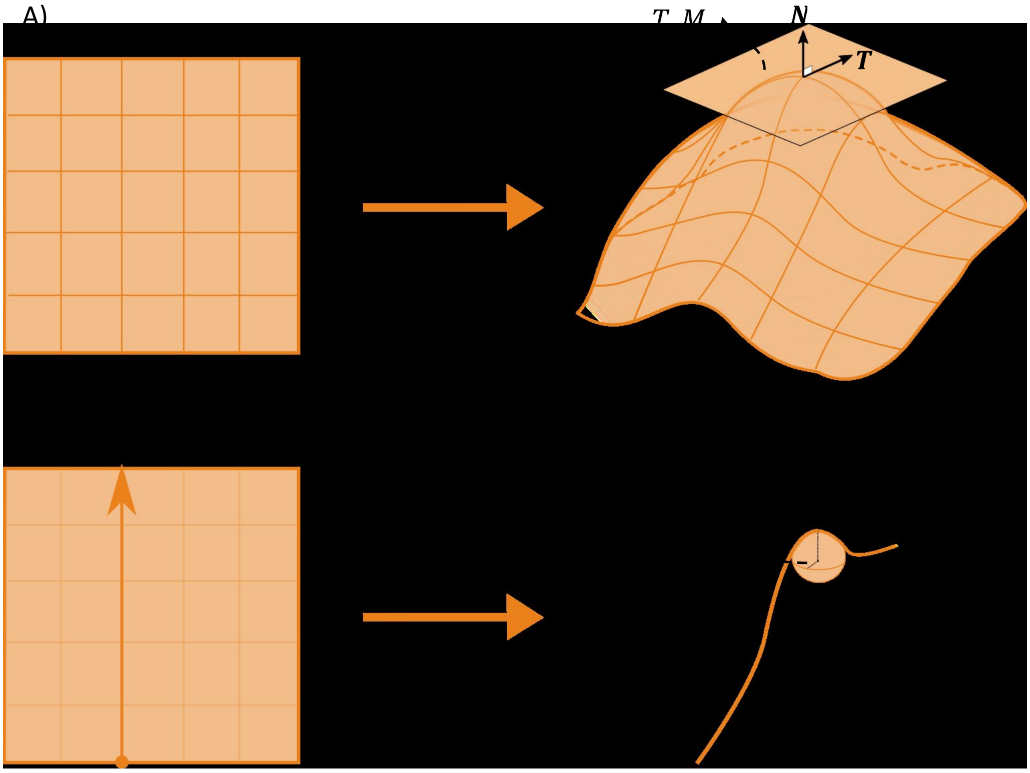

Figure 2.5.: A smooth mathematical surface can be represented by a mapping $f : \mathbb{R}^2 \to \mathbb{R}^3$ that transforms a 2D plane into a 2D surface embedded in a 3D space. Every point on the surface is associated with a tangent plane T_pM, a normal vector $\mathbf{N}$, and a tangent vector $\mathbf{T}$, which are related through the equation $d\mathbf{T}(\mathbf{X}) = -\kappa\mathbf{N}$, where κ represents the curvature in the direction $\mathbf{X}$ (for example, defining a curve embedded in the surface). An example of a 1D curve $C : I \to \mathbb{R}^3$ parameterised by $s \ \in I \in \mathbb{R}$ cut out from this plane provides a definition for an osculating circle $R(s)$.

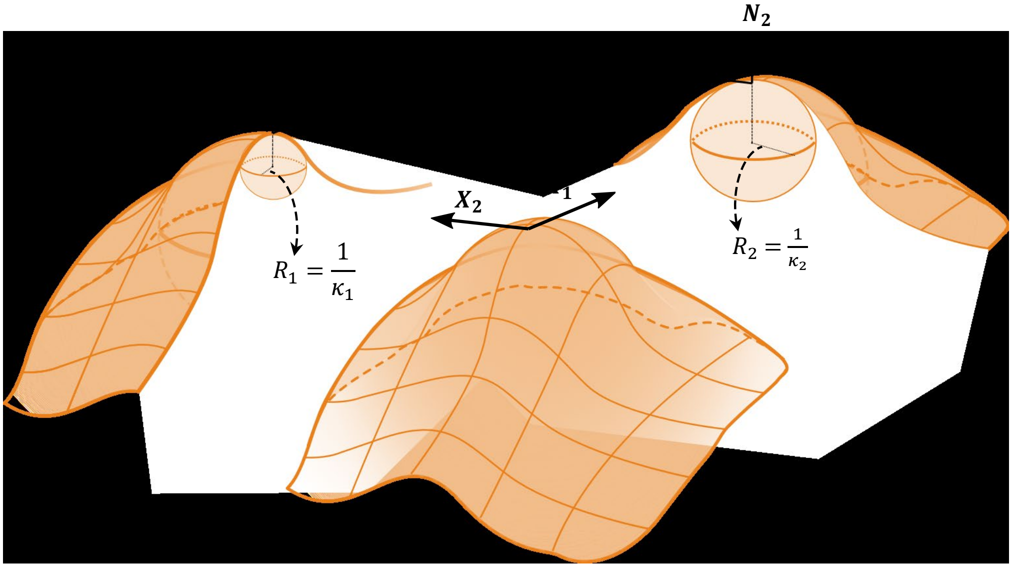

Figure 2.6.: The principal curvatures κ_1 and κ_2 are displayed alongside their osculating spheres. Notably, the angle θ of a tangent vector T can be determined at any point on the surface.

The principal curvatures κ_1 and κ_2 represent the maximum and minimum curvatures obtained when rotating θ at a given point on the surface. They are mathematically defined as:

Definition 3. *The two **principal curvatures*** are defined as $\kappa_1 = \max_\theta \ \kappa(\theta)$ and $\kappa_2 = \min_\theta \ \kappa(\theta)$.

The tangent vectors pointing in the directions corresponding to $\max_\theta \ \kappa(\theta)$ and $\min_\theta \ \kappa(\theta)$ are denoted as $\mathbf{X_1}$ and $\mathbf{X_2}$ in Figure 2.6, respectively. Euler's theorem states that the *mean curvature* H can be determined by summing the principal curvatures:

$$H = (\kappa_1 + \kappa_2) \, . \tag{2.12}$$

Note that Equation 2.12 is sometimes referred to as H_f to distinguish it from the more common mean curvature definition, $H_d = \frac{1}{2}(\kappa_1 + \kappa_2)$. Due to its prevalence in computational fluid dynamics Equation 2.12 is utilised throughout this work without the subscript f to avoid confusion. The mean curvature H is a scalar quantity which is less useful than the *mean curvature vector* $\mathbf{H}$, for an arbitrary parameterised surface (Do Carmo, 2016, p. 203):

$$\mathbf{H} = H\mathbf{N} \tag{2.13}$$

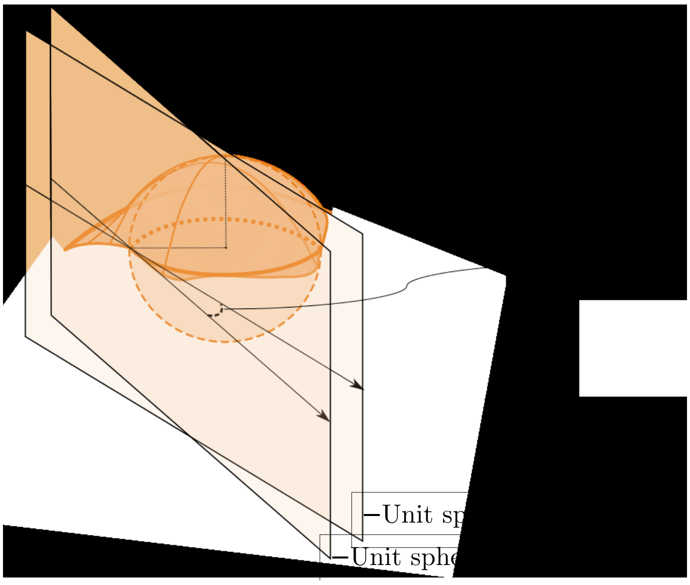

Figure 2.7.: The geodesic curvature is a measure of how much a curve deviates from being a geodesic on the unit sphere.

The Gaussian curvature, denoted by K, represents the product of the principal curvatures:

$$K = \kappa_1 \kappa_2 \tag{2.14}$$

2.4.2 Geodesic curvature, the Gauss-Bonnet Theorem and Stokes' Theorem

A final important curvature essential of modelling three-phase contact lines, k_g represents the geodesic curvature, as defined formally in Do Carmo (2016, p. 252), adapted here:

Definition 4. *Let C be an oriented regular curve contained in an oriented surface $\mathcal{S}$, and $C(s)$ is a parametrisation of C, in a neighbourhood of $p \in \mathcal{S}$. The* ***geodesic curvature*** k_g is the value of the covariant derivative of $C'(s)$ at a point p.

Intuitively, the geodesic curvature captures the local deviation of a curve C from being a geodesic on a sphere as demonstrated in Figure 2.7.

The total integrated Gaussian curvature $\int_{\mathcal{M}} K dA$ of a bounded surface $\mathcal{S}$ and the boundary integral $\int_{\partial\mathcal{M}} k_g ds$ (see Section 3 for a more detailed explanation of this integral) are connected through the surface's Euler characteristic χ

(e.g., $\chi = 1$ for a bounded disk). This relationship is known as the Gauss-Bonnet Theorem:

Theorem 1. ***Gauss-Bonnet Theorem:***

$$\int_{\mathcal{M}} K dA + \int_{\partial \mathcal{M}} k_g ds = 2\pi\chi. \tag{2.15}$$

The Gauss-Bonnet Theorem, a local-global theorem, establishes a crucial link between the local curvatures and the global characteristics of surfaces. The theorem applies to any smooth, orientated and bounded surface, including piecewise linear manifolds, enabling the formulation of a Gauss-Bonnet Theorem which also applies piece-wise to the discrete setting.

The equivalence of the Gauss-Bonnet Theorem in the smooth and discrete settings allows for an integral formulation in which our discrete contact angles are defined in a manner consistent with the smooth definition of Θ_C for any number of sampling points. In practice, this enables the use of significantly fewer computational elements in simulations while preserving the properties of the smooth model equations. A final important and powerful Theorem is Stokes' Theorem which states:

Theorem 2. ***Stokes' Theorem:*** *If c is a smooth k-chain in a smooth manifold $\mathcal{M}$, and ω is a smooth $(\mathrm{k}-1)$-form on $\mathcal{M}$, then*

$$\int_{\partial \mathcal{M}} \omega = \int_{\mathcal{M}} d\omega$$

Intuitively Stokes' Theorem can be interpreted as an extension of the Fundamental Theorem of Calculus (FTC) to higher-dimensional manifolds. For instance, in the case of a one-dimensional function scalar f (a 0 -form), the integral of the differential $f' = \frac{\mathrm{d}f}{\mathrm{d}x}$ (a 1 -form) can be determined by calculating the difference at the domain's boundaries. The formal definitions of a k-chain will be provided provided in Section 3.

2.5 Conclusions

In this Chapter an overview was provided of surface tension in multiphase systems and the importance of the force in various applications. It was demonstrated that the method to compute surface tension in CFD and energy minimisation both involve the computation of curvature computed from the the Laplacian of the interface. Developing a common framework would allow for a fast and accurate method that can be used for CFD and energy minimisation useful finding equilibrium structures as well DEM contact force parameterisation.

Therefore, the focus of this work lies in gaining a deeper understanding of the surface tension and discrete curvature. The next chapter will review basic concepts from the mathematical theory of Discrete Differential Geometry which is fundamental to developing a new formulation that proves the central hypotheses of this thesis as outlined in Hypothesis H_1 and H_2. Because all discrete data structures can be triangulated and represented as piecewise linear complexes on which the Gauss-Bonnet -and other key theorems applicable smooth and Riemannian manifolds- apply, many of the definitions from smooth differential geometry discussed in Section 3 can be applied in order to develop rigorous definitions in the discrete setting even when full knowledge of the surface is not available.

3

Discrete differential geometry for surfaces with dynamic topologies

Geometric and Topological Methods and Data Structures for Surfaces with Dynamic Topologies

This Chapter discusses methods and a library which can capture a surface interface and compute properties such as curvature efficiently under the conditions of dynamically changing topologies. The novelty of the method in comparison with the state of the art is that most efficient solvers in the field of discrete differential geometry make use of a fixed incidence array which must be rebuilt every time the topology or connectivity of the mesh changes. By contrast the method and data structures described in this Chapter is fully dynamic. Additionally, tracking connectivity and topology is inherent without the need for expensive mesh traversal operations that limit the use of DDG in dynamic frameworks.

Before developing the discrete differential geometric notions of curvature, it is required to develop a well defined discrete representation of a surface. Such constructions must have several important properties defined in order to ensure robustness and stability in any simulation. In particular the focus is consistently proving that the data structure meets the criteria of a surface as defined in Definition 2. This can be done by computing the topology of the surface. Critically, using the homology as a numerical criterion is used to prove Hypothesis H_1. Additionally the topology, while cheap to compute with the method described here, can provide valuable physical insights into a system such as the number of bubbles or pores present in a phase, the percolation paths of solid porous structures and so forth without the need for more expensive clustering, segmentation and reconstruction.

3.1 Discretisation of surfaces

In this section an object central to modelling of interfaces and their curvatures is exposited; the simplicial complex. A simplicial complex results whenever a manifold embedded in any a space $\mathbb{R}^d$ of any dimension d is **triangulated**, the terms "triangulation" and "simplicial complex representation" can therefore be used interchangeably in this context. When the triangulation of any Definition 2 (i.e. a manifold according to Definition 1) is can be found then a discrete curvature is always well defined as will be demonstrated in Section 3.4. A more general structure than a simplicial complex restricted to $\mathbb{R}^2$ is used because this is useful for computing more complex information such as the integrated curvature tensors of interest in this work. It is important to emphasise that all discrete data structures that can be embedded in a $\mathbb{R}^n$ and supplied with a metric (forming a topological space) which can be triangulated, this includes meshes of any geometric structures (polygons, cubes etc.), point-clouds (through a Delaunay triangulation), smooth surfaces with analytical expressions etc. The mathematical proof of the latter is provided later in this Section.

A fundamental problem in holding these geometric data structures in computer memory in is elaborated on in Section 3.2 which largely deals in methods to alleviate this problem as well as applications of the new data structure. First, it is shown how surfaces such as that shown in Figure 2.5 can be constructed explicitly in a discretised way. A fundamental problem in discretising smooth surfaces -and in fact all smooth mathematical objects such as vector fields- is that only a finite number of discretised points can be stored to represent an object that is defined at an infinite number of points. This is a source of numerical error and can lead to instabilities if the surface is not well refined. However, as will be demonstrated in this work, by a careful application of the rigorous theorems for differential geometry, it is possible to eliminate numerical errors completely; even with representations that have low refinement in certain cases when the geometry (and topology) is well understood such as for curvature as per Hypothesis H_2.

A demonstration of the discretisation of a spherical cap is shown in Figure 3.1. Many issues are immediately apparent in this representation. First, it is seen that curvature is not well defined on a discrete surface since it is zero everywhere except at the vertices where the Laplace-Beltrami operator Δf is not defined due to a discontinuity. Secondly the interior angles at the vertex point sum to more that 2π as would be expected in a smooth geometry.

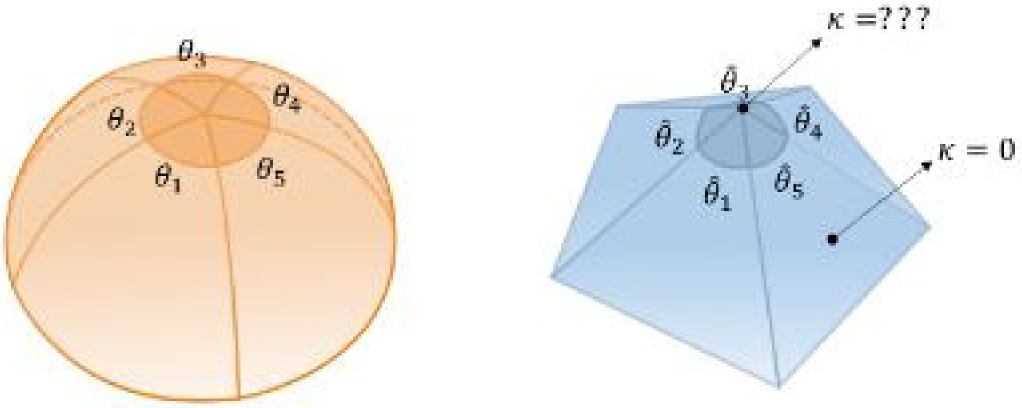

Figure 3.1.: Discretisation of a spherical cap. While curvature is well defined as the laplacial of the surface function f in a smooth context, the curvature is not well defined on a discrete surface since it is zero everywhere except at the vertex where the Laplace-Beltrami operator Δf is not defined due to a discontinuity.

Before any formalism is introduced it is helpful to understand how a triangulation of a surface f is found in general. Later a more rigorous definitions of all these concepts will be derived. For now consider the parameterised function of the torus, this equation describes the mapping $f : \mathbb{R}^2 \to \mathbb{R}^3$ where $f(u, v)$ is the vector function:

$$\begin{aligned} x_1 &= (a \cos u + b) \cos v, \\ x_2 &= (a \cos u + b) \sin v, \\ x_3 &= a \sin u. \end{aligned} \tag{3.1}$$

Where the domain is $u \in [0, 2\pi]$ and $v \in [0, 2\pi]$. The output $\mathbf{x} = (x_1, x_2, x_3)$ is the mapped position in $\mathbb{R}^3$ space. In Algorithm 2 (which requires many more constructions) it is explained how this can be computed explicitly. For now an intuitive description is that the *domain* of the function can be triangulated and then the triangulation can be mapped to the corresponding vertex positions in $\mathbb{R}^3$ as shown in Figure 3.2. The next Section will detail precise how this can be constructed efficiently.

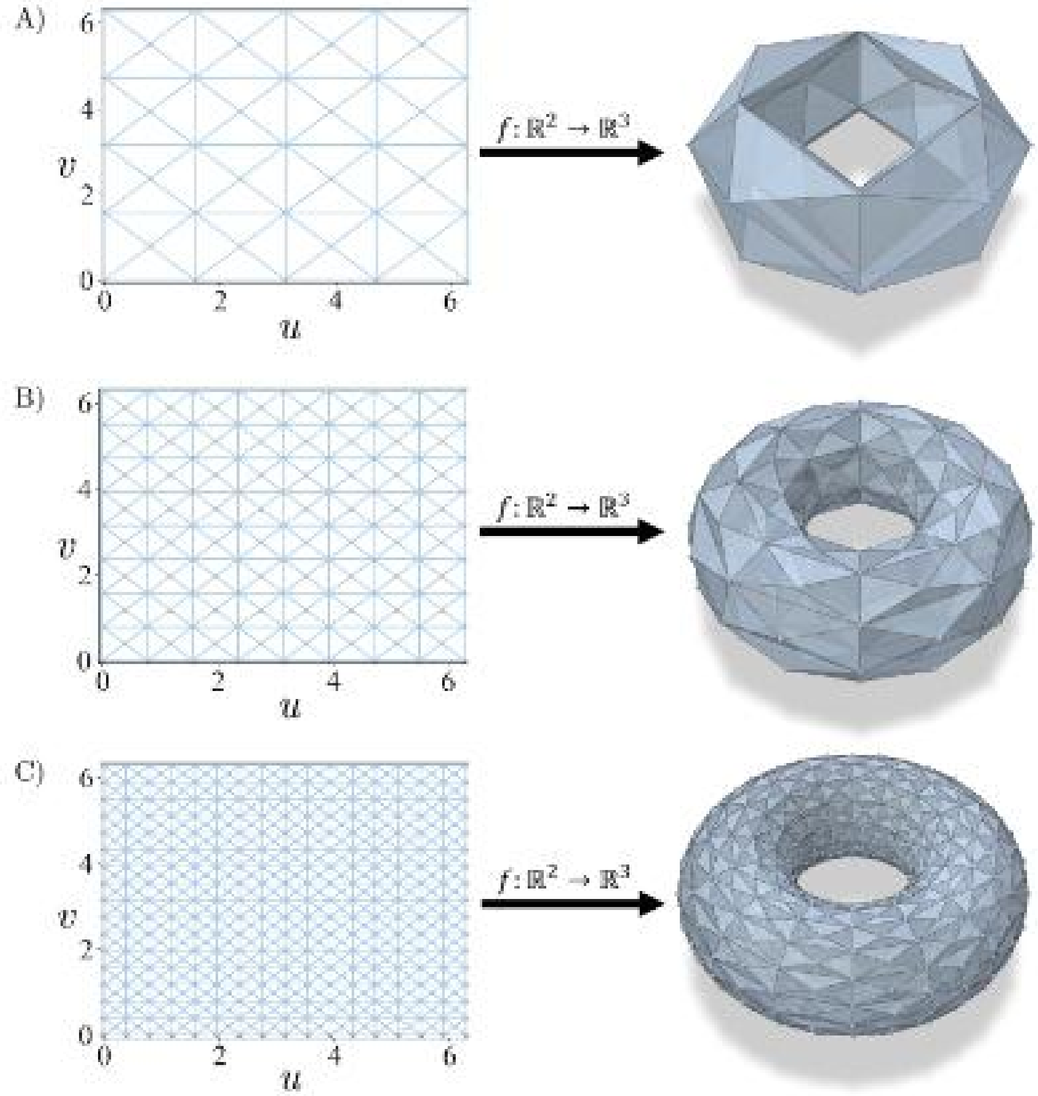
A)
B)
C)
$f: \mathbb{R}^2 \to \mathbb{R}^3$
$f: \mathbb{R}^2 \to \mathbb{R}^3$
$f: \mathbb{R}^2 \to \mathbb{R}^3$
v
u
0
2
4
6

Figure 3.2.: Triangulations of the surface function Equation 3.1. Subplots A, B and C show increasingly fine triangulations

3.2 Theoretical detail

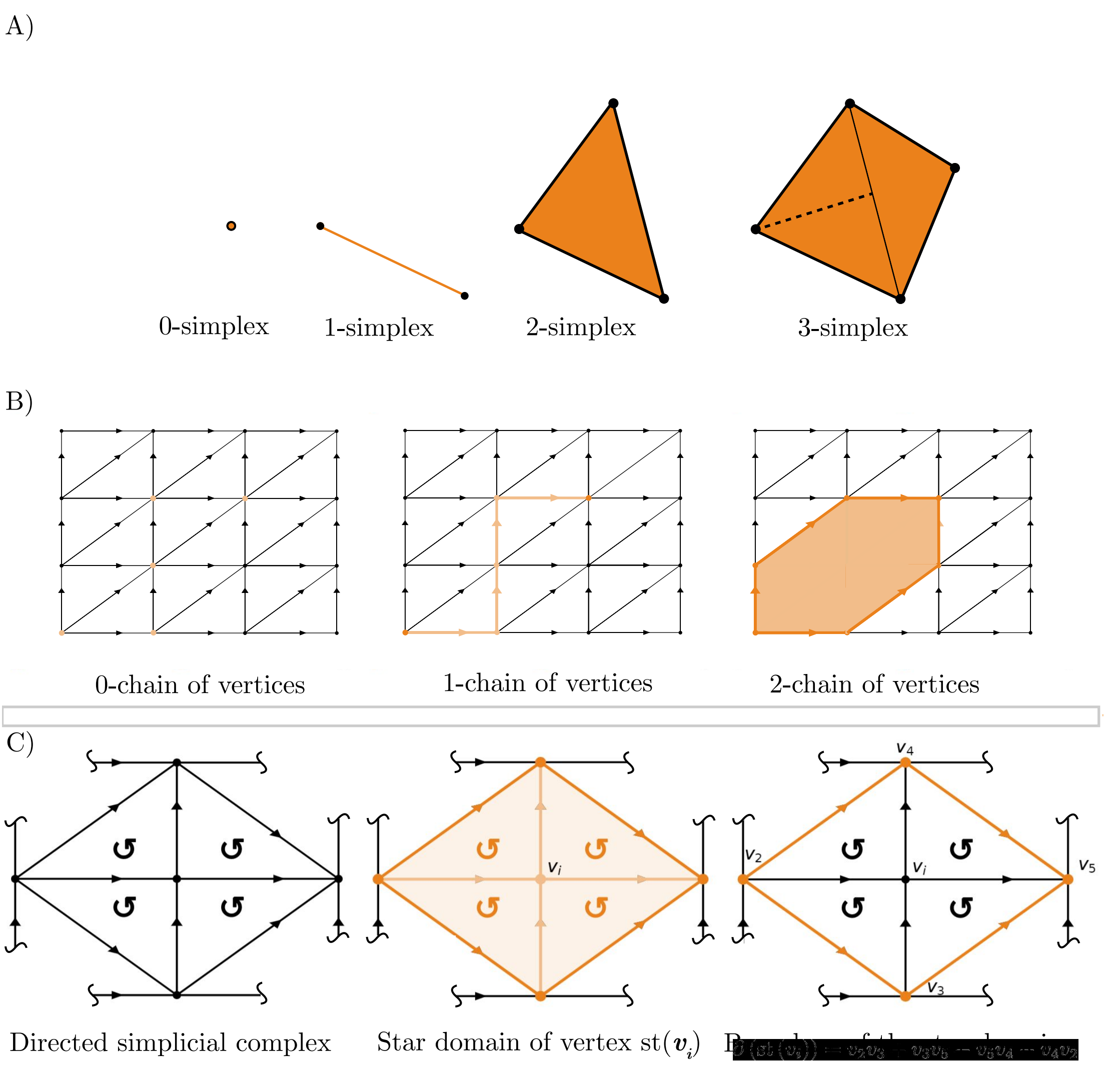

Figure 3.3.: **(A)** 0-simplex (point), 1-simplex (edge), 2-simplex (triangle) and a 3-simplex (tetrahedron). **(B)** 0-chain of vertices, a 1-chain of edges and a 2-chain of simplices. **(C)** Directed 2-simplex in the directed simplicial complex (left), star domain defined by $\mathrm{st}\,(v_i)$ (centre) and it's boundary defined as $\partial\,(\mathrm{st}\,(v_i)) = \overline{v_2 v_3} + \overline{v_3 v_5} - \overline{v_5 v_4} - \overline{v_4 v_2}$ (right)

The purpose of this section is to describe a discrete mapping $h : \mathcal{P} \rightarrow \mathcal{H}$ to provide a simplicial approximation for the (hyper-)surface f. Describing this construction will require several concepts from algebraic and combinatorial topology (Henle, 1979). The following definition was adapted from Hatcher (2002, p. 9), a visual demonstration of the concepts defined here is also shown in Figure 3.3:

Definition 5. *A **d-simplex** is a set of $d+1$ vertices in a convex polytope of dimension d. Formally if the $d+1$ points are the $d+1$ standard $d+1$ basis vectors for $\mathbb{R}^{(d+1)}$. Then the d-dimensional d-simplex is the set*

$$S^d = \left\{ (t_1, \dots, t_{d+1}) \in \mathbb{R}^{d+1} \mid \sum_{1}^{d+1} t_{d+1} = 1, t_i \geq 0 \right\}$$

For example, a 2-simplex is a triangle and a 3-simplex is a tetrahedron. The following combinatorial definition of a simplicial complex is used (Hatcher, 2002, p. 107):

Definition 6. *A **simplicial complex** $\mathcal{H}$ is a set $\mathcal{H}^0$ of vertices together with sets $\mathcal{H}^d$ of d-simplices, which are ($d+1$)-element subsets of $\mathcal{H}^0$. The only requirement is that each ($k+1$)-elements subset of the vertices of an d-simplex in $\mathcal{H}^d$ is a k-simplex, in $\mathcal{H}^k$.*

Thus each d-simplex has $d+1$ distinct vertices, and no other d-simplex has this same set of vertices. The superscript $\mathcal{H}^k$ represents the subset of $k-$dimensional simplices where for an d-dimensional problem the highest dimensional k-simplex contains $d+1$ vertices. Finally, the definition of a k-chain (Henle, 1979):

Definition 7. *A* k-***chain*** *is a union of simplices with an Abelian algebra defined on the chain arithmetic.*

For example a 0-chain is a set of vertices, a 1-chain is a set of edges and a 2-chain is a set of triangles. $C(\mathcal{H}^k)$ denotes a $k-$chain of $k-$simplices. A vertex in $\mathcal{H}^0$ is denoted by v_i. If v_i and v_j are two endpoints of a directed edge in $\mathcal{H}^1$ from v_i to v_j then the symbol $\overline{v_iv_j}$ represents the edge so that it is bounded by the $0-$chain $\partial\left(\overline{v_iv_j}\right) = v_j - v_i$ and similarly for an edge directed from v_j to v_i is $\partial\left(\overline{v_jv_i}\right) = \partial\left(-\overline{v_iv_j}\right) = v_i - v_j$. Higher dimensional simplices can be represented and directed in a similar manner, for example a triangle consisting of three vertices v_i, v_j and v_k directed as $\overline{v_iv_jv_k}$ has the boundary of directed edges $\partial\left(\overline{v_iv_jv_j}\right) = \overline{v_iv_j} + \overline{v_jv_k} + \overline{v_jv_i}$. It should be noted that only mod 2 incidence and therefore the coefficients of the chain group are limited to $\{0, 1\}$ in the domain.

A special kind of surface with boundary is now defined (Henle, 1979, p. 129) which is what will be used throughout this work as all three-phase fluid interfaces are of this type:

Definition 8. *A **surface with boundary** is a topological space obtained by identifying edges and vertices of a set of triangles according to all the requirements of the definition of a surface except that certain edges may not be identified with another edge. These edges, which violated the definition of a surface, are called **boundary edges**, and their vertices, which also violate the definition of a surface, are called **boundary vertices**.*

This also leads to an important theorem:

Theorem 3. ***Classification Theorem for Surfaces with Boundary.*** *Every compact, connected surface with boundary is equivalent to either a sphere or a connected sum of tori or a connected sum of projective planes, in any case with some finite number of disks removed.*

Theorem 3 is proven by triangulation of plane models (Henle, 1979, p. 129-131). Therefore since all plane models can be triangulated (explicit algorithms are provided in Section 3.2.2) it is mathematically proven that all possible surfaces (as defined by Definition 8) can be triangulated. This ensures that the algorithms developed here are applicable to all physical multiphase fluid systems that can simulated regardless of the geometric or topological complex of the system.

Finally, the abstractions used the simplicial complex will defined explicitly. First by formally defining the set of vertices from which 0-chains of the simplicial complex $\mathcal{H}$ are built and the edges from which the 1-chains of $\mathcal{H}$ are built. Definitions 9 through 13 provide this construction for h:

Definition 9. *Let $\mathcal{X}$ be the set of sampling points generated by a sampling sequence. The set $\mathcal{P} = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{g}(\mathbf{x}) \geq 0\}$ is a set of points within the feasible set bounded by constraints* $\mathbf{g}$.

Definition 10. *For any function f, $\mathcal{F}$ is the set of scalar outputs mapped by the objective function $f : \mathcal{P} \rightarrow \mathcal{F}$ for a given sampling set $\mathcal{P} \subseteq \Omega \subseteq \mathbb{R}^n$.*

Definition 11. *Let $\mathcal{H}$ be a directed simplicial complex. Then $\mathcal{H}^0 := \mathcal{P}$ is the set of all vertices of $\mathcal{H}$.*

Definition 12. *For a given set of vertices $\mathcal{H}^0$, the simplicial complex $\mathcal{H}$ is constructed by a triangulation connecting every vertex in $\mathcal{H}^0$. The triangulation supplies a set of undirected edges E.*

Definition 13. *The set $\mathcal{H}^1$ is constructed by directing every edge in E. A vertex $v_i \in \mathcal{H}^0$ is connected to another vertex v_j by an edge contained in E. The edge is directed as $\overline{v_i v_j}$ from v_i to v_j iff $||f(v_i) - f(v_j)|| < 0$ so that $\partial\left(\overline{v_i v_j}\right) = v_j - v_i$. Similarly an edge is directed as $\overline{v_j v_i}$ from v_j to v_i iff $||f(v_i) - f(v_j)|| > 0$ so that $\partial\left(\overline{v_j v_i}\right) = v_i - v_j$.*

For practical computational reasons the case where $f(v_i) = f(v_j)$ should also be considered. If neither v_i or v_j is already a minimiser (see Definition 14), the rule is used that the incidence direction of the connecting edge is always directed towards the vertex that was generated earliest by the sampling point sequence. If v_i is not connected to another vertex v_k then the notation $\overline{v_i v_k}$ is left undefined and let $\partial\left(\overline{v_i v_k}\right) = 0$. The higher dimensional simplices of $\mathcal{H}^k, k = 2, 3, \ldots d + 1$ are directed in an arbitrary direction which completes the construction of the complex $h : \mathcal{P} \to \mathcal{H}$. It should be noted that the function f encompasses a wide class of mathematical objects. Of particular importance is the position vector (as in the introductory example shown in Figure 3.1 where f is parameterised surface function) that will later be used to compute curvatures. Other important possible functions are fluid scalar and vector field values such as pressure and velocity. These can all be mapped onto the same complex and the homologies of the fields can be computed.

The final definition is useful when computing homologies of a function associated with a hypersurface, this function includes, for example, the gradient and the curvature of the surface itself (Endres, Sandrock, and Focke, 2018; Endres, Avila, and Mädler, 2022):

Definition 14. *A vertex v_i is a minimiser iff every edge connected to v_i is directed away from v_i, that is $\partial\left(\overline{v_i v_j}\right) = (v_{j \neq i} - v_i) \vee 0 \; \forall v_{j \neq i} \in \mathcal{H}^0$. The minimiser pool $\mathcal{M}$ is the set of all minimisers.*

Definition 14 provides a definition of one type of homology, it is worth briefly reviewing the homology of surfaces from its chain algebras. The chain groups

$C_k(\mathcal{H}^k)$ are related boundary groups B_k of $C(\partial_k \mathcal{H})$. The k-chain groups all have known ranks $c_k = |C_k(\mathcal{H})|$. The (Abelian) homology group is:

$$H_k(K) = \frac{\ker \partial_k}{\operatorname{im} \partial_{k+1}}$$

The Betti-number is simply the cardinality of the $k-$groups:

$$h_k = |H_k(K)|$$

Some special relationships are available for surfaces, let Z_0, Z_1, Z_2 be the groups of cycles $\ker \partial_2 = Z_2$, $\ker \partial_1 = Z_1$ and $\operatorname{im}(\partial_2) = B_1$, $\operatorname{im}(\partial_1) = B_0$. By convention $Z_0 = C_0$ and $B_2 = \{\varnothing\}$ for surfaces. Let c_0, c_1, c_2, b_0, b_1, b_2 be the ranks of all these groups, then according to Theorem 4 (Henle, 1979, p.164):

$$c_2 = z_2 + b_1$$

$$c_1 = z_1 + b_0$$

Applying Theorem 4 from Henle (1979, p.164):

$$h_0 = z_0 - b_0$$

$$h_1 = z_1 - b_1$$

$$h_2 = z_2 - b_2$$

where $b_2 = 0$ and $z_0 = c_0$. Note also that $c_0 = |V|, c_1 = |E|, c_2 = |F|$. Additionally the homology groups are related to the Euler characteristic:

$$\chi = h_0 - h_1 + h_2 \dots (-1)^{k-1} h_k$$

for torus $\chi = 0$, 2-connected tori $Xi = -2$. n-connected tori:

$$\chi = 2 - h_1$$

for 1-torus ($h_0 = 1, h_1 = 2, h_2 = 1$), $\chi = 2 - 2 = 0$, for 2-connected torus ($h_0 = 1, h_1 = 4, h_2 = 1$), $\chi = 2 - 4 = -2$ and for 3-connected torus ($h_1 = 6$), $\chi = 2 - 6 = -4$. The Euler characteristic for Euler graphs is also known:

$$V - E = h_0 - h_1$$

Which is useful for computing the h_1 groups on the domain of the structure presented here since h_0 is easier to compute after cycles are computed with the boundaries.

These homology groups, while being cheap to compute, provide two important numerical parameters useful in simulation:

1. The homology group can be tracked for each refinement. This means that Hypothesis H_1 is solved by providing a powerful criteria for when the topology of an interface is fully understood and further refinement is not needed and/or lower refinements can be used. When an exact curvature computation is available as per Hypothesis H_2 all surface tension forces in a system can be computed exactly.
2. The homology groups provide information about the *structure* of the physical material, such as the number of bubbles (h_2) in a fluid, as well as the percolation paths through a solid material and fluid imbibition network (h_1), without additional segmentation steps which are computationally expensive.

Computational expense and scalability is major topic of research in itself. It is especially important in fluids where numerical methods should ideally scale linearly with the number of discrete elements in a mesh. Most efficient solvers in the field of discrete differential geometry make used of a fixed incidence array (Meyer et al., 2003; Elcott et al., 2007; Mullen et al., 2009; Bell and Hirani, 2012), other formulations of single phase CFD using DDG are built on a fixed mesh (Mohamed, Hirani, and Samtaney, 2016), but these structures are expensive to recompute for interfaces with changing topologies due to a required re-meshing step which also requires the construction of a new incidence array. For this reason a new data structure was needed that can capture the abstract simplicial complex constructions efficiently in code. That is the topic of the next section.

3.2.1 Linear growth data structures in $\mathbb{R}^d$

In order to construct a linear data structure of n-vertices embedded in $\mathbb{R}^d$ space, a vertex object for each vertex v_i in the set $v_i \in \mathcal{H}^0$ is built. Each vertex object contains a hash of its geometric position in $\mathbb{R}^d$. A common cache $\mathcal{V}$ contains a cache of all vertex objects. The edges can be described by $e_{ij} = v_i v_j \in \mathcal{H}^1$, but is not constructed explicitly. Instead it is defined by an associated (unordered) set $\mathcal{N}_i \subset \mathcal{H}^0$, called the nearest-neighbour set, with each vertex i referring to the same v_i object. The special vertex set can be empty $\mathcal{N}_i = \varnothing$ or non-empty $\mathcal{N}_i = \{v_j, v_k, \dots\}$ and is hashed in code so that by construction $\mathcal{N}_i$ cannot contain its own vertex v_i.

Definition 15. *The* **nearest-neighbour set** $\mathcal{N}_i$ is a unordered set of vertices $\mathcal{N}_i = \{v_j, v_k, \dots\}$ associated with a vertex v_i which contains an element v_j for every edge $e_{ij} = v_iv_j \in \mathcal{H}^1$.

These sets represent all the edges (1-simplices) of the simplicial complex. In total this data structure contains n-vertex objects v_i together with n-sets $\mathcal{N}_i$ which fully described the connectivity of the simplicial complex. It is important to note early that edges $e_{ij} = v_iv_j$ themselves are not stored (for many problems storing the edges is intractable), only an ordered set $\mathcal{N}_i$ for each corresponding vertex v_i. In addition, the simplicial complex can be directed by associating any function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, such as a height function, sampled at every vertex point. Since the number of vertices that can be stored in memory are finite, it is always possible to construct such a function f using sorting algorithms, e.g. for example given n vertices at most the first n^n are needed to which can be assigned to each vertex to direct the edges as desired.

Note that this structure is sufficient to fully represent the connectivity of a simplicial complex $\mathcal{H}^d$, however, it is neither a unique representation nor is the structure described here guaranteed to be a simplicial complex. In the next section algorithms are presented that allow for conversion of conventional vertex-simplex data structures to the new linear structure and the conversion from the linear structure back to conventional structures containing $(d-1)$-simplex containers. In the proceeding section the construction and refinement of a unique simplicial complex in $\mathbb{R}^d$ will be demonstrated. In addition it is shown how, through refinement, (non-)convex subsets of this space can be triangulated. Finally, the computation of the boundary operator is shown, which allows for phase interface detection as well as for for a homology computation.

3.2.2 Triangulation of non-convex, non-continuous domains in $\mathbb{R}^d$ with continuous subdomains of finite size

In order to triangulate an arbitrary non-continuous, bounded domain defined in full generality by

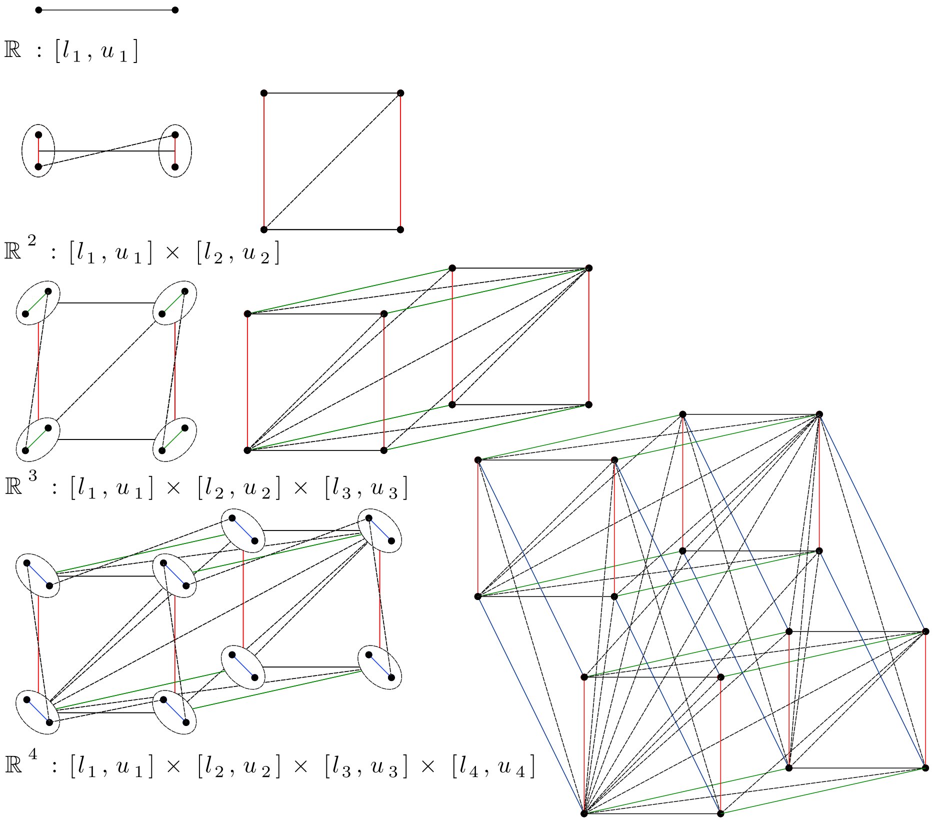
$\mathbb{R} : [l_1, u_1]$
$\mathbb{R}^2 : [l_1, u_1] \times [l_2, u_2]$
$\mathbb{R}^3 : [l_1, u_1] \times [l_2, u_2] \times [l_3, u_3]$
$\mathbb{R}^4 : [l_1, u_1] \times [l_2, u_2] \times [l_3, u_3] \times [l_4, u_4]$

Figure 3.4.: Triangulation of hypercube without explicitly constructing simplices. Starting from an interval on the real line $\mathbb{R}$, take the Cartesian product $[l_1, u_1] \times [l_2, u_2] \in \mathbb{R}^2$ and, in addition, connect every lower vertex generated to every higher vertex to find the triangulation of the 2-dimensional square. Repeating the operations produces a triangulated cube in $\mathbb{R}^3$ and a triangulated 4-dimensional hypercube in $\mathbb{R}^4$.

$$\{\mathbf{x}|\mathbf{x} \in \Omega \subseteq [\mathbf{l}, \mathbf{u}]^n \; = \; [l_1, \; u_1] \; \times \; [l_2, \; u_2] \; \times \; \dots \; \times \; [l_n, \; u_n] \subseteq \mathbb{R}^d, \\ g_i(\boldsymbol{x}) \geq 0, \; \forall i = 1, ..., m\}, \tag{3.2}$$

which starts with triangulating a hypercube Ω as a primary building block. Through the convergence proofs demonstrated in Endres, Sandrock, and Focke (2018), it is which starts with triangulating a hypercubewhich starts with triangulating a hypercubeknown that the full domain can which starts with triangulating a hypercubebe triangulated after sufficient refinement of the simplicial complex, the only assumption is that there is a finite non-empty subdomain in Ω that is Lipschitz smooth. Building a triangulation of Ω is analogous to the Cartesian product of the C_2 cyclic group and is demonstrated visually in Figure 3.4 and an algorithm is provided in Algorithm 1. An algorithm for analytical surface triangulation is provided in Algorithm 2.

Hypercube triangulation and refinement

Algorithm 1 demonstrates the hypercube triangulation described in Section 3.2.2. Figure 3.4 provides a useful visual demonstration of the algorithm. Note that the notation in the initial domainwhich starts with triangulating a hypercube

$$[\mathbf{l}, \mathbf{u}]^n \; = \; [l_1, \; u_1] \; \times \; [l_2, \; u_2] \; \times \; \dots \; \times \; [l_n, \; u_n] \subseteq \mathbb{R}^d,$$

defines a stretched hyperrectangle, a hypercube is defined by setting $l_i = 0 \; \forall i$ and $u_i = 1 \; \forall i$. Additionally, the ordered lists C_l^i and C_u^i are introduced to track the lower and upper vertices of dimension i in analogy to Cartesian products of cyclic groups. The new vertex indices are assigned as they are initiated for the first time. Finally, C_{ul} tracks the list of cross edges.

Note that the approach here differs from many state of the art topological data analysis (TDA) approaches where the sampled points are later triangulated (Delaunay triangulation, Čech and Vietoris-Rips complexes), although the 1-skeletons of such triangulations can be easily converted to the current data structure. In this case, however, the connectivity of the simplicial complex is built into the sampling routine itself, this avoids the need to compute the triangulation later which saves a lot of computational expense in the presence of changing topologies due to, for example, moving solids in a simulation.

Algorithm 1: Triangulate the domain $[\mathbf{l}, \mathbf{u}]^n = [l_1,\ u_1] \times [l_2,\ u_2] \times \ldots \times [l_n,\ u_n] \subseteq \mathbb{R}^d$.

```
Result: All vertices $v_i$ and their nearest neighbour sets $\mathcal{N}_i$ of a triangulated
        $d$-dimensional hypercube ;
Initiate $\mathcal{V} = \varnothing$ ;
Initiate the sets $\mathcal{N}_i = \varnothing$ for all $2^d + 1$ vertices in the complex ;
$v_0 = (l_1, l_2, \ldots l_n)$ ;
$v_{2^d} = (u_1, u_2, \ldots u_n)$ ;
$C_l^0 = [v_0]$ ;
$C_u^0 = [v_2^d]$ ;
$C_{ul} = [\varnothing]$ ;
for $i$ in $d + 1$ do
    for $j$, $(\mathcal{V}_l, \mathcal{V}_u)$ in $(C_l^{0,1,\ldots,i+1}, C_u^{0,1,\ldots,i+1})$ do
        for $k$, $(v_l, v_u)$ in $(\mathcal{V}_l, \mathcal{V}_u)$ do
            $a_{vl} \leftarrow \mathrm{v}_l$ ;
            $a_{vu} \leftarrow \mathrm{v}_u$ ;
            $a_{vl}^{i+1} \leftarrow \mathrm{v}_{2^d}^{i+1}$ ;
            $a_{vu}^{i+1} \leftarrow \mathrm{v}_{2^d}^{i+1}$ ;
            Initiate the new vertices with the $a$ vectors ;
            $v_m \leftarrow a_{vl}$ ;
            $v_n \leftarrow a_{vu}$ ;
            Connect all the new pairs ;
            $\mathcal{N}_i \leftarrow \mathcal{N}_i \cup v_m$ ;
            $\mathcal{N}_i \leftarrow \mathcal{N}_i \cup v_n$ ;
            $\mathcal{N}_{v_l} \leftarrow \mathcal{N}_{v_l} \cup v_m$ ;
            $\mathcal{N}_{v_u} \leftarrow \mathcal{N}_{v_u} \cup v_n$ ;
            $C_l^{i+1} \leftarrow C_l^{i+1} \cup v_m$ ;
            $C_l^{i+1} \leftarrow C_l^{i+1} \cup v_n$ ;
            $C_u^{i+1} \leftarrow C_u^{i+1} \cup v_m$ ;
            $C_u^{i+1} \leftarrow C_u^{i+1} \cup v_n$ ;
            $C_u^j \leftarrow C_u^j \cup v_m$ ;
            $C_u^j \leftarrow C_u^j \cup v_n$ ;
            $C_{ul} \leftarrow C_{ul} \cup (v_l, v_n)$ ;
        end
    end
end
Finally complete all the cross triangulations ;
for $(v_l, v_n)$ in $C_{ul}$ do
    $v_b \leftarrow \mathrm{v}_l$ ;
    $v_a \leftarrow \mathrm{v}_n$ ;
    $v_b^{i+1} \leftarrow \mathrm{v}_{2^d}^{i+1}$ ;
    $v_a^{i+1} \leftarrow \mathrm{v}_{2^d}^{i+1}$ ;
    Initiate the new vertices $v_b$ and $v_a$ and their neighbouring sets $\mathcal{N}_b$ and $\mathcal{N}_a$ ;
    $\mathcal{N}_l \leftarrow \mathcal{N}_l \cup v_a$ ;
    $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup v_b$ ;
    $\mathcal{N}_b \leftarrow \mathcal{N}_a \cup v_a$ ;
    Add new list of cross pairs ;
    $C_{ul} \leftarrow C_{ul} \cup (v_l, v_a)$ ;
    $C_{ul} \leftarrow C_{ul} \cup (v_b, v_a)$ ;
end
```

Algorithm 2: Triangulation of a surface f

```
Result: Compute a triangulation $\mathcal{V}$ of a function $f$ ;
Initiate the set $\mathcal{V}$ ;
Triangulate the domain $\mathcal{V}_{\mathtt{plane}}$ using Algorithm 1;
for $v_i$ in in $\mathcal{V}_{plane}$ do do
    Add the mapped vertex position $f_i = f(v_i)$ to the new complex vertex set:  ;
    $f_k \leftarrow f(v_i)$ ;
    $\mathcal{V} \leftarrow f_k$ ;
    for $v_j$ in $\mathcal{N}_i$ do
        Reconstruct the neighbour sets of $\mathcal{V}_{plane}$ in the projected space:  ;
        $f_j \leftarrow f(v_j)$ ;
        Add the new vertex to the nearest neighbour set $\mathcal{N}_k$ of $f_k$:  ;
        $\mathcal{N}_k \leftarrow f_j$ ;
    end
end
```

3.2.3 The boundary operator

The boundary operator is an immediately useful geometric operation (for example in computing phase interfaces in multi-phase fluid simulations or finding convex sub-domains in optimisation algorithms) which also links the geometry of the simplicial complex with the topology. Computing the boundary of a simplicial complex $\mathcal{H}^d$ or a chain $C(\mathcal{H}^d)$ and in particular the linear span of the boundary set is a sufficient operation to compute the homology groups of a $\mathcal{H}^d$ simplicial complex embedded in $\mathbb{R}^{d+1}$ space which is related by

$$H_k(\mathcal{H}) = \frac{\ker \partial_k}{\operatorname{im} \partial_{k+1}}.$$

However, careful consideration is needed to approximate the span of the operators from the graph structure which is potentially only possible in well-defined simplicial complexes. It is also possible to compute a special type of homology on gradient fields. In Endres, Sandrock, and Focke (2018) it was demonstrated how to use a function associated with the complex to compute the homologies of compact spaces. In Algorithm 3 it is demonstrated that the computation that relies only on simple set operations of cached objects and does not require any additional constructs in out-of-core memory. This also provides a major advantage in fast surface interface detection in fluid dynamics problems in $\mathbb{R}^3$ when large data sets are used to represent fluids.

Algorithm 3: Boundary operator on a chain.

```
Result: Compute the boundary of a chain $C(\mathcal{H}^d) \in \mathcal{V}$ return the set of vertices
        $v_i \in \partial C(\mathcal{H}^0) \subset \mathcal{V}$ of vertices and their nearest-neighbour sets $\mathcal{N}_i$ ;
Initiate the set $\partial C(\mathcal{H}^0) \leftarrow \varnothing$ ;
For every vertex $v_i$ check if it's neighbouring vertices are on the boundary of the
chain ;
for $v_i$ in $\mathcal{V}$ do
    Let $\mathcal{C}_{j,k,\ldots} = \{\mathcal{N}_j, \mathcal{N}_k, \ldots\}$ be all $(d)$-combinations of nearest neighbour sets
      of vertices in $\mathcal{N}_i$ ;
    for $\mathcal{N}_j, \mathcal{N}_k, \ldots$ in $\mathcal{C}_{j,k,\ldots}$ do
        if $\mathcal{N}_j \cap \mathcal{N}_k, \ldots \backslash \{v_i, v_j, v_k, \ldots\} \; \forall l \in \{j, k, \ldots\} = \varnothing$ then
            $\partial C(\mathcal{H}^0) \leftarrow \partial C(\mathcal{H}^0) \cup \{v_j, v_k, \ldots\} \; \forall l \in \{j, k, \ldots\}$ ;
        else
            Continue;
        end
    end
end
Only boundary vertices should have connections, simply remove all unconnected
vertices from the return nearset-neighbour set ;
for $v_i$ in $\mathcal{V}$ do
    for $v_j$ in $\mathcal{N}_i$ do
        if not $v_j \in \partial C(\mathcal{H}^0)$ then
            $\mathcal{N}_i \leftarrow \mathcal{N}_i \backslash v_j$ ;
        end
    end
end
```

3.2.4 Algorithms

Conversion of conventional vertex-simplex to linear vertex-vertex structures

Converting from conventional high memory vertex-simplex to vertex-vertex structures (which are lower memory) when given a set of simplices $\mathcal{S}$ is demonstrated in Algorithm 4.

Algorithm 4: Convert vertex-simplex structures to vertex-vertex structures.

```
Result: All vertices $v_i$ and their nearest neighbour sets $\mathcal{N}_i$ ;
Initiate the sets $\mathcal{N}_i = \varnothing$ for all vertices in the complex ;
Every vertex is connected to every other vertex in a given simplex $s \in \mathcal{S}$, simply
connect all vertices ;
for $s$ in $\mathcal{S}$ do
    for $v_i$ in $\mathcal{S}$ do
        $\mathcal{N}_i \leftarrow \mathcal{N}_i \cup \{v_j | \forall j \in \mathcal{S}\}$ ;
    end
end
```

The *hyperct* library can then be used to compute boundaries and homologies. Converting from vertex-vertex structures back to vertex-simplex structures is more elaborate as shown in Algorithm 5. The superscript s^i is used to donate the entry i in a list or vector simplex container s.

The Algorithm 5 always produces a valid simplicial complex as long as the sets $\mathcal{N}_i$ where constructed from vertex-edge data of a structure that is also a simplicial complex with all vertices in regular position (i.e. there are no degenerate faces in the simplicial complex). This can be demonstrated by a simple proof by contradiction:

Proof. By Definition 6 the only requirement of a simplicial complex is that the each $(k+1)$-element subset of vertices in the (d)-simplices in $\mathcal{H}^d$ is a k-simplex in $\mathcal{H}^k$. By construction all simplices are $(d+1)$-dimensional subsets $\mathcal{H}^d$ and therefore $k \leq d$. Suppose that there is a $(k+1)$-subset in $\mathcal{H}^k$ containing $(k+1)$-vertices $v_i, v_j \dots v_{k+1}$ that is not a simplex. Then the union of sets $\mathcal{N}_i \cup \mathcal{N}_j \cdots \cup \mathcal{N}_{k+1}$ is the empty set. Since at least one vertex $v_k \notin \mathcal{N}_i$ then such a $(k+1)$-simplex cannot be constructed by Algorithm 5 as it contradicts the first *if* statement that $v_k \in \mathcal{N}_i$. □

Algorithm 5: Convert vertex-vertex structures to vertex-simplex structures.

```
Result: All vertices $v_i$ and all $(d+1)$-dimensional simplices $s \in \mathcal{S}_i$ ;
Initiate the sets $\mathcal{S}_i = \varnothing$ for all simplices in the complex ;
Initiate a list, tuple or vector $s$ of size $d+1$ ;
for $v_i$ in $\mathcal{V}$ do
    All simplices built from $v_i$ contain $v_i$;
    $s^0 = v_i$ ;
    For every $v_j \in \mathcal{N}_i$ loop through its nearest neighbours in $\mathcal{N}_j$, for every $v_k \in \mathcal{N}_j$
    that is also in $\mathcal{N}_i$ it is desired to build simplices connecting to the current v;
    for $v_j$ in $\mathcal{N}_i$ do
        The second entry of the current simplex always contains $v_j$;
        $s^1 = v_j$ ;
        Let $l$ be the index used to track simplex entries;
        $l = 1$ ;
        for $v_k$ in $\mathcal{N}_j$ do
            if $v_k \in \mathcal{N}_i$ then
                $l$ += $1$;
                if $l \leq d+1$ then
                    $s^l = v_k$ ;
                else
                    When a simplex is full add it to the set of simplices $\mathcal{S}$ and then
                    start a new simplex $s$ with the first two entries identical to the
                    previous $s$ ;
                    $\mathcal{S} \leftarrow \mathcal{S} \cup s_l$ ;
                    $l = 2$ ;
                    $s^l = v_k$ ;
                end
            end
        end
    end
end
```

3.3 Comparisons to state of the art

3.3.1 On the need for new scalable data structures and benchmark comparisons to state of the art

Simplicial complexes are central mathematical objects in many areas of computational algebraic and combinatorial topology (De Loera, Peterson, and Edward Su, 2002; Meunier, 2006; Musin, 2015) as well as numerical methods, which has more recently found applications in the fields of fluid simulation (Elcott et al., 2007; Mullen et al., 2009; Bell and Hirani, 2012; Misztal et al., 2013; Mohamed, Hirani, and Samtaney, 2016), global optimisation (Paulavičius et al., 2014; Paulavičius and Žilinskas, 2014; Endres, Sandrock, and Focke, 2018) where simplicial complexes are used, for example, as a back-end for finding optimal neural network architectures (Li et al., 2021). Simplicial complex structures also have many prominent applications in topological data analysis (TDA) (Edelsbrunner, Letscher, and Zomorodian, 2000) and computational geometry applications such as manifold learning (Freedman, 2002). Many practical applications of TDA has been found in machine learning (Carrière et al., 2020; Pun, Xia, and Lee, 2018; Hofer et al., 2017; Hofer et al., 2019; Hu et al., 2019; Khrulkov and Oseledets, 2018; Kwitt et al., 2015; Moor et al., 2020; Ramamurthy, Varshney, and Mody, 2019; Rieck et al., 2018; Rieck, Bock, and Borgwardt, 2019; Zhao and Wang, 2019; Royer et al., 2021), materials science and chemistry (Lee et al., 2018; Townsend et al., 2020), in multiphase computation fluid dynamics (Misztal et al., 2013), medical applications such as MRI (Rieck et al., 2020; Fadnavis et al., 2021) and big data analysis in health care (Joshi and Joshi, 2019). Simplicial complexes are commonly obtained through an operation called a triangulation which acts on the underlying space which in turn can be either continuous (through sampling) or discrete (such as point clouds and graph networks). There are many different types of (abstract) simplicial complexes that have been defined based on their triangulation methods and homologies, for example the Delaunay triangulation, Čech complex, the Vietoris-Rips complex and the Witness complex (Silva and Carlsson, 2004; Edelsbrunner and Harer, 2010). A more recent work by Boissonnat and Maria (2014) introduced the Simplex Tree structure which allows for the representation of abstract simplicial complexes of any dimension. While such data structures greatly enhance the tractability of higher dimensional simplicial complexes, any data structure that relies on the explicit construction of simplices remains a fundamental bottleneck for many potential applications in TDA (Chazal,

2017) as well as for triangulations of continuous spaces. As an example, in triangulating the unit cube or hypercube, which is among the simplest and the most important objects in d-dimensional Euclidean space, it is known that the best lower bound number of simplices required in any minimum triangulation grows combinatorially with dimension d as $O(d!)$ (Smith, 2000; Zong, 2005). The hypercube is particularly useful to triangulate since its refinement in a bounded domain can be used to triangulate any compact domain (Endres, Sandrock, and Focke, 2018) which can provide a deeper understanding of the underlying hypersurface.

In Section 3.2 the simplest possible data structure was demonstrated, which still stores only the connectivity of the simplicial complex. This both triangulates domains and can represent 1-skeletons of a triangulation without the need to explicitly construct any simplex containers. In addition, it was shown that many important geometric and topological operations are available with this data structure. In particular, the boundary computation operators ∂_k on this structure which connects the set operation of chain algebras (topology) with the geometry represented by the simplicial complex (Henle, 1979, p. 158) in any dimension was shown. An algorithm is presented that can triangulate and refine d-dimensional hypercubes allowing for the triangulation of arbitrary continuous domains in $\mathbb{R}^d$ through stretching and refinement. Such an operation is useful for triangulating non-continuous embeddings in $\mathbb{R}^d$ which is useful in applications such as three-phase computational fluid dynamics (CFD) simulations and the out-of-core boundary computation can be used to find interfaces efficiently on large datasets. A rudimentary python implementation is made available in the $hyperct$ library[1] which also allows for the conversion to traditional simplicial complex structures such as the Delaunay triangulation available in the QHull library (Barber, Dobkin, and Huhdanpaa, 1996) as well as the Alpha and Vietoris-Rips complexes (Boissonnat and Maria, 2014).

3.3.2 Benchmark comparisons to state-of-the-art

Here a simplicial complex $\mathcal{H}^k$ embedded in $\mathbb{R}^d$ space is considered. The formal definition of the underlying concepts k-simplices, k-chains and an (abstract) simplicial complex together with its homology is defined in Section 3.2. An intuitive description of a simplicial complex is a set of k-simplices as building blocks shown in Figure 3.3A subject to formal criteria defined in Section 3.2. A k-chain of k-simplices (Figure 3.3B) is also denoted as $C_k(\mathcal{H})$.

[1] Released under MIT license, available at: https://github.com/Stefan-Endres/hyperct

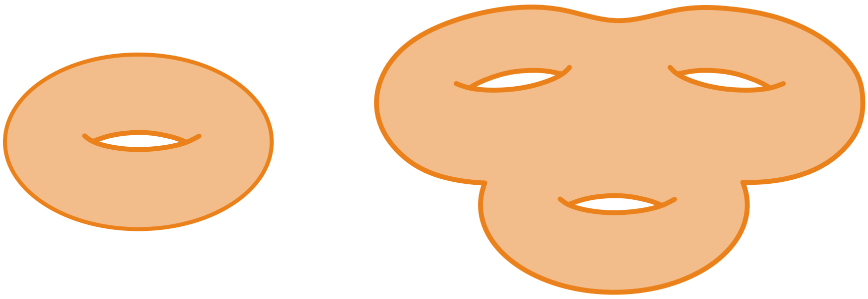

Figure 3.5.: A 1-connected torus (left) and a 3-connected torus (right) embedded in $\mathbb{R}^3$.

Such chains can additionally be directed such as demonstrated in Figure 3.3C. As discussed in Section 3.2, equipping an algebra to these chain groups can be used to compute homology groups useful for deducing the topological properties of the underlying data. Note that the dimension k need not be equal to d and can be both lower $k < d$ than underlying embedded space d (such as when triangulating a hypersurface embedded in $\mathbb{R}^d$) or higher than the underlying dimension $k > d$ (for example common when computing persistent homologies on point clouds). In this work n-dimensional points represented as vertices v_i (or $0-$simplices) are considered with the only requirement that every point v_i is embedded in the same dimension. This is commonly obtained through operations such as sampling (of a hypersurface, or from physical space and time series measurements), triangulation (of a continuous or non-contuous space) or labelling of non-numerical data projected into the same dimension.

In this section the application of the linear simplicial complex structure is demonstrated with the $hyperct$ library implementation in triangulating the n-tori embedded in $\mathbb{R}^d$ and compare the resulting memory consumption and number of data containers needed. A n-Torus is a well understood topological surface containing n holes (cf. Figure 3.5), the minimum 1-Torus allows for a well defined benchmark to compare the memory consumption of different simplicial complex triangulations. The comparison was done against the Alpha complex and Vietoris-Rips complexes implemented in the $gudhi$ library (Maria et al., 2014), a popular library for topological data analysis (TDA), the flag complex from the FLAGSER library (Lütgehetmann et al., 2020) which is based on the RIPSER library by Bauer, 2021, as well as the Delaunay complex computed using the QHull library (Barber, Dobkin, and Huhdanpaa, 1996).

3.3.3 Case study: 1-Tori in $\mathbb{R}^d$

A minimum triangulation of the 1-Torus in $\mathbb{R}^d$ can be constructed trivially by building the hypercube Ω in $\mathbb{R}^{d-1}$, identifying all vertices and finally gluing all geometrically opposing faces. In Figure 3.6 the results are shown for the number of constructed data containers or simplices required for a minimum triangulation and in Figure 3.7 the results comparing the total bytes of memory consumed are shown. A direct memory comparison between a Python-based HyperCT and C/C++ optimised implementations of QHull is not feasible due to implementation differences between the two libraries. Even though a Pythonic API might be available for QHull, the memory profiling is not possible due to the I/O differences that occur when interfacing the libraries. Therefore the number of data containers used by the structure is used as a comparison instead in Figure 3.7 and only QHull is benchmarked in Figure 3.6. The results demonstrate the large improvement in the tractability of triangulating surfaces using the linear simplicial complex data structure. The new data structure requires less computer memory than the other complexes in the higher dimensions implying greater tractability in higher dimensional spaces.

3.3.4 Case study: Comparison to flag complex on undirected graphs

The flag complex is the simplest structure that is comparable to the structure presented here in the sense that it can be used to represent higher dimensional simplicial complexes as well as compute their homologies. Table 3.1 presents a comparison to the flag complex structure and the difference in cardinalities required to represent data in a simplicial complex. While the $RIPSER$ and $FLAGSER$ libraries are highly efficient and mature libraries capable of very fast computations, the comparison highlights the potential of $hyperct$ to model a very large amount of data. For example Lütgehetmann et al., 2020 studies directed graphs that model brain microcircuitry reconstructions. As a comparison the C. Elegans (cf. Figure 3.8) dataset (an undirected graph) is used, as recorded in the associated repository (Appendix C) $hyperct$ required an estimated 0.227184 MiB compared to 589 MiB for $FLAGSER$ to represent the complex and compute the boundaries. Computing homologies in $FLAGSER$ is far faster and more efficient, however, the potential to develop efficient algorithms on the sparser structure should be

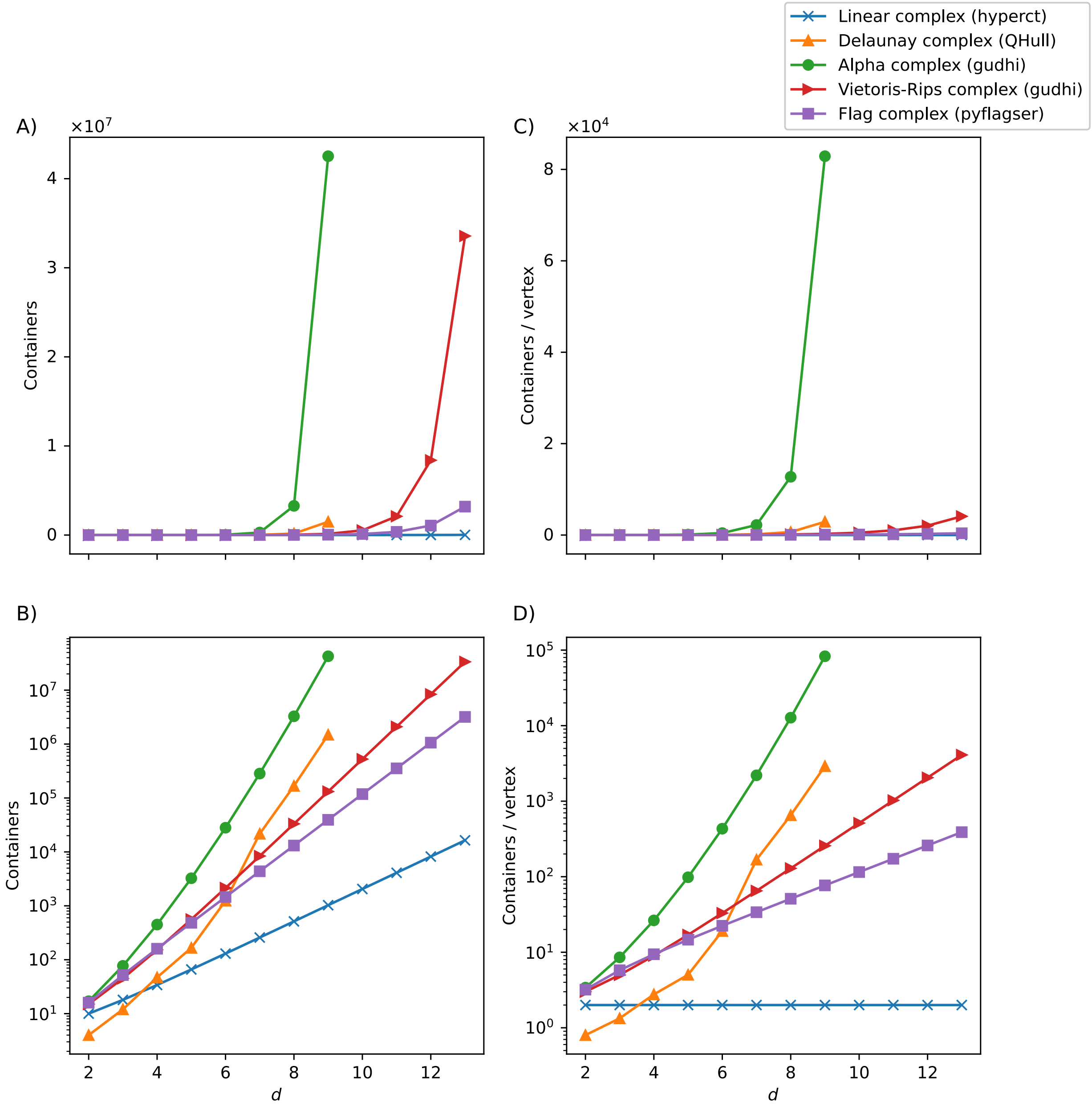

Figure 3.6.: Comparison of simplex containers (Delauney complex) and data containers (Linear complex) needed for a minimum triangulation of 1-Tori embedded in $\mathbb{R}^{d-1}$. A) It can be observed that the new linear complex data structure requires far less memory than the Delaunay complex structure in higher dimensions. B) The semi-log plot of the same data as (A) is shown for greater clarity, in this plot it can be seen that the Delaunay complex requires less memory in the lower dimensions which is expected theoretically since a far smaller number of simplices suffice to triangulate the hypercube in lower dimensions. C) contains the same data sets as (A) normalised to the number of vertex points and D) is a semi-log plot of the same normalised data as (C) for clarity.

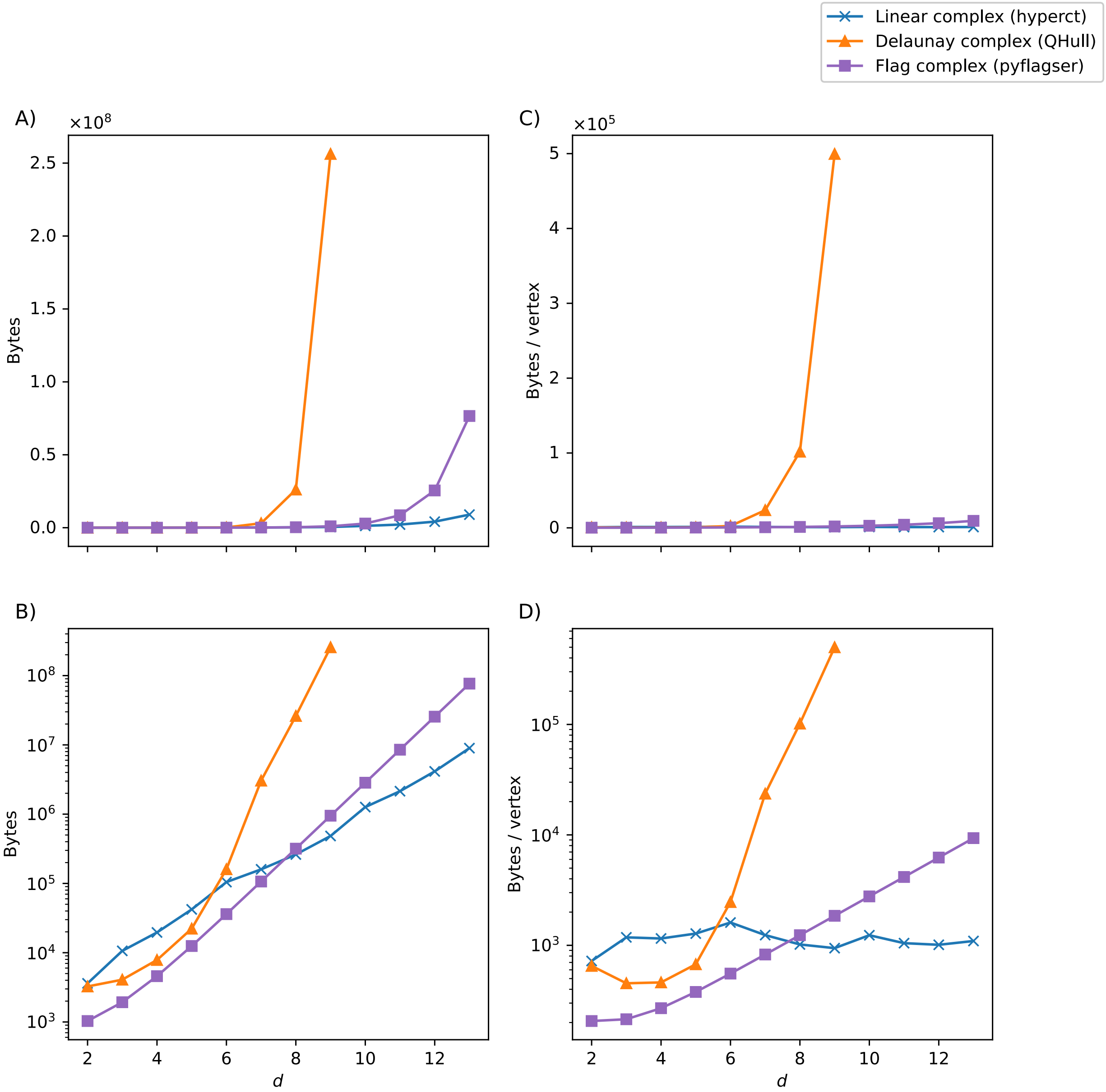

Figure 3.7.: Comparison of the computer memory needed for a minimum triangulation of 1-Tori embedded in $\mathbb{R}^{d-1}$. A) It can be observed that the new linear complex data structure requires far less data containers than the other data structures in higher dimensions. B) The semi-log plot of the same data as (A) is shown for greater clarity, in this plot it can be seen that the Delaunay complex requires less data containers in the lower dimensions which is expected theoretically since a far smaller number of simplices needed to triangulate the hypercube in lower dimensions. C) contains the same data sets as (A) normalised to the number of vertex points and D) is a semi-log plot of the same normalised data as (C) for clarity.

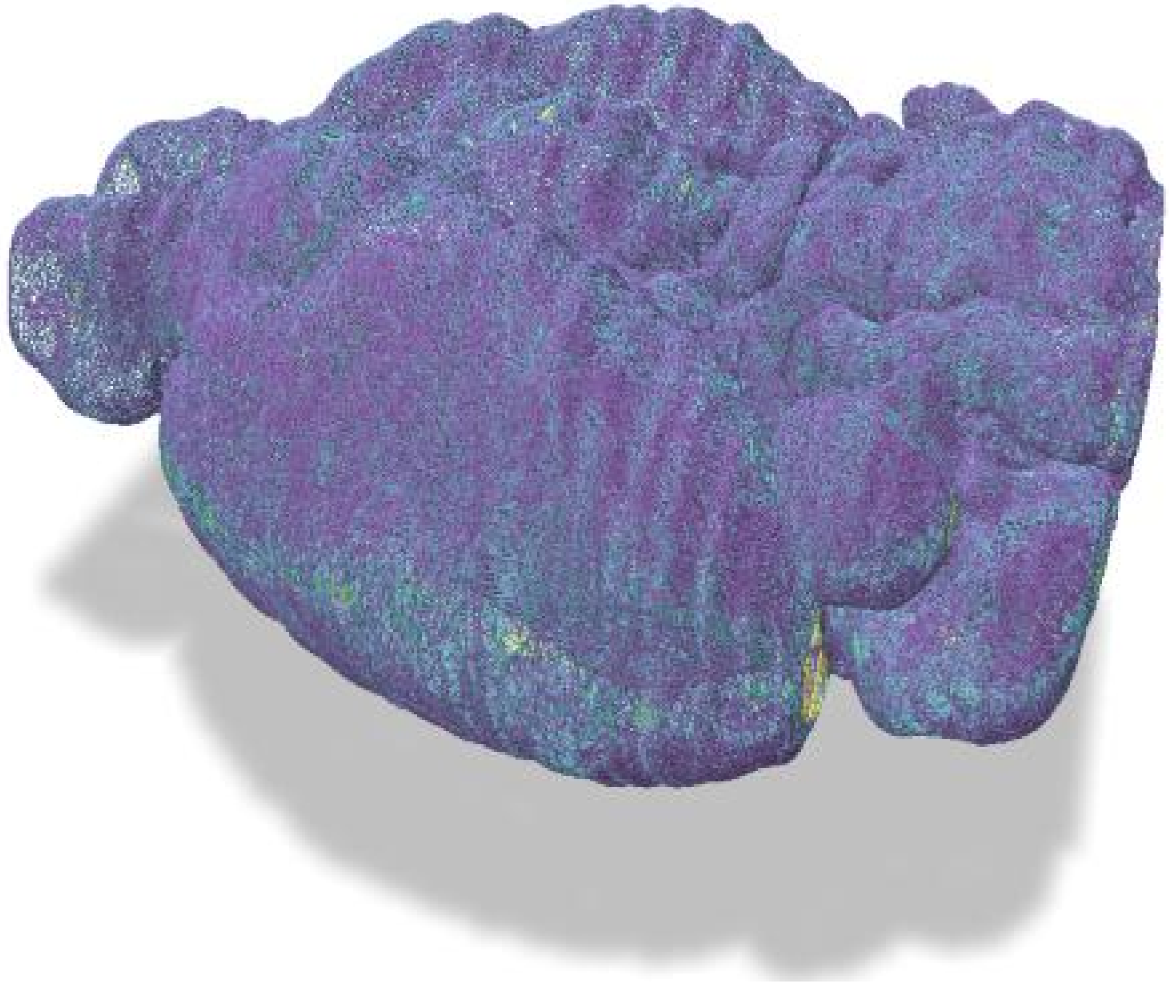

Figure 3.8.: The nervous system in a typical Caenorhabditis elegans hermaphrodite (neural) network that was used to derive the unconnected graph in the KONECT dataset (Watts and Strogatz, 1998; *Caenorhabditis elegans (neural) network dataset – KONECT* 2018).

highlighted with this result.

3.3.5 Applications for surfaces embedded in $\mathbb{R}^3$

Microsctructure analysis in multiphase CFD simulations

In CFD applications where the changing microstructure is important for modelling materials property and performance changes, simulations present a data challenge due to the large number of discrete points involved. For example, Figure 3.9 which contains 73 198 658 discretised fluid elements representing a boundary compuation on fluid films above 600000 solid-phase

	hyperct	Flag complex (FLAGSER)
Structure	$V = \{v_1, v_2, v_3, \dots\}$ $NN = \{\mathcal{N}_i, \mathcal{N}_i, \mathcal{N}_i, \dots\}$	$V = \{v_1, v_2, v_3, \dots\}$ $E = \{(v_i, v_j), (v_l, v_;), \dots\}$
Cardinality (undirected)	$2\|V\|$	$\|V\| + \|E\|$
Cardinality (directed)	$3\|V\|$	$\|V\| + \|E\|$

Table 3.1.: Comparison of linear- and flag complex representations.

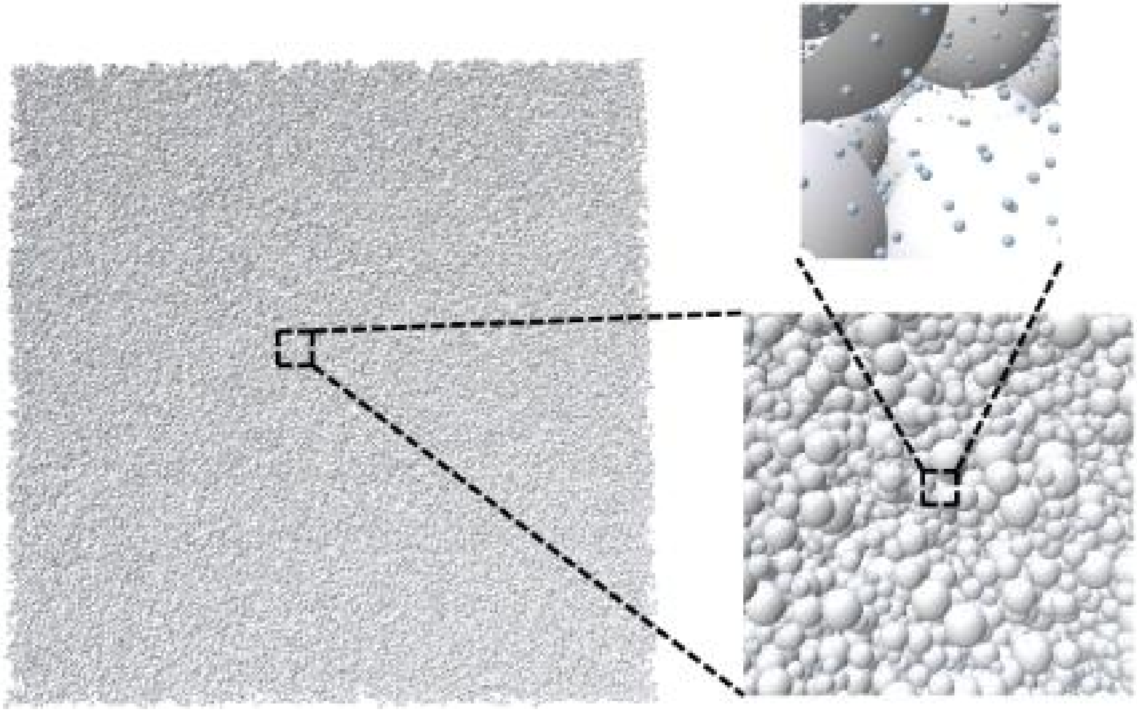

Figure 3.9.: Fluid elements generated on a particle film discertisation of 600000 solid nanopartilces.

nanoparticles. An explicit simplicial complex on data structures this large present a problem both in the memory required to contain all the simplices, but more importantly a limitation on the in-core computations for algorithms that requiring traversing the simplex structures of the mesh. The hyperct library allows for an out-of-core mesh computation which allows for the fast detection of merging fluid clusters as well as their interface detection through boundary computation. Comparable solutions such as the half-edge mesh structure cannot be used to represent both the 3-dimensional simplicial complex (or bulk fluid) and its boundary (surface interface) on the same data structure. Computing boundaries on the same mesh can also be used to detect topological properties that are important for practical applications such as the number of different percolation networks and their tree structure, the number of gas bubbles trapped in the film, the number of different porous networks and diffusion pathways pathways and the birth and death of merging liquid bridges over time.

Refinement and boundary computation in $\mathbb{R}^3$

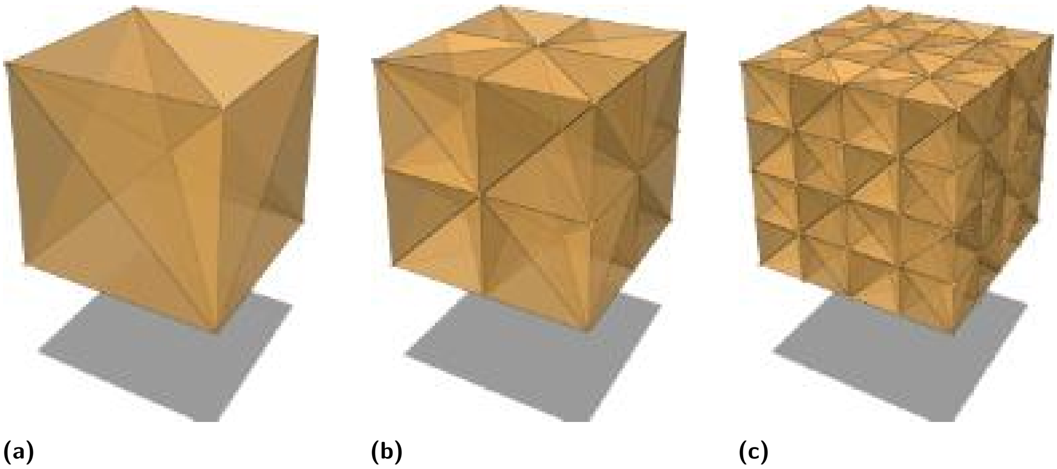

Figure 3.10.: Triangulated cubes with (a) no refinement (b) one generation of refinement and (c) with two generations of refinement. Simplices are slightly transparent to show interior triangulation.

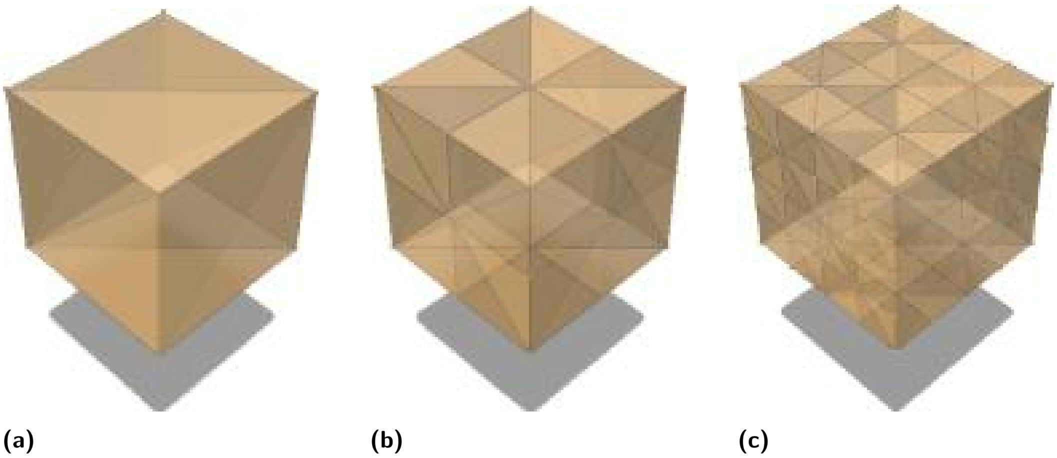

Figure 3.11.: Boundary computation of Triangulated cubes with (a) no refinement (b) one generation of refinement and (c) with two generations of refinement. Notice that all vertices in the interior have been removed by the boundary computation.

In order to demonstrate both the refinement and the boundary operation, the operations on a 3-connected Tori embedded in $\mathbb{R}^3$ is shown visually. The computations are useful in applications such as manifold learning and in multi-phase computational fluid dynamics where computing the curvature on the surface interfaces requires a boundary computation step (Endres, Ciacchi, and Mädler, 2021). The nearest-neighbourhood containers $\mathcal{N}_i$ allow for a fast local interface detection using only simple set operations which is advantageous when large data sets are used and out-of-core local processing is required.

Refinement involves simply repeating the triangulation at different origin $v_0 = (l_1, l_2, \dots l_n)$ and supremum vectors $v_{2^d} = (u_1, u_2, \dots u_n)$ or refining all star domains as described in the library source code. The former is faster at a higher cost of memory while the latter is slower, but requires no additional memory containers. Figure 3.10 demonstrates the triangulation and refinement of a cube embedded in $\mathbb{R}^3$.

Surface detection and refinement in $\mathbb{R}^3$

Here it is demonstrated that the triangulation and boundary computation of a 3-Torus. The basic principle can be demonstrated by, for example, computing the boundary $\partial(\mathcal{H})$ of the triangulated cubes $\mathcal{H}$ in Figure 3.10 as shown in Figure 3.11. Note that all vertices in the interior have been removed and only the boundary vertices remain in Figure 3.11, for an arbitrary (non-)convex domain (the set defined in Equation 3.2 repeated here for convenience):

$$\begin{aligned}\{\mathbf{x}|\mathbf{x} \in \Omega \subseteq [\mathbf{l}, \mathbf{u}]^n \; &= \; [l_1,\ u_1] \ \times \ [l_2,\ u_2] \ \times \ \dots \ \times \ [l_n,\ u_n] \subseteq \mathbb{R}^d, \\ & g_i(\boldsymbol{x}) \geq 0,\ \forall i = 1, ..., m\},\end{aligned}$$

and this domain can be triangulated by refining and removing any vertices that fall outside the constraint oracle functions $g_i(\boldsymbol{x})$. In this example the domain $[-1,\ 2] \times [-1,\ 2] \times [-0.5,\ 0.5]$ was used and

$$\begin{aligned}g_1 &= (x_1^2 + x_2^2 + x_3^2 - (a^2 + b^2))^2 - 4a^2(b^2 - x_3^2) \\ g_2 &= ((x_1 - 1)^2 + x_2^2 + x_3^2 - (a^2 + b^2))^2 - 4a^2(b^2 - x_3^2) \\ g_3 &= (x_1^2 + (x_2 - 1)^2 + x_3^2 - (a^2 + b^2))^2 - 4a^2(b^2 - x_3^2),\end{aligned}$$

where $a = 0.5$ and $b = 0.3$ are the major and minor radii of of each torus. Next, simply compact the equations to find the desired set

$$g(\boldsymbol{x}) = (g_1 \geq 0) \vee (g_2 \geq 0) \vee (g_3 \geq 0).$$

Once the 3-dimensional simplicial complex $\mathcal{H}$ has been recovered by triangulating and refining the set within Equation 3.2, the boundary of the entire complex $\partial(\mathcal{H})$ can be computed. The result is the 2-dimensional surface of a 3-torus embedded in $\mathbb{R}^3$ shown in Figure 3.13.

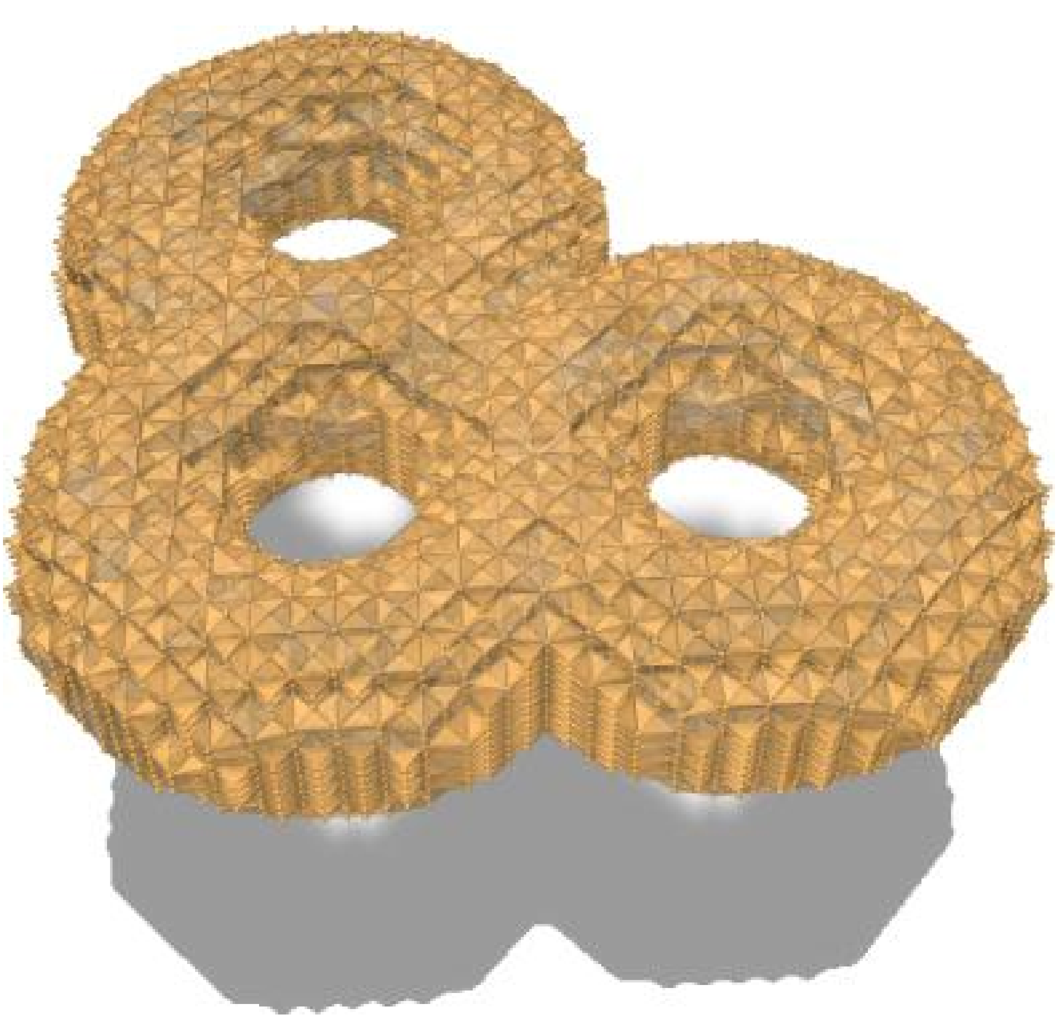

Figure 3.12.: Boundary computation $\partial(\mathcal{H})$ of a triangulated 3-torus embedded in $\mathbb{R}^3$.

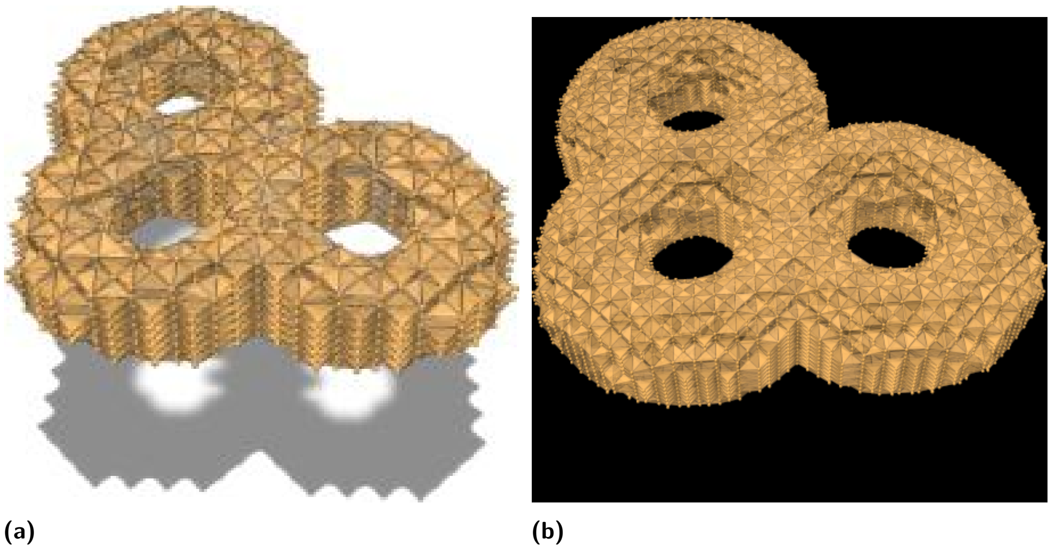

Figure 3.13.: Boundary computation $\partial(\mathcal{H})$ of a triangulated 3-torus embedded in $\mathbb{R}^3$ with (a) fewer refinements and vertices and (b) more refinements and vertices. Note that the topology is preserved in (a).

It is also possible to keep refining or to use fewer refinements in this test case as shown in Figure 3.13. Computing homologies as described in the main body provides a qualitative indicator that can be computed to help track the progress of the triangulation in understanding the underlying space in any dimension. As demonstrated in Figure 3.14 and in the accompanying notebooks there are no vertices or simpilces remaining in the interior of the simplical complex after the boundary computation.

Homologies of a 1-connected torus in $\mathbb{R}^3$

Figure 3.15 demonstrates the application of the boundary operator to find closed loops and homology ranks of a simple torus surface $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ where $f(u, v)$ is the parametric equation of the torus Equation 3.1. Additionally, supplying the surface with an arbitrary gradient vector field $v : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ in this case $v(x_1, x_2, x_3) = x_1$ allows for the constructed simplical complex to be directed. While this homology represents the computation of gradient field homology of f it should be noted that there many other types of homologies that can be computed using the domain structure of $\mathcal{H}$ itself.

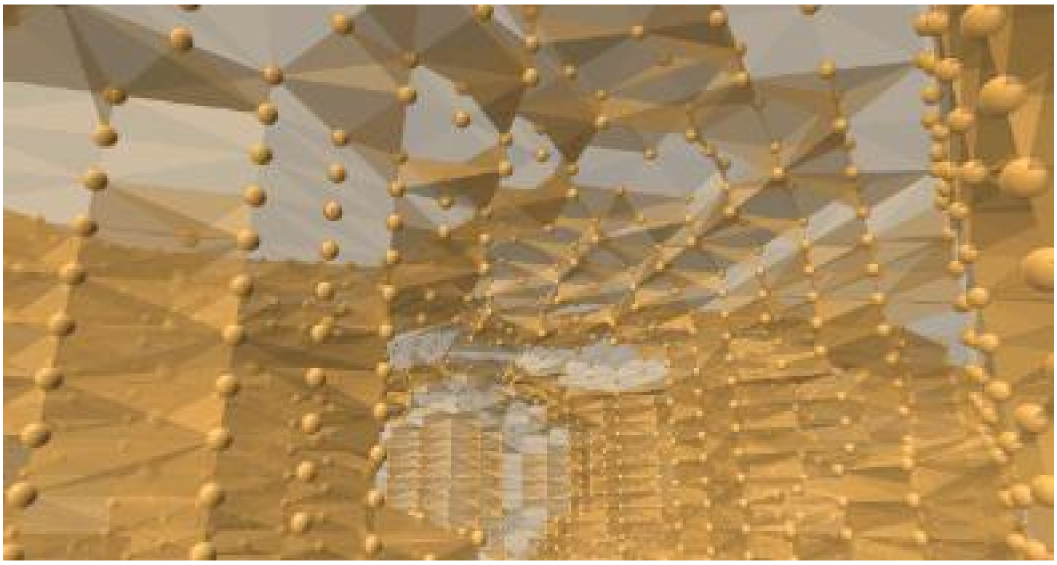

Figure 3.14.: Interior of the torus in Figure 3.13. with boundary computation $\partial(\mathcal{H})$ of a triangulated 3-torus embedded in $\mathbb{R}^3$.

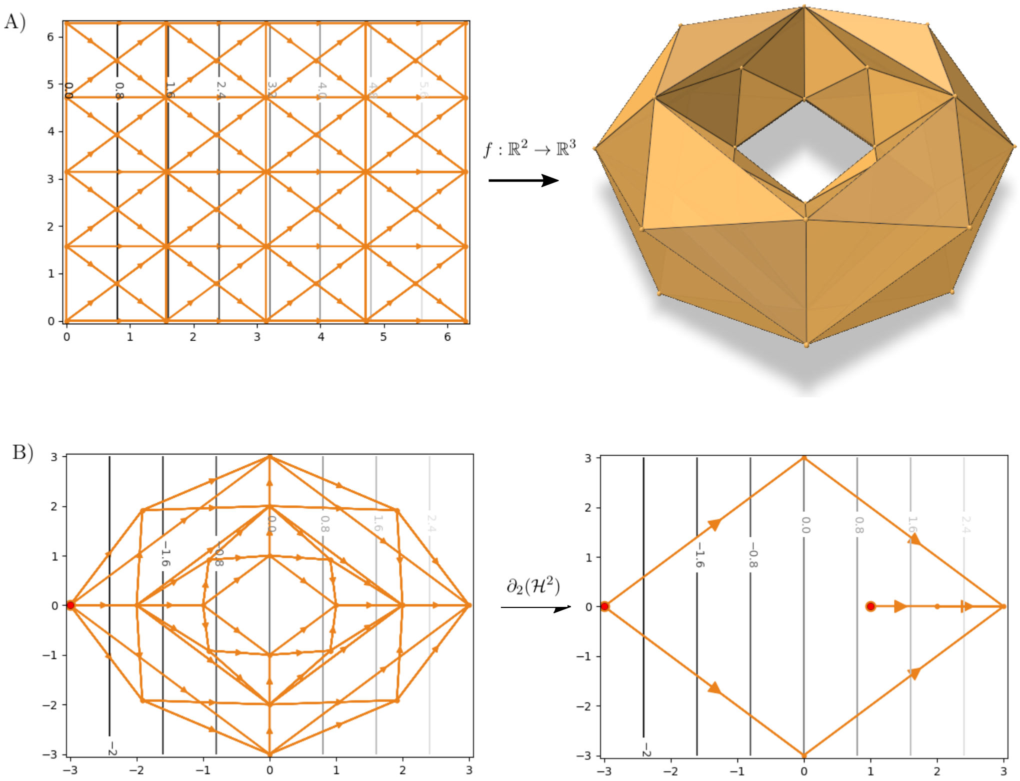
A)
$f : \mathbb{R}^2 \to \mathbb{R}^3$
B)
$\partial_2(\mathcal{H}^2)$
0.0
0.8
1.6
2.4
3.2
4.0
4.8
5.6
−2
−1.6
−0.8
0.0
0.8
1.6
2.4

Figure 3.15.: A) A mapping of a triangulated plane torus $[0, 2\pi] \times [0, 2\pi] \in \mathbb{R}^2$ to a surface embedded in $\mathbb{R}^3$. Supplying (any) gradient field additionally allows for the structure to be directed. B) Top down view of the edges of the torus in embedded in $\mathbb{R}^3$. The boundary computation on the directed complex results in two closed paths, the number of paths in this case can be detected immediately (red dots) due to two local minima in the gradient field resulting from the boundary computation (right).

3.4 Derivation of discrete curvatures, contact angles and accuracy estimates

The machinery is now in place to derive the core results of this dissertation which will prove the second result outlined in H_2. The simplicial complex representation $\mathcal{H}$ of the surface $\mathcal{S}$ defined in this chapter (Definition 6 and Definition 8 respectively) will be used to relate vertices $v_i \in \mathcal{H}$ which have corresponding position on a surface $\mathbf{f_i} = f(v_i) \in \mathbb{R}^3$. The vertex cache of $\mathcal{H}$ is called $\mathcal{V}$ and the boundary of $\mathcal{V}$ and $\mathcal{S}$ is denoted by $\partial\mathcal{V}$ and $\partial\mathcal{S}$ respectively throughout.

A lot of the results in this section, which proves that the discretised surface can exactly represent the curvature under conditions of sufficient refinement, are based on the powerful Gauss-Bonnet and Stokes' theorems Theorem 1 and Theorem 2. The key insight is that the Gauss-Bonnet Theorem can always be applied trivially to a locally bounded disk which always has a known Euler characteristic $\chi = 1$.

Interior mean normal curvature

For reasons outlined in Section 3.2 only a simplicial complex is used in this section (with the understanding that this applies to all computational geometric data structures that are manifold). In Figure 3.16, the essential definitions used in this section are shown. Every simplicial complex has a dual complex, in this case the circumcentric dual shown in Figure 3.16 in the darker shade. The notation and proofs used in this section closely follow the textbook by Crane (2018).

The notation of Figure 3.16 has the following meaning:

- $\boldsymbol{f_i}$ - position of surface vertex
- ℓ_{ij} - length of edge ij
- A_{ijk} - area of triangle ijk
- C_{ijk} - area of the dual of the triangle ijk

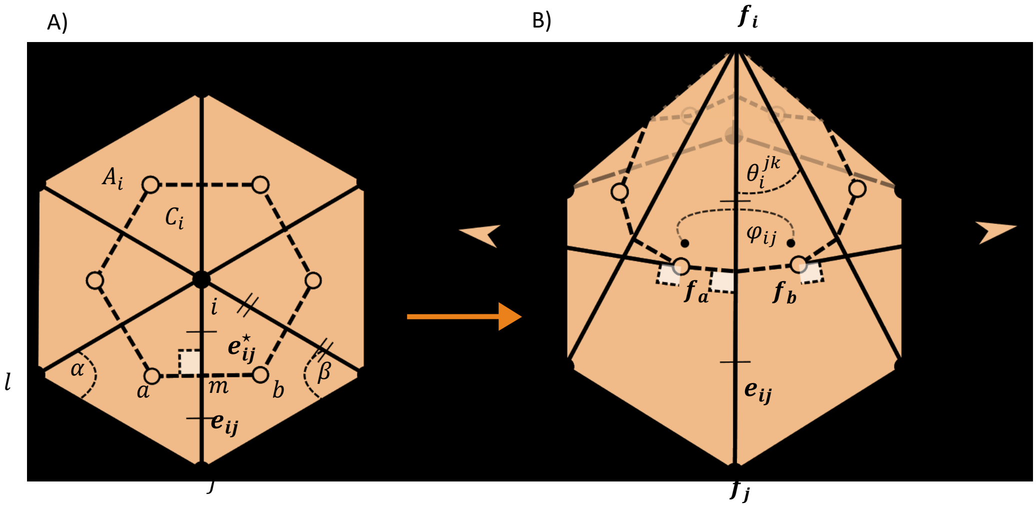

Figure 3.16.: A) The simplicial complex of a triangulation in the plane together with its circumcentric dual complex. The star domain $st(v_i)$ is the set of all simplices neighbouring a vertex v_i or i in simplified notation, $st(v_i)$ has a total area A_i. The circumcentric dual complex produces one (dual) vertex per triangle, here called a and b. The shaded area is the dual area C_i of the circumcentric dual cells with the dual vertices a and b associated with each edge $\mathbf{e_{ij}}$ with midpoint m, and corresponding interior angles α and β. Every edge $\mathbf{e_{ij}}$ has a corresponding unique perpendicular dual edge $\mathbf{e^{\star}_{ij}}$. B) The surface function f maps vertices from the plane in $\mathbb{R}^2$ to the corresponding vertices in $\mathbb{R}^3$ as discussed at the start of the Chapter, the star domain $st\left(\mathbf{f_i}\right)$ is the set of all simplices neighbouring $\mathbf{f_i}$. Each edge $\mathbf{e_{ij}}$ has two normal vectors $\mathbf{N_a}$ and $\mathbf{N_b}$ associated with its dual.

- $\boldsymbol{N}_{ijk}$ - unit normal of triangle ijk

- θ_i^{jk} - interior angle at vertex v_i of triangle ijk

- φ_{ij} - dihedral angle at oriented edge ij

Finally, it is possible to derive a discrete normal curvature $\widehat{H}$. Stokes' Theorem (Theorem 2) also allows for the definition of the concept of a discrete differential by integrating a 1 -form $\boldsymbol{\alpha}^1$ along an edge $\boldsymbol{e}_{ij}$ formed by a line bounded by two vertices from $\mathrm{f_i}$ to $\mathrm{f_j}$:

$$\widehat{\boldsymbol{\alpha}}_{eij} = \int_{\boldsymbol{e_{ij}}} \boldsymbol{\alpha}^1,$$

this integral can be understood as sampling n points along edge and adding the sum of projections along the vector $\boldsymbol{e_{ij}}$ to

$$\int_{\boldsymbol{e_{ij}}} \boldsymbol{\alpha}^1 \approx |\boldsymbol{e}| \left(\frac{1}{n} \sum_{i=1}^{N} \boldsymbol{\alpha}^1_{\boldsymbol{p_i}}(\boldsymbol{e}/|\boldsymbol{e}|) \right)$$

However, a key fact about the *position* vectors $\mathrm{f_i}$ is that the position on every point of the edge $\boldsymbol{e}_{ij}$ does not need to be known in order to compute the total *integrated* change of the one-form df acrosss the edge (through FTC and Stokes' itself applied in 1D). Therefore the integrated change in position is known exactly:

$$\int_{\boldsymbol{e}_{ij}} df = \mathrm{f_i} - \mathrm{f_j} \tag{3.3}$$

By integrating the smooth $H\mathbf{N}$ in a star domain around the dual area of a point f_i on the surface is found:

$$\begin{aligned} \widehat{H\boldsymbol{N}dA}_i &= \int_C H\boldsymbol{N}dA = \int_C df \wedge d\boldsymbol{N} = \int_C d\boldsymbol{N} \wedge df = \int_C d(\boldsymbol{N} \wedge df) \\ &= \int_{\partial C} \boldsymbol{N} \wedge df = \sum_j \int_{\boldsymbol{e}^\star_{ij}} \boldsymbol{N} \wedge df \\ &= \sum_j \boldsymbol{N_a} \times (m-a) + \boldsymbol{N_b} \times (b-m) \end{aligned} \tag{3.4}$$

where Stokes' Theorem was used in the fourth step and Equation 3.3 is used to compute the integral of the dual edges $\boldsymbol{e}^\star_{ij}$ resulting in $(m-a)$ and $(b-m)$ crossed with the respective normal vectors. Here the cross product replaced he wedge product $\wedge$ of the differential forms df and $d\mathbf{N}$ which is possible due to working with 2D simplices embedded $\mathbb{R}^3$. This demonstrates one method that can be used to compute the integrated curvature exactly. Note, however, that the ratio between the primary and dual length can also be expressed through the Cotan formula, therefore taking the ratio as a scalar multiple of the original edge $\frac{|\boldsymbol{e}^{\star}_{ij}|}{|\boldsymbol{e}_{ij}|}\boldsymbol{e}_{ij} = \frac{|\boldsymbol{e}^{\star}_{ij}|}{|\boldsymbol{e}_{ij}|}(\boldsymbol{f_i} - \boldsymbol{f_j})$ results in:

$$\widehat{H\boldsymbol{N}\boldsymbol{d}A}_i = \frac{1}{2}\sum_{j\in st(\boldsymbol{f_i})} (\cot\alpha_{ij} + \cot\beta_{ij})\,(\boldsymbol{f_i} - \boldsymbol{f_j}) \tag{3.5}$$

Thus the length of the original primal edges in Equation 3.5 can be used to compute the same integral as the dual area in Equation 3.4. The discretised definition of this integral is the Cotan formula (Pinkall and Polthier, 1993; Dyer and Schaefer, 2009). It is important to note that $\widehat{H\mathbf{N}\mathbf{d}A_i}$ is an integrated quantity, to recover the point-wise curvature at a vertex f_i it must be normalised with its dual area C_{ijk} that was integrated over:

$$\widehat{H_i} = \widehat{H\boldsymbol{N}\boldsymbol{d}A}_i/C_{ijk} = \frac{1}{2}\sum_{j\in st(\boldsymbol{f_i})} (\cot\alpha_{ij} + \cot\beta_{ij})\,(\boldsymbol{f_i} - \boldsymbol{f_j})\,/C_{ijk} \tag{3.6}$$

As a consequence of Stokes' Theorem applied to a constant curvature surface, the integrated area is exact in the dual cell of a minimised surface (Figure 3.16). The integrated quantity $\overline{H\ \mathrm{N}}\mathrm{dA}_i$, termed a discrete one-form, is integrated over the dual cell rather than the full area, as demonstrated in Figure 3.16, due to the use of dual cells in the derivation of Equation 3.5.

The cotangent formula presented in Equation 3.5 is extensively documented in literature for simulations involving curved surfaces, both in physical finite element method (FEM) simulations (Nguyen, Karčiauskas, and Peters, 2014) and in modern discontinuous Galerkin (DDG) interpretations (Grinspun et al., 2006; Ziegler et al., 2008). However, the approximation and physical interpretation of the geodesic boundary curvature remain less established. For instance, popular discretisations based on interior angles suffer from the drawback that, due to intricacies in how the cotangent formulation integrates over the dual area C_i instead of the total area A_i, the mean normal curvature on the boundaries is often disregarded in many applications, as exemplified in Figure 3.17.

The cotan formula has the important property that it retains the correct value for $\widehat{H\boldsymbol{N}\boldsymbol{d}A}_i$ under any refinement at the equilibrium geometry. The point-wise value $\widehat{H_i}$ also retains the correct value when provided that the interface is convex, the simplicial complex is symmetric and the curvature of the surface is constant. The Gauss-Bonnet Theorem establishes the validity of these relationships for any surface, with or without a boundary. For a given refinement on an imperfect surface, the vertex's local star domain simply represents a surface with a boundary. In essence, we demand that the physics computed from the curvature be accurately represented, even at the expense of introducing numerical uncertainty over time. In regions where the uncertainty is significant, the mesh or triangulation can be refined before taking a time step in a full simulation.

It is of interest to know if other formulations from the field of DDG also have such promising rigorous properties. Since they are generally trying to describe point-wise *approximations* (cf. Table C.1) as opposed to exact integrated quantities it turns out, as elaborated on in Appendix C, that the Cotan formula is currently unique in allowing for exact quantities to be computed. This means that unfortunately there is currently no exact discretely integrated Gaussian curvature K known from literature, Gaussian curvature is essential for relating the discrete mesh to smooth three-phase contact angles on the boundaries of the surface as discussed in the next part of this Section.

Boundary gas-liquid-solid interface contact angles

The next step is derive novel notions of discrete geodesic curvature (defined for the smooth version in Definition 4) and the concept of a discrete contact angle in relation to the true smooth contact angle. As a first step it should be noted that the Gauss Theorem in Equation 2.15 applies to any surface, including piece-wise linear manifolds, which allows for the definition of a discrete Gauss-Bonnet Theorem by simply summing up the integrals of local patches of piece-wise linear triangles surrounding every vertex in the cache:

Theorem 4. ***Discrete Gauss-Bonnet Theorem:***

$$\sum_{i \in int(\mathcal{V})} \widehat{K_i} + \sum_{i \in \partial \mathcal{V}} \widehat{k}_{g,i} \;= 2\pi\chi. \tag{3.7}$$

The discrete quantities of the Gaussian and geodesic curvatures are denoted by $\widehat{K_i}$ and $\widehat{k}_{g,i}$, respectively, as will be demonstrated in this section. The equivalence of the Gauss-Bonnet Theorem in both the smooth and discrete contexts enables the formulation of an integral expression, in which the discrete contact angles are defined in a manner that ensures consistency with the smooth definition of Θ_C for any number of sampling points. This approach permits the utilisation of a significantly reduced number of computational

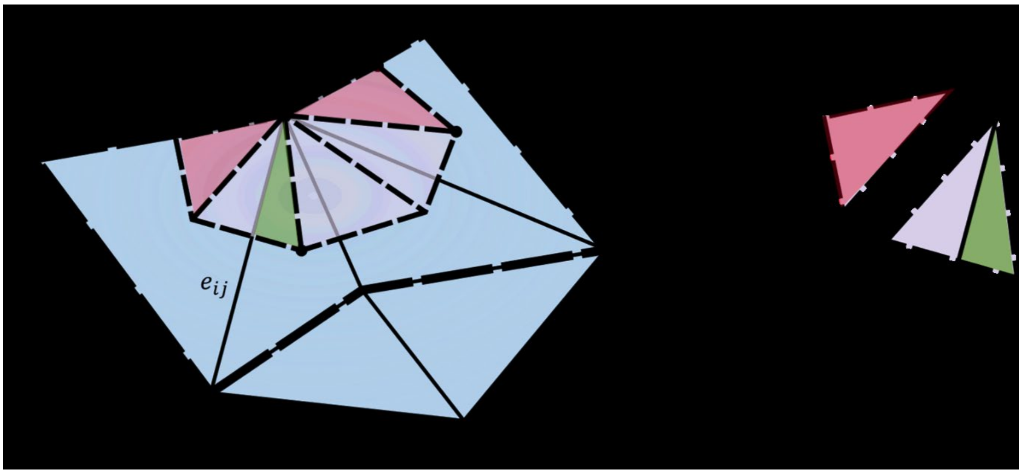

Figure 3.17.: A) A discretised patch of manifold (blue) surrounding a boundary vertex $\boldsymbol{f_i}$, connected to an interior vertex $\boldsymbol{f_j}$ and boundary vertex $\boldsymbol{f_k}$. B) All interior edges $\mathbf{e_{ij}}$ have two associated dual triangles (shown here in purple and green for the interior dual triangles). The boundary edges $\mathbf{e_{ik}}$ have only one dual triangle per edge shown in red. This boundary area was not considered in previous works.

elements in simulations while preserving the properties of the smooth model equations.

In Figure 3.17 a discretised patch of a surface cutout near the boundary is shown. It is immediately apparent that the dual area surrounding boundary is not actually included in the total integral of Equation 3.5 as the dual areas near the boundary vertex $\boldsymbol{f_i}$ is not integrated in any of the interior vertices by definition. The area in Figure 3.17 depicted in red surrounding fig$\boldsymbol{f_i}$ is unique compared to the integral of the mean normal curvature of vertices in the interior (such as the remaining portion of the shaded dual area shown in purple and green) due to only have one dual area associated with a boundary edge such as e_{ik}. This is illustrated in Figure 3.17B, there exists only one dual triangle per edge on the boundary edges, unlike the interior which consistently possesses two associated dual areas. Consequently, Incorporating a term for the boundary, as depicted in Figure 3.17, the mean normal curvatures on the boundary vertices are recomputed based on Equation 3.5 when this boundary is included. This derivation commences from Equation 3.4 with $\mathrm{st}\left(\boldsymbol{f_i}\right) = C$, where ∂S encompasses the entire set of boundary vertices:

$$
\begin{aligned}
&\int_C H\mathbf{N}dA = \int_C df \wedge d\mathbf{N}\\
&\int_C df \wedge d\boldsymbol{N} = \int_C d\boldsymbol{N} \wedge df\\
&\int_C d\boldsymbol{N} \wedge df = \int_C d(\boldsymbol{N} \wedge df)\\
&\int_C d(\boldsymbol{N} \wedge df) = \int_{\partial C} \boldsymbol{N} \wedge df\\
&\int_{\partial C} \boldsymbol{N} \wedge df = \sum_j \int_{e^\star_{ij}} \boldsymbol{N} \wedge df\\
&\sum_j \int_{e^\star_{ij}} \boldsymbol{N} \wedge df = \sum_{j \in st(f_i) \setminus \partial \mathcal{V}} \boldsymbol{N_{ijl}} \times (m-a) + \boldsymbol{N_{ijk}} \times (b-m)\\
&+ \sum_{k \in st(f_i) \cap \partial \mathcal{V}} \boldsymbol{N_{ijl}} \times (m-a),
\end{aligned}
\tag{3.8}
$$

where Stokes' Theorem was again used in the fourth step. The equality symbol on the fifth step is again due to Stokes' theorem (or the fundamental theorem of calculus Consequently, it is possible to identify an approximation for tintegrated along an edge). As before in the interior case, the ratio between the primary and dual length can also be expressed through the cotan formula resulting in:

$$
\begin{aligned}
\widehat{H\boldsymbol{N}dA_i} =& \frac{1}{2} \sum_{j \in st(\boldsymbol{f_i}) \partial \mathcal{V}} \left(\cot \alpha_{ij} + \cot \beta_{ij}\right) \left(\boldsymbol{f_i} - \boldsymbol{f_j}\right)\\
&+ \frac{1}{2} \sum_{k \in st(\boldsymbol{f_i}) \cap \partial \mathcal{V}} \left(\cot \alpha_{ij}\right) \left(\boldsymbol{f_i} - \boldsymbol{f_k}\right)
\end{aligned}
\tag{3.9}
$$

where $\partial \mathcal{V}$ is the global set of boundary vertices for the equation.

Equation 3.9 enables the definition of a mean normal curvature on boundaries, which is exact for locally minimal surfaces through Stokes' Theorem. This allows for the provision of an alternative interpretation of the geodesic curvature. Recognizing that a local patch of the bounded surface at the manifold boundary can be approximated as a convex spherical cap, Figure 3.18 demonstrates the geometric meaning of the quantities. Consequently, it is possible to identify an approximation for the discrete contract angle on the boundaries $\widehat{\Theta}$ through the abstracted relationship to $\widehat{K_i}$, as will be elaborated on in the subsequent paragraphs.

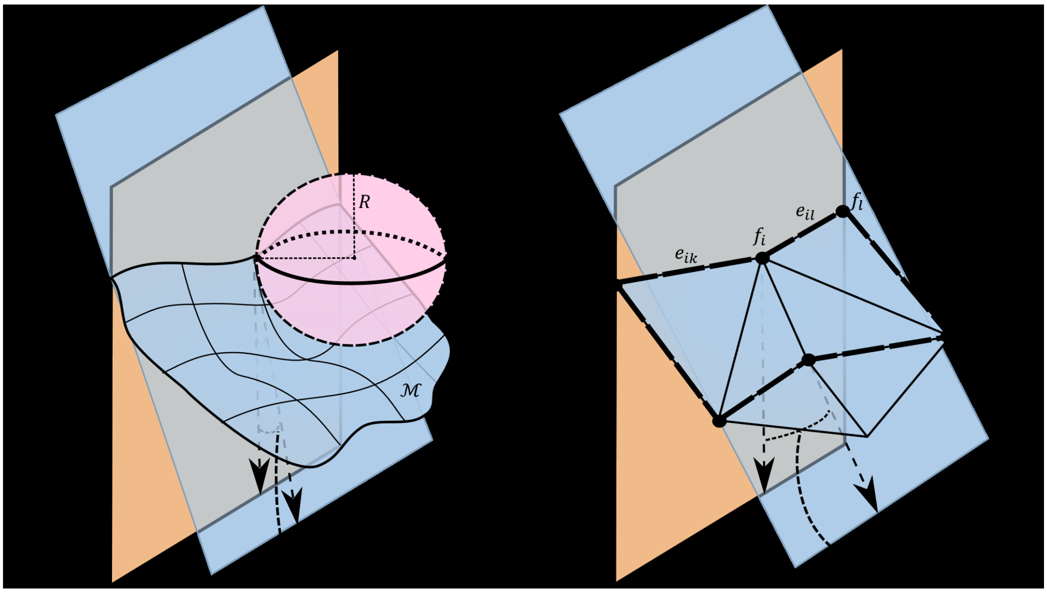

Figure 3.18.: A) Demonstration of the discrete approximation of the smooth contact angle Θ_C. For any arbitrary surface $\mathcal{S}$, a local patch defined by its boundary $\partial\mathcal{S}$ can be excised; this is topologically equivalent to a disk (and therefore has an Euler characteristic $\chi = 1$). B) On a three-phase contact line, a discrete vertex $\mathbf{f_i}$ has a maximum of two neighboring boundary vertices $\mathbf{f_k}$ and $\mathbf{f_j}$ (because it is a discretised curve), which defines the unique direction provided later by Equation (15). Note further that $st\,(\mathbf{f_i})$ is analogous to $\partial\mathcal{S}$ and more precisely it represents an integral approximation of a local neighborhood bounded by $\partial\mathcal{S}$. It can be seen that, in general, the discrete angle θ is not equal to Θ_C. Instead, here the correct angle Θ_C can be obtained by computing a discrete approximation of mean normal and Gaussian curvatures $\widehat{H}$ and $\widehat{K}$ in the domain $st\,(\mathbf{f_i})$, where $\widehat{K} = K$ (exactly analytical) and the manifold is convex for minimal surfaces at equilibrium.

Instead of defining explicit point-wise angles, the contact angles will be defined implicitly through the local-global equilibrium of the curvature (Gauss-Bonnet Theorem). The true contact angle Θ_C is related to the discrete principal curvature estimates $\widehat{\kappa_1}$ and $\widehat{\kappa_2}$ (discrete analogies of the integrated smooth quantities κ_1 and κ_2). In turn, $\widehat{\kappa_1}$, $\widehat{\kappa_2}$, and $\widehat{K_i}$ are related to $\widehat{H_i}$ from Equation 2.12 and Equation 2.14. A crucial insight of the Gauss-Bonnet Theorem is that the global topology of a surface is invariant to local perturbations of the surface. Therefore, the GB relation $\int_{\partial \mathcal{S}} k_g ds = 2\pi\chi\ - \int_{\mathcal{S}} KdA$ (where $\chi = 1$ for a local interface "disk" and ds is a segment of the boundary curve $\partial \mathcal{S}$ on the three-phase contact line) can be used to relate the integrated curvatures in a patch of fluid-fluid surface interface $\mathcal{S}$, as demonstrated in Figure 3.19.

Additionally, when this patch is bounded to a solid-fluid-fluid three-phase contact line, there exists a (smooth) angle $\Theta_{Local}(s)$ between the solid-phase tangent (cf. Figure 3.19B) and the fluid interface tangent at every point s on the contact curve. The "Local" subscript of $\Theta_{Local}(s)$ is used to emphasise the currently computed angle when the interface is in a non-equilibrium state (as opposed to the equilibrium contact angle Θ_C). Figure 3.19 demonstrates how an understanding of (integrated) local curvatures can be connected to an average Θ_C on the contact line. In particular, because the sum of $\int_{\partial \mathcal{S}} k_g ds$ and $\int_{\mathcal{S}} KdA$ is fixed, an averaged three-phase contact angle $\widehat{\Theta_{Local}}$ exists. The term $\int_{\mathcal{S}} KdA$ intuitively indicates how much the patch curves away locally at the boundary.

Through the Gauss-Bonnet theorem, the integrated curvature $\int_{\mathcal{S}} \widehat{K_i} dA$ of all the perturbations of the patch of the fluid-fluid interface $\mathcal{S}$ can be related to an equivalent spherical cap and the smooth integrated curvature $\int_{\mathcal{S}} KdA$ of an equivalent unperturbed sphere. From this relationship, the integrated geodesic curvature $\int_{\partial \mathcal{S}} k_g ds$ can be computed. The precise relation for a spherical cap is given by $k_g = \sqrt{\frac{1}{K}} \tan \widehat{\Theta}$, where latitude $\lambda = \Theta$ (cf. Figure 3.19B and Figure 3.19C; the latitude is a plane cut-out of the great sphere). When the surface is perturbed, this equation provides a good approximation of a contact angle at a point on the three-phase contact line. When the surface is minimised (at equilibrium or at a saddle point), the integral of the discrete contact angle is exact due to the integration being constant. This motivates the development of a discrete Gaussian curvature so that the correct smooth geometry of a physical system can be recovered under any discrete refinement. In Section 4, the equilibrium of a capillary rise system is demonstrated, and it is shown that this formulation of a discretely approximated physical system is independent of the refinement of the complex and the number of vertices or points used in the discretisation.

According to the Gauss-Bonnet theorem, the equilibrium value of κ_i is independent of the problem's geometry (such as tube radius or the geometry of the three-phase path). A formulation for principal curvatures at a point is given by:

$$\begin{aligned} \widehat{\kappa_1} &= \frac{1}{2}\widehat{H}_{\,i} - \sqrt{(\frac{1}{2}\widehat{H_i})^2 - \widehat{\kappa_1}\widehat{\kappa_2}} \\ \widehat{\kappa_2} &= \frac{1}{2}\widehat{H_i} + \sqrt{(\frac{1}{2}\widehat{H_i})^2 - \widehat{\kappa_1}\widehat{\kappa_2}}, \end{aligned} \tag{3.10}$$

where $\widehat{H_i}$ is known from Equation 3.10, the two equations can be solved for the two discrete principal curvatures. It will be demonstrated that, numerically, the analytical Gaussian curvature computed using $K_i = \widehat{\kappa_1}\widehat{\kappa_2}$ is exactly equal (subject to floating-point error) to the analytical solution $K_i = K$ in the symmetric case, which is already a significant improvement over the known angle defect notion of $\widehat{K_i}$ for simulating physical systems.

In summary, $\widehat{H_i}$ is explicitly defined in Equation 3.5, Equation 3.6 and Equation 3.9 through the bounded cotan formula and is computed from the mesh geometry. This value produces an accurate approximation of Laplacian pressure Equation 2.1 (see results in Section 4) under any refinement when the boundary vertices are equilibrated into the correct position (or equivalently when the surface is minimised). The discrete value is exact (by construction) when the smooth surface is locally minimised. From the definition for $\widehat{\kappa_1}$ and $\widehat{\kappa_2}$ in Equation 3.10, a discrete geodesic curvature was defined that allows the boundary vertices to precisely converge to the correct coordinates for a given three-phase contact angle Θ_C through its relation to $\widehat{K_i}$, as demonstrated in Figure 3.19. As a result, it can be expected to retain accurate estimates of the mean curvature H and therefore Young-Laplacian pressure while the boundary motion of the three-phase contact line converges to the correct value when the surface tension is balanced. In idealised cases where the surface is minimal and the mean curvature H is constant, it can be expected to find the exact analytical solution under any refinement, as demonstrated in Section 4).

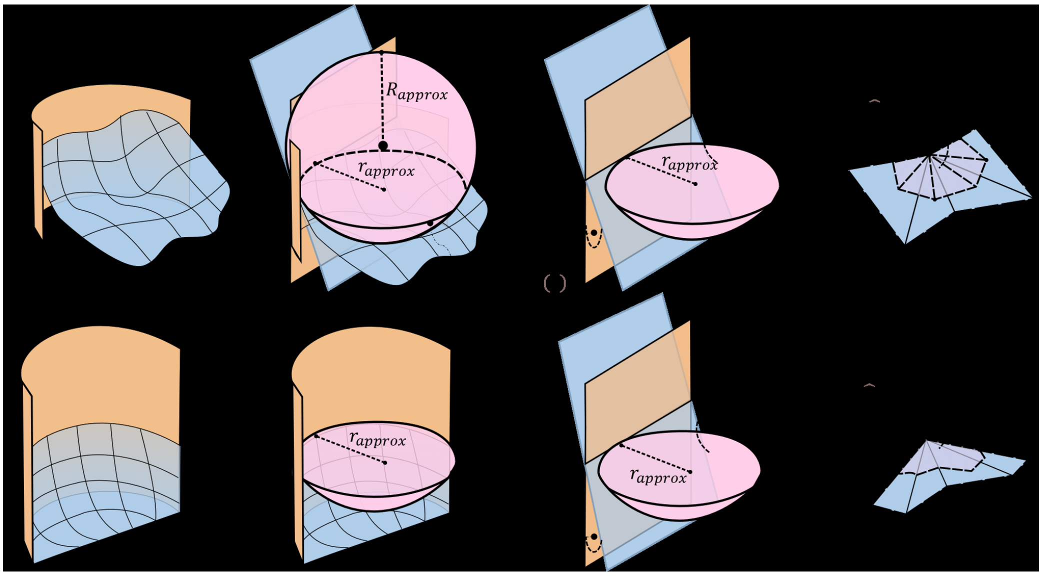

Figure 3.19.: Demonstration of the connection between a three-phase contact angle $\Theta_{\text{Local}}(s)$ and the local Gaussian curvature. A) An arbitrary path of fluid interface in a three phase system that is not in equilibrium. B) At any three-phase contact point there is a uniquely associated fluid-fluid tangent plane (blue), solid contact tangent plane (orange) and additionally an osculating circle (pink) is associated with the approximate plane cut-out circle radius r_{approx} at a latitude $\lambda = \Theta_{\text{Local}}(s)$ with great sphere radius R_{approx}. C) The spherical cap of the smooth great sphere defined by the osculating circle is additionally constrained to a point on the three phase-contact line and therefore has a unique cut at λ. D) The discretisation of the local patch of manifold in (A) has an associate discrete mean normal and Gaussian curvatures from Equations 3.6 and 3.9. The integrated value $\int_{\mathcal{S}} \widehat{K}_i dA$ is approximated from the integration using Stokes' Theorem in the shaded dual area. $\int_{\mathcal{S}} K dA$ can be set to $\int_{\mathcal{S}} \widehat{K}_i dA$ from which the smooth osculating circle geometry can be computed. R_{approx} is found from $\int_{\mathcal{S}} K dA$ and r_{approx} is found from the integrated geodesic curvature $\int_{\partial \mathcal{S}} k_g ds = 2\pi\chi - \int_{\mathcal{S}} K dA$. Therefore, the only source of error is the numerical integration error along the edges of the patch in (D). E) In contrast to the non-equilibrium interface in (A), the fluid interface here is at equilibrium (zero net forces at all points on the three-phase contact line), also known as a minimal surface meaning that it has constant K everywhere and in (G) the latitude $\lambda = \Theta_C$ (exactly) and in (H) the integrated value $\int_{\mathcal{S}} \widehat{K}_i dA$ is exactly equal to the true smooth value $\int_{\mathcal{S}} K dA$ due to the fact that Equation 3.9 is integrating over a constant Gaussian curvature. Figure adapted from Endres, Avila, and Mädler (2022),

Forces on the three-phase boundary lines

Finally, an explicit force vector can be calculated from the force balance at a point on the three-phase boundary curve. If the system is not at equilibrium at an arbitrary contact angle Θ_{Local}, a non-zero net tension γ_{net} exists:

$$\gamma_{net} \ = \gamma_{SL} + \gamma_{LG} cos\Theta_{Local} \ - \ \gamma_{SG},$$

when the system is at equilibrium:

$$0 \ = \gamma_{SL} + \gamma_{LG} cos\Theta_{C} \ - \ \gamma_{SG}.$$

Subtracting the two equalities:

$$\gamma_{net} \ = \gamma_{LG}(cos\Theta_{Local} - \ \cos\Theta_{C}),$$

which has units of N·m^{-1}. By integrating this scalar quantity along a segment ds, the magnitude of the force acting on the contact line in the segment ds can be determined:

$$F_{\mathbf{b}} = \int \gamma_{net}(s)ds \ = \ \int \gamma_{LG}(s)(\mathrm{cos}\Theta_{Local}(s) - \cos\Theta_{C}(s))ds$$

with units in N. Integrating along a definite segment such as an edge e_{ij} provides a well-defined approximation for the averaged force over the edge:

$$F_{\mathbf{b}} = \int_{e_{ij}} \gamma_{net}(s)ds = \int_{e_{ij}} \gamma_{LG}(s)(\mathrm{cos}\Theta_{Local}(s) - \cos\Theta_{C}(s))ds \tag{3.11}$$

In conclusion, the force vector is restricted to the intersection of the fluid interface boundary with the solid phase, as the force balances in other directions are assumed to be zero in the absence of material deformation. Consequently, the force vector $\mathbf{F_b}$ at a point is given by:

$$\mathbf{F_b} = F_{\mathbf{b}} \frac{\mathbf{e_{ij}} \wedge \ \mathbf{e_{ik}}}{\| \ \mathbf{e_{ij}} \wedge \ \mathbf{e_{ik}} \ \|} \tag{3.12}$$

Assuming that $\gamma_{LG}(s)$ and $\Theta_C(s)$ are constant parameters over s, the only remaining task is to compute an approximation of the current (now discrete) angle $\widehat{\Theta_{Local}}(s)$.

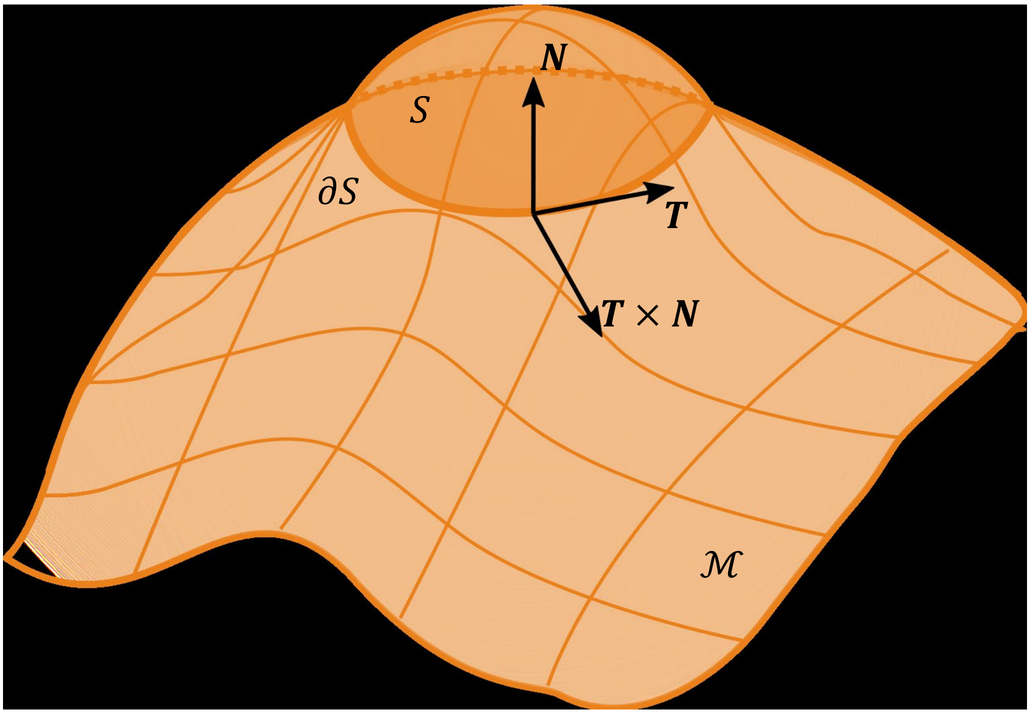

Figure 3.20.: An arbitrary patch of surface interface S and the relations between the tangent vector $\mathbf{T}$ on the boundary ∂S, the normal vector $\mathbf{N}$ and the cross product $\mathbf{T} \times \mathbf{N}$.

Derivation of the Young-Laplacian pressure from curvature

The Young-Laplace Equation 2.1 can be employed to determine the equilibrium pressure difference Δp across a phase interface in equilibrium, based on the constant mean normal curvature H of the interface. In the preceding section, an average non-equilibrium net force acting on an arbitrary segment of the three-phase contact line was derived and linked to the discrete mean normal curvature $\widehat{H}_i$. The relationship between the non-equilibrium $\Delta p, H, \widehat{H}_i$, and the average force $\mathbf{F}_{\text{int}}$ acting on an arbitrary patch of surface $\mathcal{S}$ (e.g., Figure 3.20) between two phases in the interface interior is also significant, for instance, in relating the net forces between two particles due to capillary liquid bridge formation.

Consider two immiscible phases, denoted 1 and 2, with a pressure difference $\Delta p = p_1 - p_2$ between them. The net force at the interface can be calculated using the following equation:

$$\mathbf{F}_{\textbf{int}} = \int_S (p_1 - p_2) \cdot \mathbf{N} dA + \gamma \int_{\partial S} \mathbf{T} \times \mathbf{N} ds \qquad (3.13)$$

When $\mathbf{F}_{\textbf{int}} = 0$ at equilibrium:

$$-\int_S (p_1 - p_2) \cdot \mathbf{N} dA \ = \gamma \int_{\partial S} \mathbf{T} \times \mathbf{N} ds$$

Applying Stokes' Theorem:

$$\begin{aligned} -\int_S (p_1 - p_2) \cdot \mathbf{N} dA \ &= \gamma \int_{\mathbf{S}} d\left(\mathbf{N} \wedge d\mathbf{f}\right) \\ \int_S (p_1 - p_2) \cdot \mathbf{N} dA \ &= \gamma \int_S d\mathbf{f} \wedge d\mathbf{N} \\ \int_S (p_1 - p_2) \cdot \mathbf{N} dA \ &= \gamma \int_C H\mathbf{N} dA \end{aligned}$$

In the limit where S shrinks to a point:

$$\Delta p = \gamma(h) H \tag{3.14}$$

Equation 3.14 relates the pressure difference Δp and the net force net force $\mathbf{F}_{\text{int}}$ across a phase interface in equilibrium on an arbitrary patch of surface interface S with the smooth mean normal curvature H is zero (Equation 3.14). In the second step the minus sign appears due to the anti-symmetric operation on the right-hand side. The discrete mean normal curvature $\widehat{H_i}$ can be used to approximate H, enabling the computation of the pressure difference Δp in capillary rise simulations and the integrated net forces between particles in liquid capillary bridges.

3.5 On the difference between use of point-wise and integrated DDG formulations

The formulations presented in Section 3.4 involve a subtle approximation when total integrated quantities such as $\int_S HNdA$ are divided by the dual area C_i (similar to a gradient). In Section 3.4, the potential for the exact total curvature integral tensor to be directly used in numerical methods is highlighted through a force computation that can be linked to the momentum balance. Symmetric bridges of equally sized particles allow for analytical solutions (Endres, Avila, and Mädler, 2022) and therefore serve as an ideal test case for comparing numerical methods. Using an asymmetric discretisation of such a bridge, the errors in point-wise estimates are demonstrated, along with a comparison to the total integrals. Later in this work, an analytical surface function for asymmetric bridges of differently sized particles is also introduced (Section 4). This method can be employed to compute more accurate forces in liquid bridges for DEM parameterisation as well as modeling wetting of agglomerates in direct numerical simulation.

In Endres, Avila, and Mädler (2022) and in this work in Section 4 it is shown that for some arbitrary patch of interface $\mathcal{S}$, the mean normal curvature $H\boldsymbol{N}$ can be computed precisely using the explicit closed form solution of the Stokes integral as derived in Equation 3.9 from which the equality can be set:

$$\int_S H\boldsymbol{N}dA = \widehat{HNdA_i}$$

From Equation 3.13 to the force acting on the surface S can also be related to curvature. For two immiscible phases 1 and 2 with a pressure difference $\Delta p = p_1 - p_2$ between them, the net force is:

$$\boldsymbol{F}_S = \int_S (p_1 - p_2) \cdot \boldsymbol{N}dA + \gamma \int_{\partial S} \boldsymbol{T} \times \boldsymbol{N}ds = \gamma \int_S H\boldsymbol{N}dA$$

Following through the derivation:

$$\boldsymbol{F}_S = \boldsymbol{N}dA$$

In the second step the minus sign appears due to the anti-symmetric operation on the right-hand side. Remembering that in the limit where $\mathcal{S}$ shrinks to a point it is possible to retrieve the Young-Laplace Equation $\Delta p = \gamma H$. A fundamental problem when trying to find the approximate value of $\int_S HNdA = \widehat{HNdA_i}$ at a point was elucidated by Wardetzky (2008) who rigorously proved that the cotan formula fails to converge in the space of interest. In Endres, Avila, and Mädler (2022) and shown in Section 4 the approximation $\frac{\int_C HNdA \cdot N}{C_i}$ will be used and it will be demonstrated that this approximation was sufficiently accurate for symmetric triangulations shown in several test cases.

In concluding this chapter, a final important relation back to momentum balances in continuum CFD is the total force acting on the boundary of a three-phase contact line $\mathrm{S}_b \subset \partial S$ with contact angle parameter Θ_C in a nonequilibrium state with the three-phase contact line angle $\Theta_{\text{local}}(s)$ at each point $s \subset \partial\mathcal{S}$ from Equation 3.11. These equations are sufficient to define the multiverse momentum balance. Starting fundamentally from Newton's

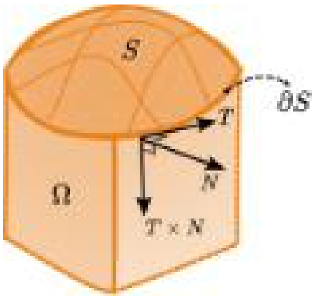

Figure 3.21.: A control volume Ω that is intersected by a phase interface S. Note that a control volume in the bulk fluid can have $S =$ and therefore Equation 6 is fully general. A three-phase contact line $S_b \subset \partial S$ may or may not exist on the boundary of the interface, however, by construction the contact line cannot be in the open set area of S and similarly it is only possible for S to be subset of $\partial\Omega$ and is never contained in the open set volume of Ω.

second law $F_i = ma_i$, in any component direction i, a Cauchy momentum balance by considering a control volume $\Omega \subset \mathbb{R}^d (d = 2, 3)$ can be derived:

$$\int_\Omega \rho \frac{Du_i}{Dt} dV = \int_\Omega \rho f_i dV + \int_\Omega \nabla_j \sigma_i^j dV + \int_S (p_1 - p_2) \cdot N_i dA + \gamma \int_{\partial S} H N_i dA + \int_{S_b} \gamma(s) \left(\cos \Theta_{\text{local}}(s) - \cos \Theta_C(s)\right) ds \tag{3.15}$$

where the component of the Cauchy stress tensor (a constituent equation for fluids leading to the NavierStokes ($\mathrm{N-S}$) equation) is $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}, \boldsymbol{\tau}$ is the deviatoric stress tensor, $\mathbf{I}$ is the identity matrix). The other terms include the material derivative of velocity (in vector form $\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial dt} + \mathbf{u} \cdot \nabla \mathbf{u}$) which is equal to the integral of the stress and body forces f_i (ex. gravity) acting on the volume element. Because Equation 3.15 applies to any arbitrary control volume, the equation holds and the differential form of the Cauchy momentum equation (and in turn the multiphase $\mathrm{N-S}$ equation) can be obtained. However, in this work the case is considered where a control volume is bounded by a phase interface S intersecting the control volume as demonstrated in Figure 3.21. The body forces $\int_\Omega \rho f_i dV$ term and the acceleration term $\int_\Omega \rho \frac{Du_i}{Dt} dV$ is invariant to the interface size since the interface does not have bulk components to act on the mass of the control volume. While Figure 3.21 is written in differential form, the actual solution of momentum balance equations is typically computed by solving the potentials in integro-differential form. Consequently, in most methods available in software libraries, Figure 3.21 can be directly utilised with an appropriate coordinate system. Section 4.2 will demonstrate the difference between these approaches in experimental results simulating particle-particle liquid bridges.

4 Applications of Discrete Differential Geometry for Test Cases with Three-Phase Equilibrium

Discrete differential geometry in the context of physical multiphase systems

In this chapter specific applications of the method and libraries developed in Section 3 are validated in the context of physical multiphase systems. A particular focus is on validating the method against the fundamental test cases of capillary rise, for which exact abstract solutions exist. The method's validation using experimental sessile microdroplet data is also demonstrated. Finally, the method is applied to symmetric and asymmetric particle-particle liquid bridges in order to determine the forces involved in the agglomeration mechanism.

Section 4.2 focuses on a core result of this work; the precise calculation of curvature on arbitrarily discretised surfaces, enabling the computation of surface tension forces with exceptional precision. It is important to distinguish between the symmetric case (for which well-known solutions exist) and the asymmetric case (for which exact analytical solutions were previously lacking). The validation test cases demonstrate the method's versatility in a variety of applications and especially with respect to the agglomeration mechanism in wet particulate systems.

4.1 Test Cases for Analytical Validation

This section showcases applications of DDG curvature computations to the physical processes of capillary rise, and Sessile droplet formation, along with the associated errors. These case studies can be applied to any physical

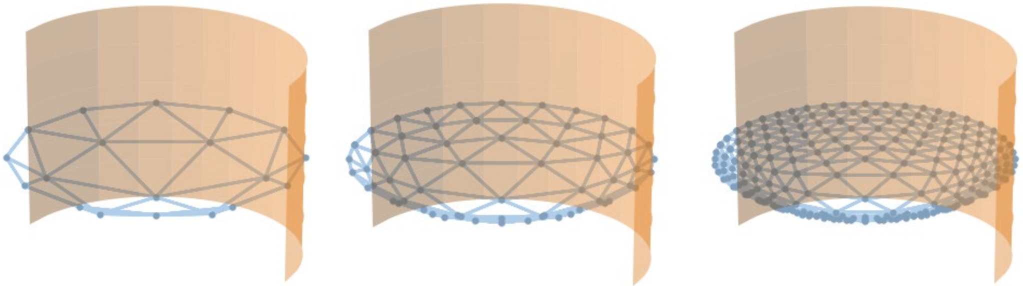

Figure 4.1.: Examples of increasingly fine discretisations of a capillary rise film in 3 dimensions using n = 20, n = 68 and n = 249 vertices.

surface embedded in three-dimensional space through the Gauss-Bonnet theorem. However, a homology computation is necessary to apply the theorem to a given surface topology (therefore, caution should be exercised in ensuring the correct refinement error to prevent solid phases from intersecting a local integration domain of fluid-fluid or fluid-gas interfaces). A homology computation for this purpose can be carried out in three-dimensional space with relatively low computational cost as discussed in Section 3. The accuracy of the DDG curvature computations is demonstrated for the case of a macroscopic capillary rise process. The second case study showcases the computation and refinement in non-ideal droplets and compares the results to the corresponding Surface Evolver software implementation by (Brakke, 1992).

4.1.1 Capillary rise (surfaces with boundary)

The equilibrium solution of a capillary rise (Equation 2.2) exhibits a minimal surface with a known analytical curvature. Specifically, when the radius of the tube is sufficiently small, it can be idealised as a minimum sphere. This serves as a motivating test case for the discrete boundary contact angle formulation. The aim of this study is to determine discrete approximations of a capillary surface interface with a known analytical mean normal curvature $H = \frac{2}{R}$ in order to gain insights into both the accuracy of the formulation and the numerical accuracies achievable when simulating non-ideal systems. Three parameters are considered: the three-phase contact angle Θ_C, the number of boundary vertices (N), and the tube radius $a = 0.5$ mm. The capillary length $\lambda_c = \sqrt{\frac{\gamma}{\Delta\rho g}} = 0.00271$ m at STP. Figure 4.1 illustrates the half-cylinder cut-out discretisation of this system.

Here the Young-Laplace pressure is determined for a capillary rise of water in a solid tube with a radius of $a = 0.5$ mm, assuming a three-phase contact angle of $\Theta_c = 45^{\circ}$. The current formulation enables the accurate prediction of the Young-Laplace-derived capillary force under any refinement of the film triangulation (Figure 4.2). The capillary force error is defined as $\left(F_{cap} - \widehat{F_{cap}}\right)/F_{cap}$, where $\widehat{F_{cap}}$ represents the force computed using the numerical method and F_{cap} represents the analytical force derived from the Young-Laplace equation. Notably, a very small number of vertices can still reproduce the correct analytical solution, thereby accurately predicting both the Young-Laplace pressure and the capillary motion on the boundaries. The integration error reported in Figure 4.2 is a crucial tool in dynamic simulations, providing an estimate of the local geometrical precision of the simulation.

For instance, in intricate geometries such as micropores, it is crucial to adjust the refinement based on the local variations in pore dimensions to prevent integration errors over any significant features. Since the integration between edges was performed using only the boundary vertex values on the edges, the error in the scalar field values, according to Stokes' Theorem, is simply integration using the trapezoidal rule. Figure 4.3 provides a more detailed illustration of the trapezoidal rule and the error computation. It is important to note that, due to the fundamental theorem of calculus, integration over an edge $\mathbf{e_{ij}}$ is exact when the surface is Lipschitz continuous.

Since the interior height map is continuously captured across a dual domain through Stokes' Theorem in Equation 3.4, there is no integration error if the interior of the domain $(\mathbf{fi})$ is also Lipschitz continuous. It is crucial to consider the error since, for instance, when the solid geometry intersects the domain $(\mathbf{fi})$, the surface ceases to exist in a subset of the domain, necessitating interface refinement to resolve the geometric error in the simulation. The error when integrating over an edge $\mathbf{e_{ij}} = \mathbf{fj} - \mathbf{f_i}$ is equivalent to the error of the trapezoidal rule, which propagates through the Stokes' integration and should therefore be controlled during any dynamic simulation of non-ideal geometries to resolve non-convex areas of the solid phase and maintain high accuracy while the system evolves over time.

Accuracy of mean normal curvature estimates for different triangulations of a capillary rise problem.

Next, consider the symmetrical minimum discretisation of a sphere, enabling the study of the total integral error for different incidences on a three-phase

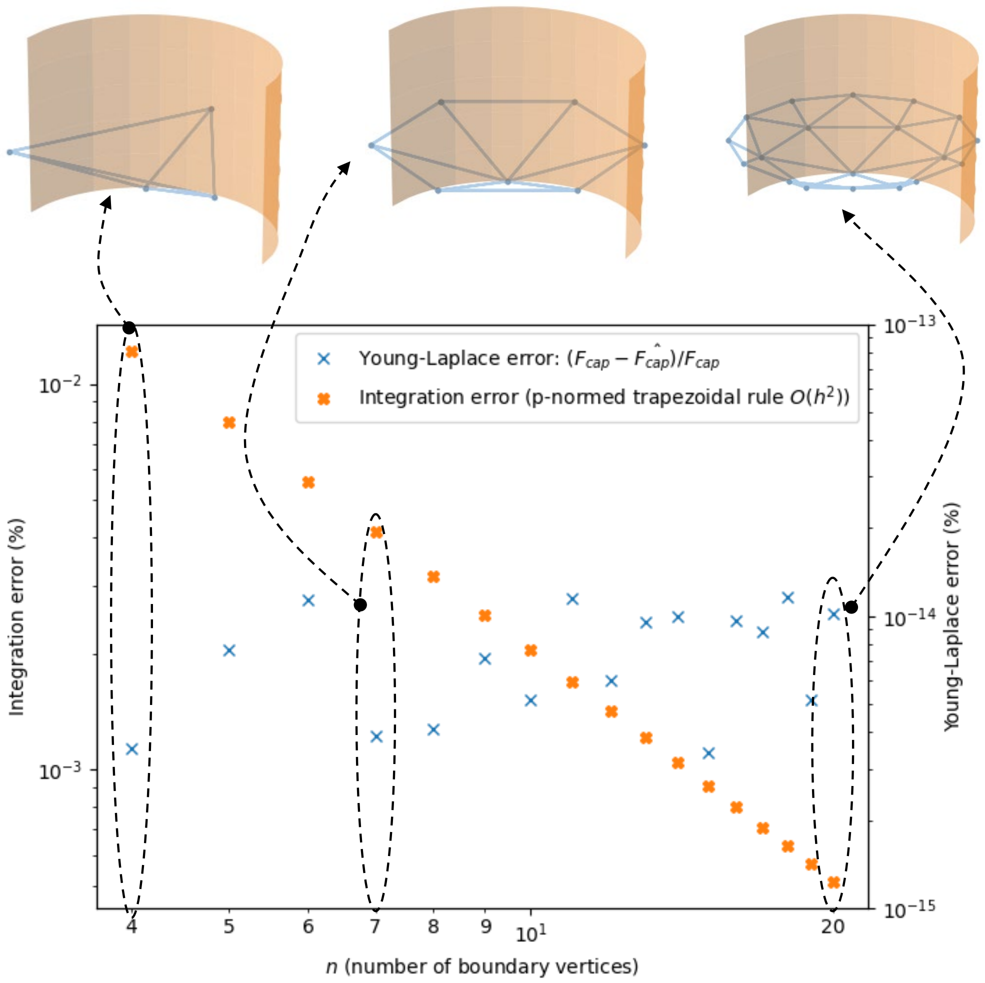

Figure 4.2.: Refining the accuracy of the surface interface at a contact angle $\Theta = 45°$ and tube radius $a = 0.5$ mm. It should be noted that the approximations for normalised mean normal curvature (orange) and the discrete Laplacian pressure $\Delta\widehat{p}$ are almost precisely on the analytical values of H and Δp even when the integration error (orange) is high. The equilibrium films for n=4, n=7 and n=20 are shown on top. The computed convergence rate from a fitted trendline is ≈ 1.975 in agreement with the second-order convergence of the trapezoidal rule.

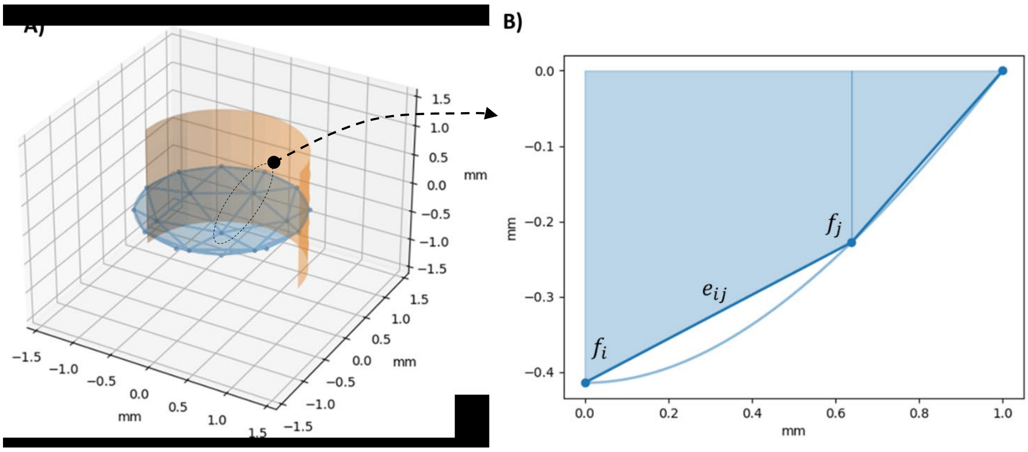

Figure 4.3.: A) A triangulation on a smooth liquid-air interface in a tube with a=1.0 mm. B) The Trapezoidal rule in the x_1=0 plane cut-out from A). The error across an edge e_{ij} is known to be equal to $\epsilon = \frac{f''(\varepsilon)(b-a)^3}{12n^2}$ where ε is any number in an interval $[a, b]$ and therefore the integration error is proportional to $\left\|(f_j - f_i)^3\right\|$. The error can be further reduced by O(n^{-2}) by sampling n vertices on the edge e_{ij}.

boundary vertex. By analyzing the curvature in the vicinity of a vertex, the mean normal curvatures and numerical accuracies can be determined for various refinements. Figure 4.4 presents the results for a spherical cap with a radius $a = 0.5$ mm (a specific theoretical case where $H = \kappa_1 + \kappa_2 = K = \kappa_1\kappa_2$ at $\Theta_C = 0$; note that for significantly smaller, more realistic a values, H becomes very large relative to K, and it is challenging to visualise the comparison on a single graph). Observe that for any value of Θ_C, the dual area normalised mean normal vector curvature $\widehat{H_i}$ calculated using Equation 3.6 provides an almost exact estimate of the curvature H for the entire range of Θ_C.

A remarkable observation from Figure 4.4 is that it enables the demonstration of a highly accurate estimate for the mean normal curvature in the vicinity of a vertex. Furthermore, for higher values of Θ_C, an equivalent estimate for the Gaussian curvature can be obtained, which compensates for the lack of precise concepts of discrete Gaussian curvature available for physical simulation. This partially motivates the use of the relationship between $\widehat{H_i}$ and $\widehat{K_i}$ in this formulation. This implies that with the current formulation, the physical forces can be recovered with near-perfect accuracy, subject to the integral error over the boundary path edges of the neighborhood. Crucially, the curvature-driven physical forces remain unchanged under refinement of the mesh in convex domains. It is essential to resolve the geometry such that it is convex in the local domain to a specified precision, for which an accuracy

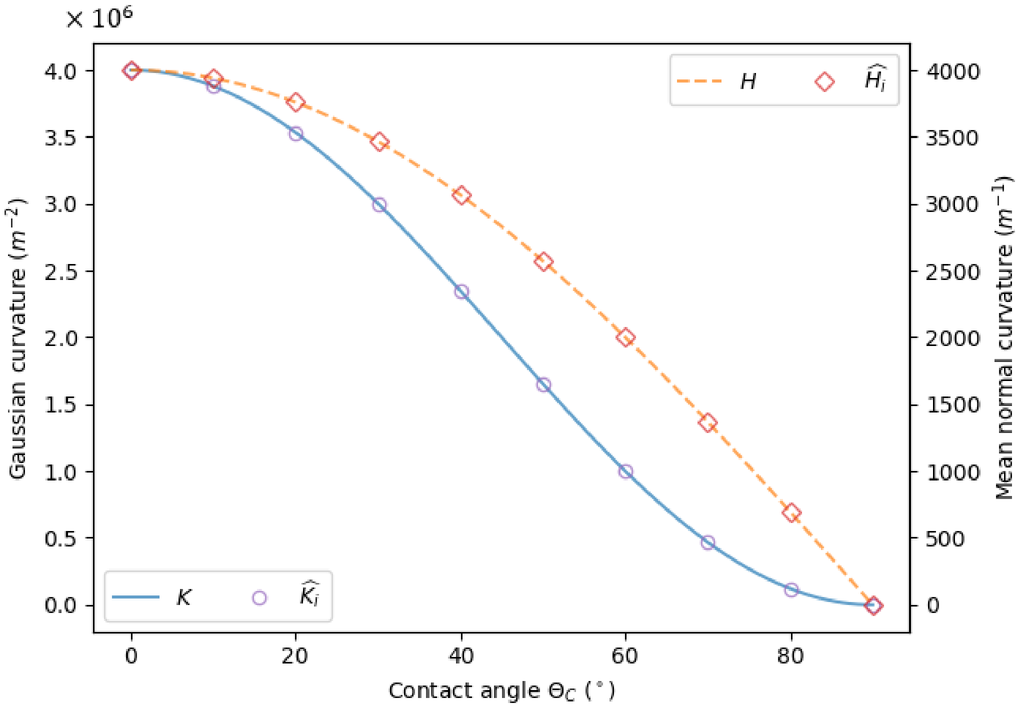

Figure 4.4.: The discrete Gaussian curvature $\widehat{K}_i$ and discrete mean normal curvature $\widehat{HN}_i$ are nearly equal to the exact smooth analytical solutions of K and H throughout the range of hydrophilic three phase contact angles Θ_C.

can be calculated around a given vertex neighborhood. In the following subsection, it is demonstrated how the definitions for boundary curvatures of Boundary gas-liquid-solid interface contact angles can be employed in the capillary rise interface energy minimization problem and lead to the correct hydrostatic equilibrium.

4.1.2 Mean curvature interface energy minimisation of a capillary rise

This section demonstrates the convergence of the discrete formulation to the correct equilibrium solution shown in Figure 4.5. The internal energy of the interface is defined by

$$U^j = \int_{\partial S} \gamma^j dA. \tag{4.1}$$

Interface energy minimisation and the algorithmic details involved in are provided in Section D, in short this is simply the minimisation of Equation 4.1 by moving the discrete representation of the surface interface in the direction of the energy gradient, subject to the physical constraints of the system. In addition to using the same model for water, a gravity field with g $=(0,0,-9.81)$ m $\cdot$ s^{-2} is added, and a contact angle of $\Theta_C = 20°$ is specified. Starting from a flat reservoir of water, the film rises (as expected due to the surface tension and under the thermodynamic assumptions stated in Section D to an equilibrium solution (with the meniscus at the Jurin height) with the lowest vertex on the meniscus near $h_J = \frac{2\gamma\cos\Theta_C}{\rho g a}$ with a Young-Laplacian error of less than 1×10^{-10} at equilibrium for all films, including very sparsely refined films. To achieve equilibrium, the films were perturbed when the film was stuck in a local minima saddle point during the interface energy mimisation. Such saddle points are described in more detail in Brakke (1992).

This finding is crucial as it demonstrates that the sampling density of a fluid interface can be exceptionally low. As long as the area containing a domain st $(\mathbf{f_i})$ is convex, the fluid geometry can be resolved with high precision. This implies that in the simulation of larger-scale multiphase systems with far larger surface areas to track, the formulation can be employed to significantly reduce computational costs and enhance tractability for many systems of interest with small Weber numbers (where capillary forces dominate over inertial forces).

4.1.3 The Sessile Droplet Test Case for Experimental Validation

Case study: Sessile droplets (non-ideal surfaces) droplets)

In order to illustrate the benefits of the discrete differential geometry boundary formulation over traditional surface interface minimisation techniques, the case of a Sessile droplet is demonstrated. The Sessile droplet is a non-ideal surface, closely approximating but not exactly resembling a spherical cap. The experimental Sessile droplet data from Murray, Fox, and Narayanan (2020) is used here, which describes an evaporating water droplet in contact with a gold-coated flat surface with a measured three-phase contact angle of $\Theta_C \ = 46.8°$ and an initial mass of 1.20 μg) at a temperature of 26 °C. To model water, the IAPWS equation of state (Wagner and Pruß, 2002) was utilised to establish the relationship between pressure and density, and the revised IAPWS model (Petrova and Dooley, 2014) for surface tension. The geometric data was extracted from Fig. 4 in Murray, Fox, and Narayanan

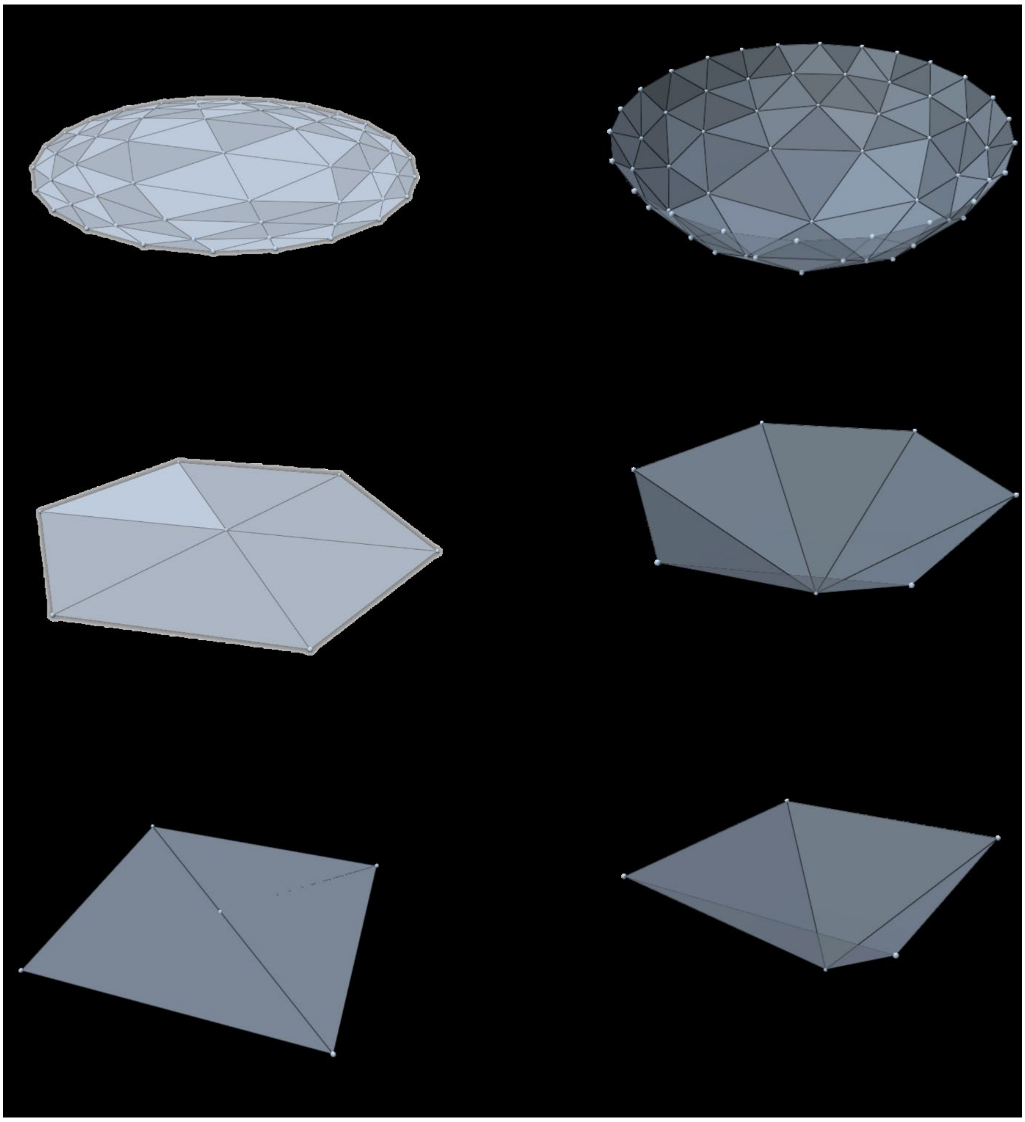

Figure 4.5.: Mean curvature interface energy minimisation for water in a tube with a radius (a = 0.5) mm and a contact angle of $\Theta_C = 20$ degrees, transitioning from a flat film to an equilibrium meniscus. A) The final capillary height rise h_{final} is close to the expected height from the analytical solution, $h_J = 0.00279$ m. B) An identical geometric solution to A) is obtained using a much sparser fluid interface with $n = 7$ vertices. C) The solution is invariant to the number of boundary vertices and the edge incidence of the mesh because forces are integrated over discretised areas. This enables the use of much sparser simulations to achieve accurate results, as demonstrated in the case of $n = 5$ shown here. The final global error (calculated as the net force at the vertex point, which is expected to be zero) at equilibrium is less than 10^{-10} for all vertices.

(2020) using the Canny edge detector available in scikit-image (Walt et al., 2014) with a Gaussian smoothing of $\sigma = 3$.

The curvature interface energy minimization is initiated using an initial cylinder with the same mass as the Sessile droplet. The energy is minimised, and the resulting equilibrium droplet is compared to physical data from a Sessile droplet test, as shown in Figure 4.6. The model error is less than 0.001% from each vertex to the nearest data point, even using the minimum (symmetric) triangulation of the cylinder. The results can also be compared to conventional methods, such as the Quadratic model for surface energy implemented in the Surface Evolver software (Brakke, 1992). The Surface Evolver results in Figure 4.7 demonstrate that the droplet data can only be accurately represented after a large amount of refinement, which is unnecessarily expensive compared to the DDG boundary formulation. This implies that for physical simulation, less discrete data is needed to accurately represent the surface interface, and the mean normal curvature can be computed more accurately at a lower resolution of discrete points. This also has general surface applications, where the boundary angles are fixed constraints, and the DDG boundary formulation can be used to produce equilibrium surfaces with greater accuracy.

The relative error from these simulations are summarised in Figure 4.8. Figure 4.8 should be considered an indirect comparison intended solely to highlight the qualitative trends of the convergence of the respective formulations to the data as the number of vertices in the triangulation is increased. Notably, the DDG formulation converges even for a sparse triangulation (and subsequently produces a consistent error with the later increase in error possibly due to floating-point error accumulation), whereas the conventional Surface Evolver software requires more vertices. A direct quantitative comparison is challenging due to (i) the existence of experimental errors in the Sessile droplet study, (ii) the parameter dependence of the Canny edge detector algorithm, and (iii) the use of geometric volume in Surface Evolver, which produces a higher error bias than the DDG implementation, which is specifically designed for sparse systems. Nevertheless, the qualitative advantage of employing the new three-phase angle formulation is clearly demonstrated in Figure 4.8.

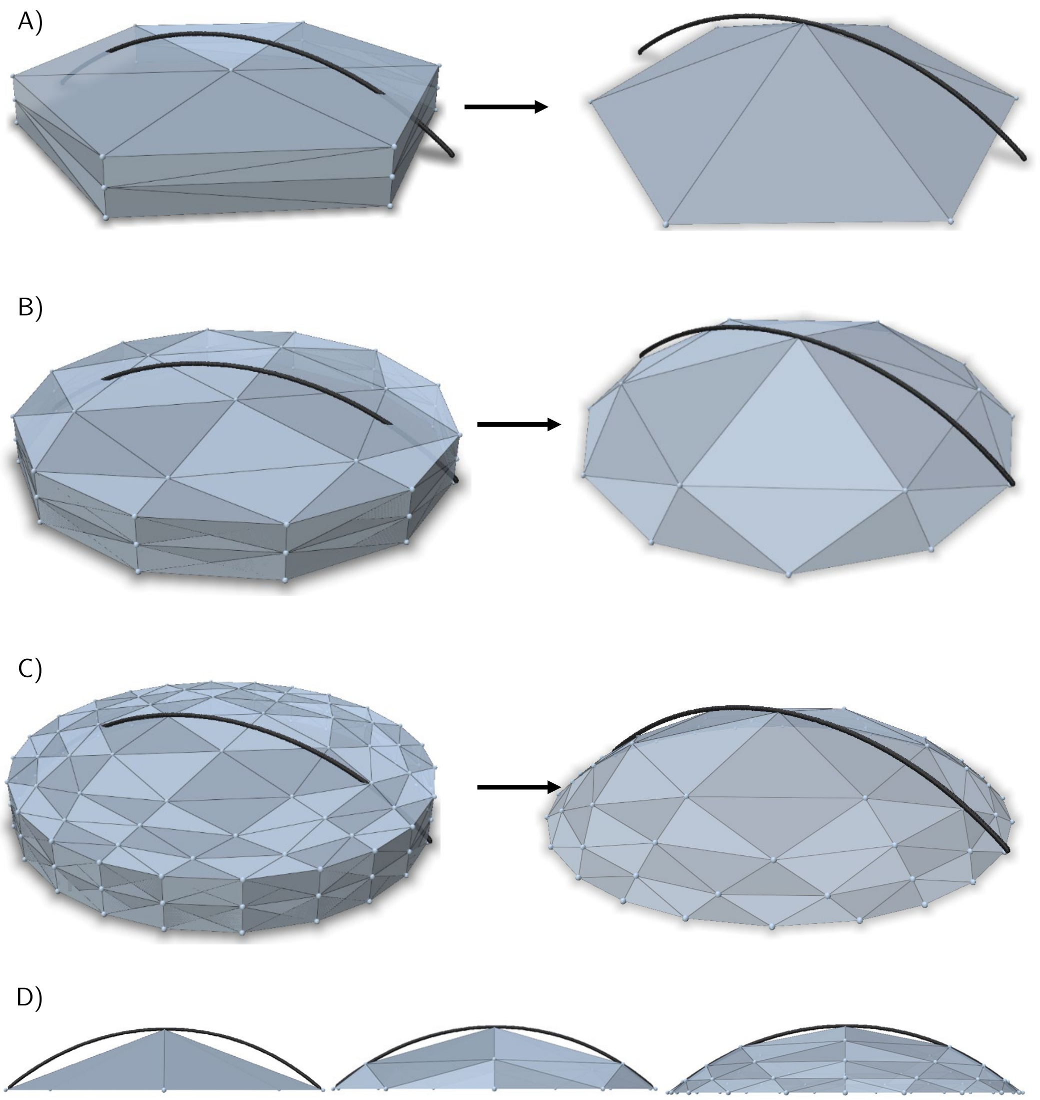

Figure 4.6.: A) A sparse simplicial complex triangulation of an initial cylinder of water (blue) with the same mass as the Sessile microdroplet discretised with $n = 43$ initial vertices. The data extracted from Murray, Fox, & Narayanan (2020) using the Canny edge detector algorithm is shown in the black bold curve. The complex is then allowed to move according to the algorithm described in Section D and converges to $\sum_{n \in \, \partial \mathcal{S}} \parallel \Theta_C - \Theta_{Current} \parallel / \, n = 2.838 \times 10^{-13}$ and $n = 7$ final vertices. As an error estimate compare the norm of the interior vertices on the $y = 0$ axis that are the closest the data points on that plane resulting in $\sum_{\mathbf{f_i} \in \, \mathcal{S} \setminus \partial \mathcal{S} | y=0} \min_{i,j} \frac{\| \mathbf{f_i} - \mathbf{f_j^*} \|}{\mathbf{f_j^*}} = 2.527 \times 10^{-6}$ where $\mathbf{f_j^*}$ denotes a data point on the $y = 0$ axis (shown in black) and $\mathbf{f_i}$ is a vertex in the complex interior (blue). B) An intermediately sparse initial cylinder with $n = 43$ initial vertices are plotted together with the data points. The system converges to $\sum_{\mathbf{f_i} \in \, \mathcal{S} \setminus \partial \mathcal{S} | y=0} \min_{i,j} \frac{\| \mathbf{f_i} - \mathbf{f_j^*} \|}{\mathbf{f_j^*}} = 2.527 \times 10^{-6}$ with $n = 19$ vertices. C) A higher sampled triangulation with $n = 109$ initial vertices. The system converges to an average error of $\sum_{\mathbf{f_i} \in \, \mathcal{S} \setminus \partial \mathcal{S} | y=0} \min_{i,j} \frac{\| \mathbf{f_i} - \mathbf{f_j^*} \|}{\mathbf{f_j^*}} = 9.176 \times 10^{-6}$ with $n = 61$ vertices in the final complex. D) Side profile of simplicial complexes shown in A, B, C. After reaching equilibrium, the final equilibrium vertices on the y-axis closely match the data points, with a nearest distance error of less than 0.001% for each vertex.

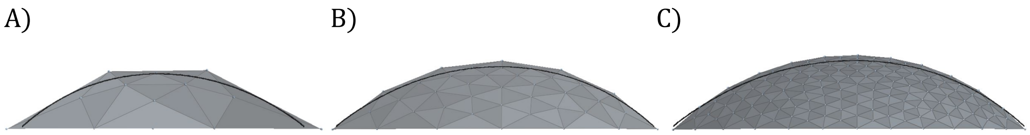

Figure 4.7.: A) For sparse refinements Surface Evolver cannot accurately represent the geometry at equilibrium ($\sum_{\mathbf{f_i}\in\ \mathcal{S}\backslash\partial\mathcal{S}|y=0} \min_{i,j} \frac{\|\mathbf{f_i}-\mathbf{f_j^*}\|}{\mathbf{f_j^*}} = 0.01093$ for $n = 12$ vertices). The half spheres shown are using a symmetric boundary condition. Note in particular that the boundary formulation assumes that the geometric angle is equal to the three-phase contact angle (Figure 1B) which results in the boundary facets not having all vertices near data as well as other defects such as the top vertex not converging to the correct position. B) The droplet converges to equilibrium only after sufficient refinement (at the minimum energy we found $\sum_{\mathbf{f_i}\in\ \mathcal{S}\backslash\partial\mathcal{S}|y=0} \min_{i,j} \frac{\|\mathbf{f_i}-\mathbf{f_j^*}\|}{\mathbf{f_j^*}} = 3.52 \times 10^{-3}$ and $\sum_{\mathbf{f_i}\in\ \mathcal{S}\backslash\partial\mathcal{S}|y=0} \min_{i,j} \frac{\|\mathbf{f_i}-\mathbf{f_j^*}\|}{\mathbf{f_j^*}} = 2.27 \times 10^{-3}$ in (C) respectively at the energy minima using $n = 39$ and $n = 141$ vertices). Note that Surface Evolver enforces the three-phase contact angle on facets (here between the triangles of the interface triangulation and the ground plane). Therefore, from this half-sphere cut out, the contact angle on the edge doesn't necessarily match a tangent line from the data at the three-phase contact point.

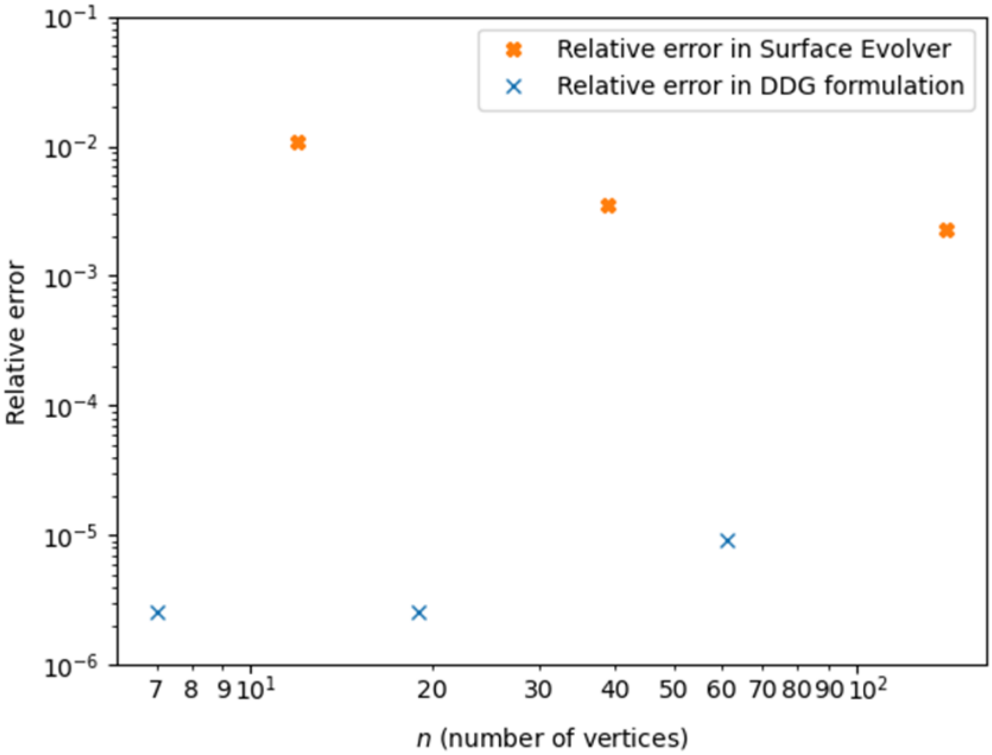

Figure 4.8.: Relative errors of the presented DDG formulation were compared to the Surface Evolver model. The relative errors are computed using the formula $\sum_{\mathbf{f_i}\in\ \mathcal{S}\backslash\partial\mathcal{S}|y=0} \min_{i,j} \frac{\|\mathbf{f_i}-\mathbf{f_j^*}\|}{\mathbf{f_j^*}}$ (the norm of the interior vertices on the y=0 axis that are the closest the data points on the axis).

4.2 Agglomeration: Particle-Particle Liquid Bridges

The agglomeration mechanism in particle-particle liquid bridges

This section compares the accuracy of the DDG formulation developed here for liquid capillary bridges. Capillary bridges play a crucial role in particle systems (see Balakin et al. (2013) for a review) in addition to various other applications in materials and colloidal science. For example, they also hold significance in biological applications, such as simulations of cell structures (Pampaloni and Stelzer, 2009) and the study of respiratory diseases (Alencar et al., 2005).

4.2.1 Symmetric liquid bridges between two particles of the same size

Case study: Capillary bridges

The resulting capillary force between two particles is determined using the following equation:

$$F_{\text{Cap}} = \Delta P A_m + 2\gamma U_m \tag{4.2}$$

Where A_m and U_m are the area of the cross-subsection and the circumference of the meniscus neck respectively, γ is the surface tension, and the Laplacian pressure drop is given by Equation 2.1. Analytical solutions to symmetric capillary bridges in equilibrium have been found by substituting the Laplacian equation into Equation 2.10 and solving the resulting PDE for a constant mean curvature H on a given resolved surface embedded by a surface function f. It was proven by Delaunay (1841) that all equilibrium surfaces that are a solution to this PDE system have constant mean normal curvature H everywhere on the surface. As the primary application of DDG is to accurately simulate dynamic systems, encompassing inertial effects and viscous forces, validating its effectiveness for non-symmetric, non-equilibrium systems is not possible due to the absence of known analytical solutions for such systems.

Therefore, to assess the numerical method's performance, the equilibrium surface is discretised and the accuracy of the discrete solution is compared to the analytical solution, as in the capillary rise test case.

Accuracy of mean normal curvature estimate for different triangulations of an equilibrium capillary bridge problem

In order to validate the numerical accuracy of the discrete differential geometry (DDG) formulation, the exact analytical equation for the Catenoid surface, which represents the analytical solution for the capillary bridge system, was employed:

$$\begin{aligned} x &= q \ \cosh\left(\frac{v}{q}\right)\cos u \\ y &= q \ \cosh\left(\frac{v}{q}\right)\sin u \\ z &= v. \end{aligned} \tag{4.3}$$

Where q is a constant, v is any real number in this case within the chosen interval $v \in [-1,\ 1]$ and $u \in [0, 2\pi)$. Catenoids always have a mean normal curvature $H(u, v) = 0$ at every point on the surface. These can take on the appearance of a discretised liquid bridge as shown in Figure 4.9.

Integrating the solution over the entire simplicial complex eliminates all signed values, resulting in a zero mean normal curvature. This process correctly recovers the total forces exerted in both directions across the bridge $(\mathbf{N}_{+\mathbf{z}} = (0,\ 0,\ 1)$ and $\mathbf{N}_{-\mathbf{z}} = (0,\ 0,\ -1))$ as shown in Figure 4.10. However, it is important to note that while the correct integrated solution can be recovered as expected, since the true integrated value is effectively computed (this equation is an equality), the point-wise approximation is found to be less accurate. In other words, the discrete value $\widehat{H_i}$ for non-convex surfaces is less accurate than the convex case of capillary rise systems.

The latter exhibits near-exact accuracy when $\widehat{H_i}$ is compared to the analytical H at a given point on the surface (x, y, z) under any refinement, even using a minimal number of vertex points. This is demonstrated in Figure 4.10 where the value of $\widehat{H_i}$ is calculated using the the point-wise value for $\widehat{H_i}$ computed as $\widehat{H_i} = H\boldsymbol{N}dA_i \cdot \widehat{\boldsymbol{N}_i}\left(\boldsymbol{f}_i(u, v)\right)/C_i$ based on Equation 3.6. Additionally, for

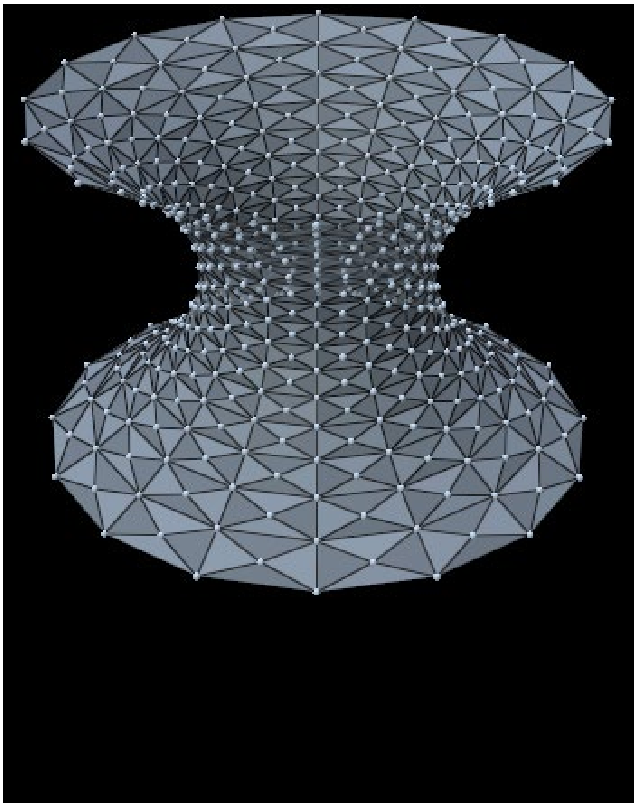

Figure 4.9.: A discretised Catenoid surface with $q = 1$ and $v \in [-1,\ 1]$.
as shown in

this case $\widehat{\mathbf{N}}_i$ is estimated as $\widehat{\boldsymbol{N}}_i\,(\boldsymbol{f_i}(u,\ v)) \ \ = \boldsymbol{f_i}(u,\ v)^{\smile}\ (0, 0, z_i)$ where $\widehat{\boldsymbol{N}}_i$ is estimated as $\widehat{\boldsymbol{N}}_i\,(\boldsymbol{f_i}(u, v)) = \boldsymbol{f_i}(u, v) - (0, 0, z_i)$ and z_i is the height component of $\boldsymbol{f}_i(u, v)$. $\widehat{K}_i$ is computed from the principal curvatures computed from the implicit principal curvature relations Equation 3.10. it should be noted that while the total errors shown in Figure 4.10 are almost negligible, the point-wise errors as measured by comparing to the analytical solutions at each vertex are inaccurate as shown in Figure 4.11. Therefore, it is symmetries in the opposing normal vector directions cancelling out this error thereby producing a low total error when compared the total bridge force. In the next subsection the asymmetric forces are closely investigated in order to develop a better understanding of the point-wise vs. integrated curvature.

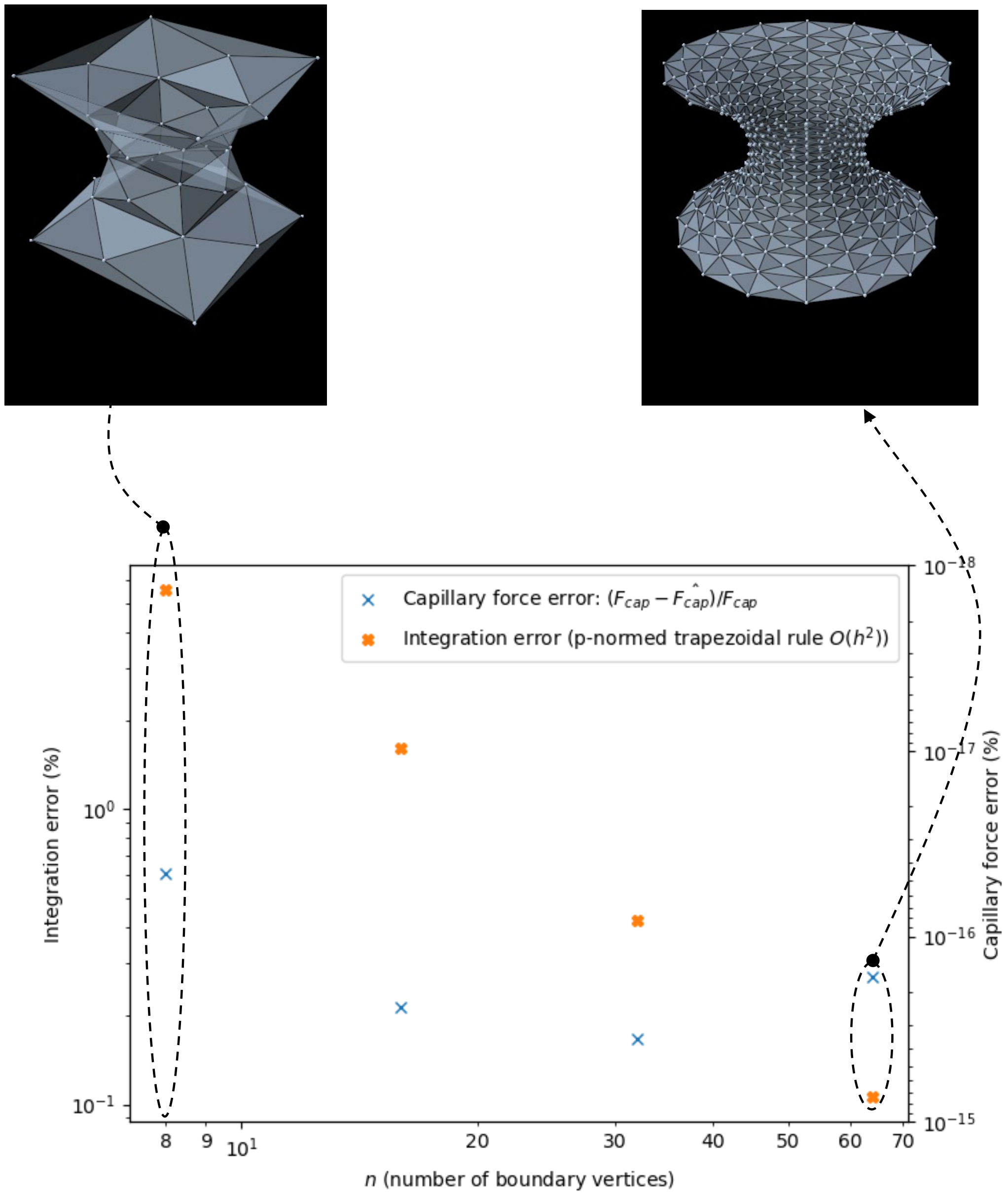

Figure 4.10.: The Capillary force errors of the numerically computed force when compared to the analytical result under increasingly detailed refinements of the surface. The integration error (calculated using the p-norms over the surface) is also shown and the computed convergence rate from a fitted trendline is $\approx$ 1.9048.

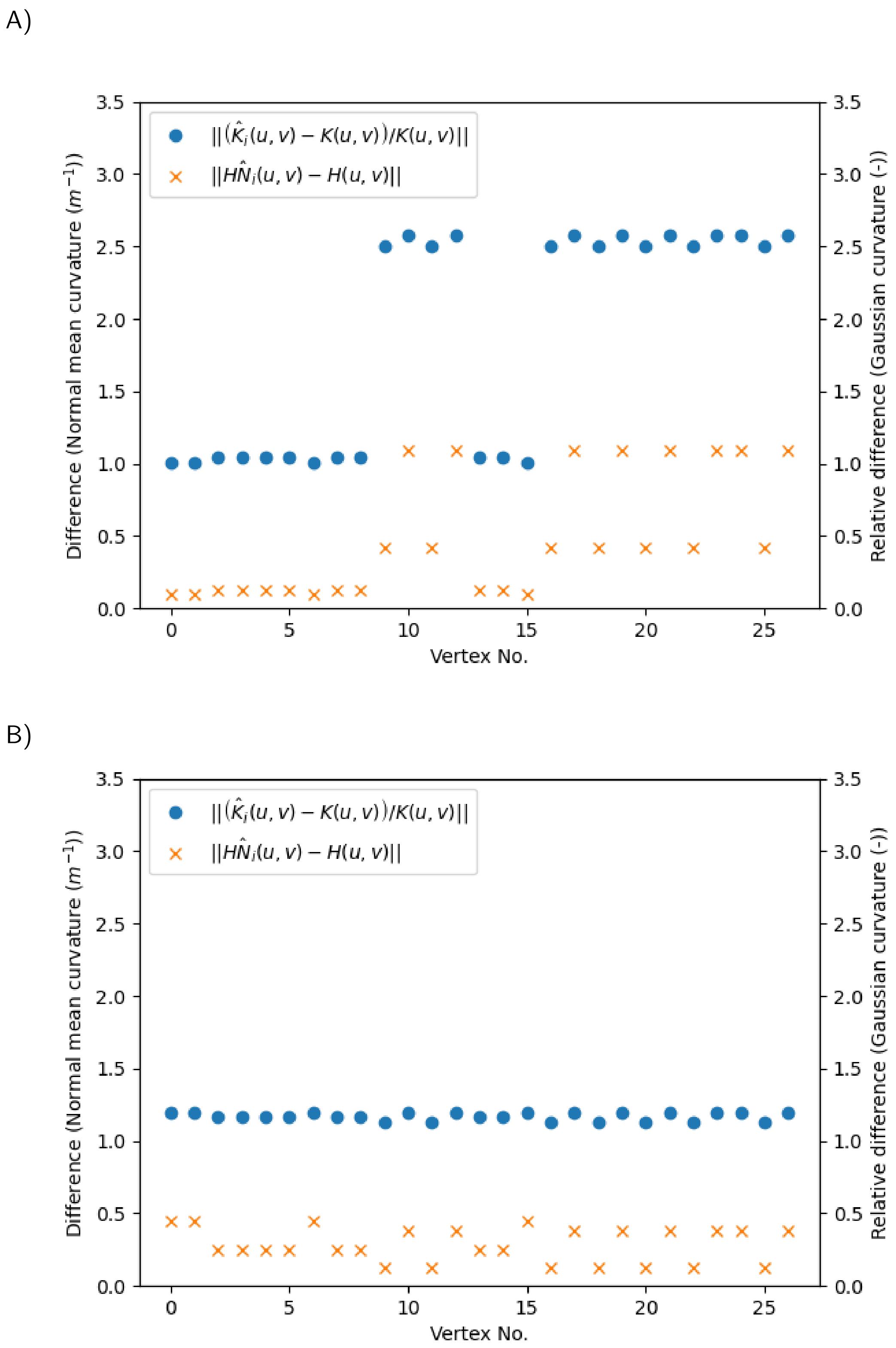

Figure 4.11.: A) The difference between the computed numerical mean curvature $\widehat{H}_i$ at a point u, v and the analytical solution $H(u, v) = 0$ at the point i. It is expected that every value (orange cross) should be close to zero, also subject to the error of the estimated surface normal $\mathbf{N_i}$ used to compute $\widehat{H}_i$. B) The greater refinement of the complex reduces the error compared to (A), which uses a sparser refinement (only the first few vertices are shown for comparison).

4.2.2 Asymmetric liquid bridges between two particles of the different sizes

4.2.2.1 Description of analytical test cases

In order to demonstrate the explosion in errors associated with point-wise estimates of curvature an asymmetrically triangulated mesh on the Catenoid surfaces shown in Figure 4.16 is used. The Catenoid surface is a minimal surface that has curvature $H = 0$, this presents to most ideal case possible to compute numerical methods in liquid-bridge like interfaces. The Catenoid surface has a mapping from the plane to an embedded surface $f : \mathbb{R}^2 \to \mathbb{R}^3$:

$$\begin{aligned} x(a,b) &= q \cdot \cosh\left(\frac{a}{q}\right) \cdot \cos a \\ y(a,b) &= q \cdot \cosh\left(\frac{a}{q}\right) \cdot \sin a, a \in [0, 2\pi), b \in \mathbb{R} \\ z(a,b) &= b. \end{aligned} \tag{4.4}$$

And a newly derived asymmetric variant is given by:

$$\begin{aligned} x(a,b) &= q \cdot \cosh\left(\tfrac{b}{q}\right) \cdot \cos a + w \cdot \cos a \\ y(a,b) &= q \cdot \cosh\left(\tfrac{b}{q}\right) \cdot \sin a + w \cdot \sin a \quad , a \in [0, 2\pi), b \in \mathbb{R} \\ z(a,b) &= b, \end{aligned} \tag{4.5}$$

where q and w represent surface constants related to the bridge volume and particle film height at the bridge boundary. Triangulation of the Catenoid surface involves sampling points on the surface using algorithm 2. Asymmetries are introduced by refining the triangles that are incident on only one of the boundary vertices (Figure 4.13). This results in both asymmetries in the local areas and asymmetric mesh elements such as "skinny triangles" with two edge lengths significantly longer than the remaining edge. Finally, mean normal curvatures are computed using Equation 3.5 and compared to the analytical solution. For the point-wise error estimates, local dual areas C_i as well as errors for unnormalised integrated estimates are computed by subtracting the analytical solution from the vector $\left\| \int_C H\boldsymbol{N}dA - \widehat{HNdA_i} \right\|$ and then summing the vector components to obtain a scalar value.

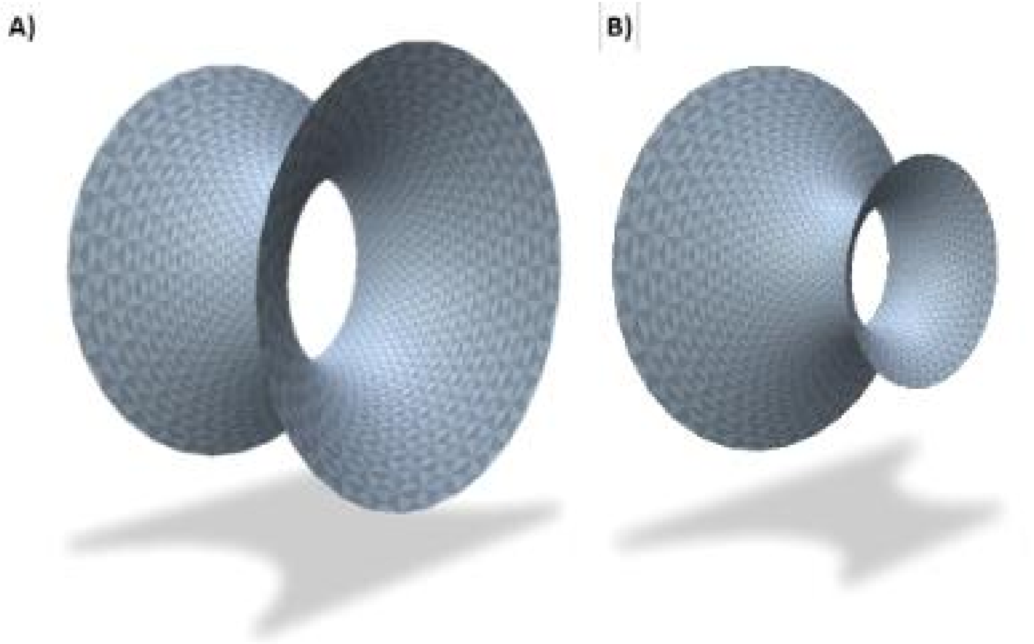

Figure 4.12.: A) A Catenoid surface is a minimal surface ($H = 0$) at every point together an analytical expression for the normal vector $\boldsymbol{N}$ at every point on the surface. B) An asymmetric Catenoid surface for liquid bridges connecting particles of different diameters.

4.2.2.2 Numerical results

The numerical results are presented in Figure 4.15 and Figure 4.16. It is immediately evident that point-wise estimates of curvature in the asymmetric direction $(0, 0, 1)$ become unacceptably large with increasing refinement. In contrast, the point-wise estimates of curvature for the regular symmetric triangles in the $(0, 0, -1)$ direction exhibit relatively high accuracy. Importantly, the integrated mean normal vector elements are always exact, subject only to floating-point errors, even when integrated over asymmetrically discretised triangulations. This highlights the utility of employing the latter formulation for arbitrary meshes.

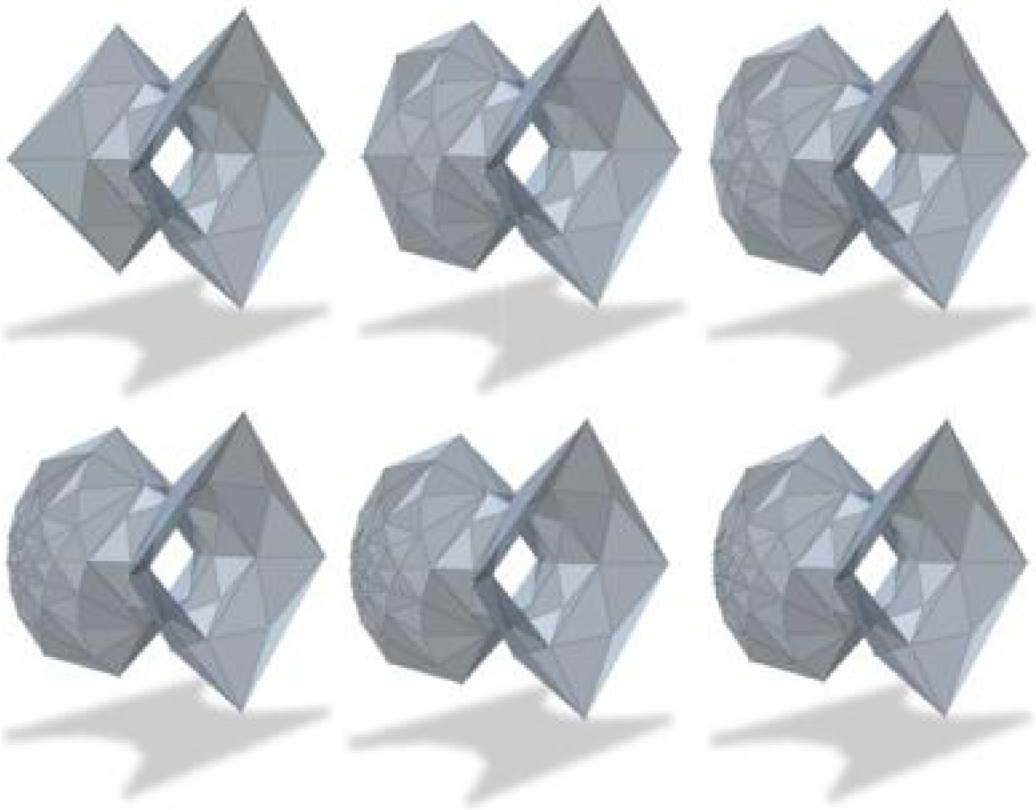

Figure 4.13.: Asymmetric refinement of the discretised Catenoid surface shown in Figure 4.12A. Note that refining only around one of the boundary vertices produces so-called "skinny triangles" which are asymmetric and result in numerical instabilities when attempting to compute point-wise curvature estimates. As demonstrated later in this section in Figure 4.15 these produce incorrect estimates of point-wise curvatures for which the error increases considerably at greater refinements. The triangulations contain 36, 45, 54, 63, 72 and 81 vertices respectively. The initial triangulation contains 16 vertices on the right side of the bridge and 4 in the centre. Therefore, the remainder of vertices generated are the number of asymmetrically generated vertices by subdividing every edge in every triangle connected to left-centre most vertex.

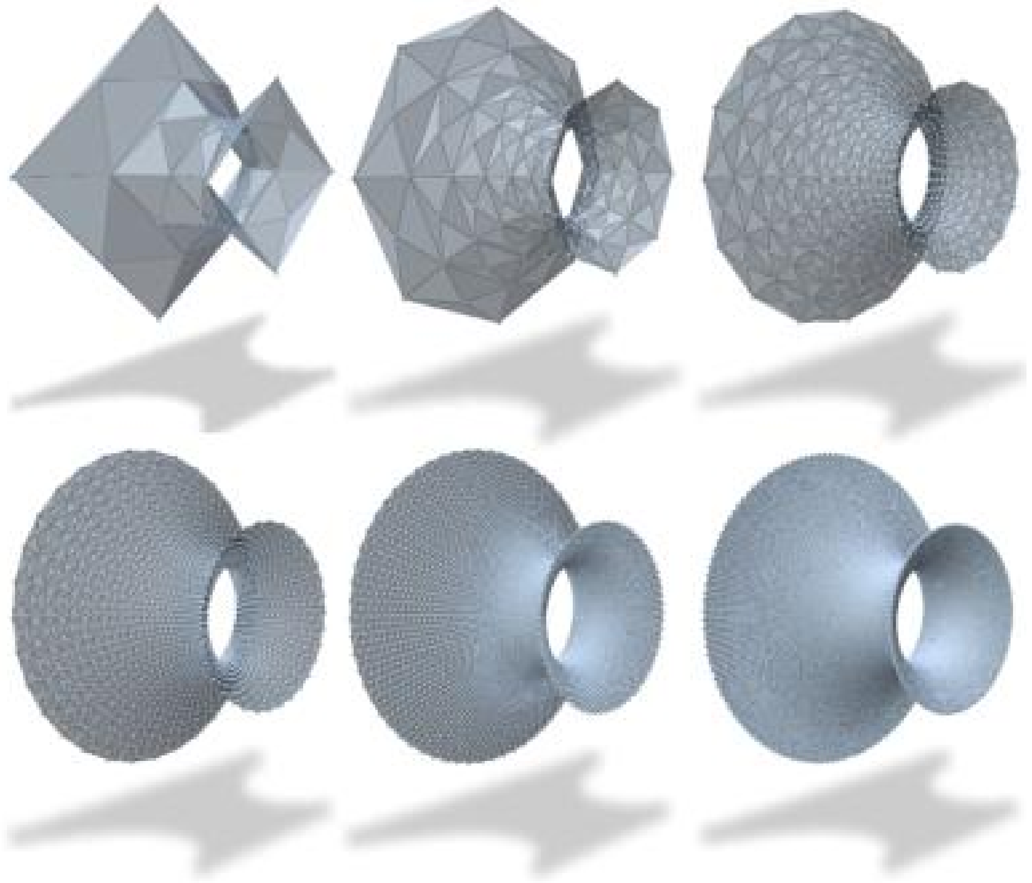

Figure 4.14.: Refinement of the discretised Catenoid surface shown in Figure 4.12B. The triangulations contain 36, 136, 528, 2080, 8256 and 32896 vertices respectively. The vertices are generated by subdividing in the initial plane triangulation and projecting the new plane triangulation to 3 dimensions.

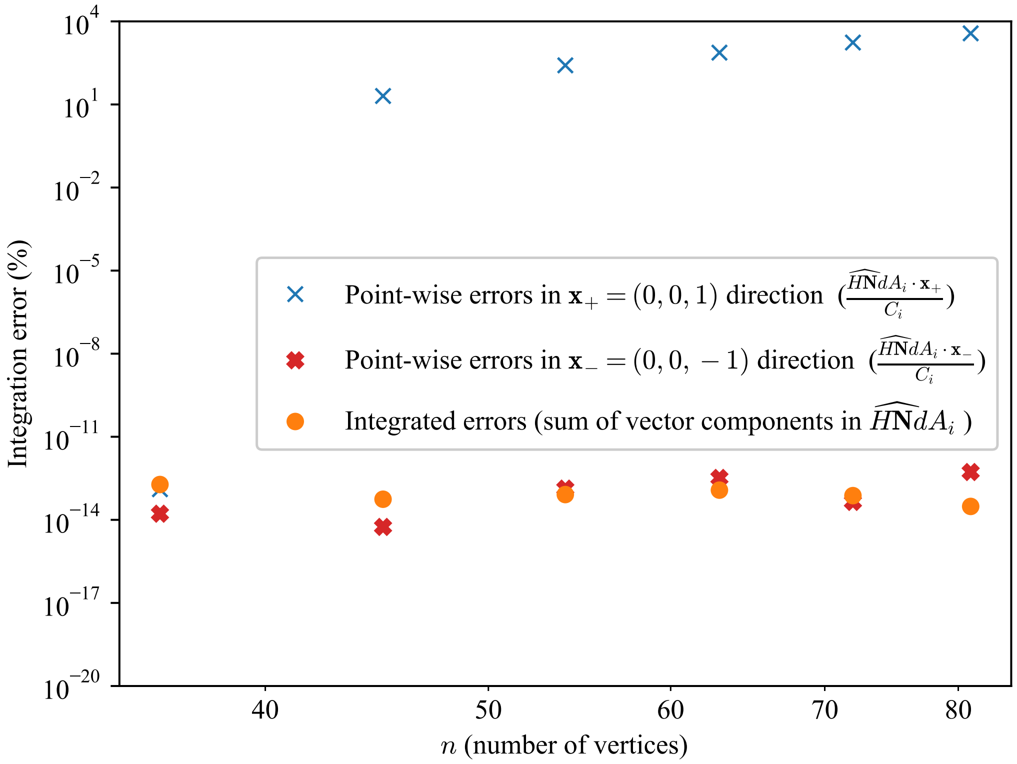

Figure 4.15.: The errors present in point-wise and unnormalised integrated estimates of the mean normal curvature of the asymmetrically refined liquid bridges shown in Figure 4.13. For the first point containing only 36 vertices the errors are all low because the triangulation is still symmetric (cf. Figure 4.13). For subsequent refinements the point-wise errors in the $(0, 0, 1)$ direction accumulate rapidly, at 81 vertices they are several orders large, this is primarily due to the small area of C_i in the asymmetric regions of refinement, which exacerbates the incorrect curvature estimates. On the contrary the total integrated point-wise error in the symmetric $(0, 0, -1)$ direction as well as the total integrated errors overall are nearly zero around the order of 10^{-17} largely indicating staggered floating-point error. The key observation is that the integrated errors $\widehat{HNdA_i}$ are always low independent of the chosen direction or quality of triangulation refinement.

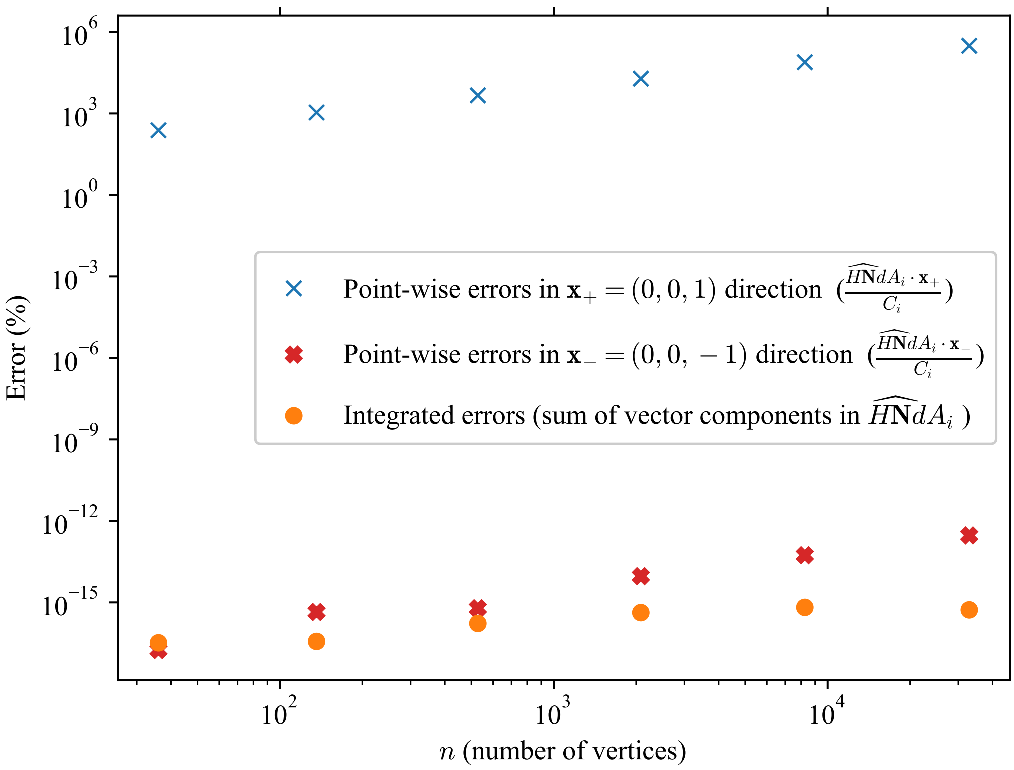

Figure 4.16.: The errors present in point-wise and unnormalised integrated estimates of the mean normal curvature of the symmetrically refined asymmetric liquid bridges shown in Figure 4.14. Note that similar to the test case in Figure 4.15 the point-wise errors in the $(0,0,1)$ direction accumulate rapidly as the area of C_i decreases. On the contrary the total integrated point-wise error in the symmetric $(0,0,-1)$ direction as well as the total integrated errors overall are nearly zero around the order of 10^{-15} largely indicating staggered floating-point error.

4.3 Conclusions

In concluding this chapter the infeasibility of using point-wise methods in the numerical approximation of curvatures has been demonstrated. It was also demonstrated that only using the total integral formulation allows for the precise values of mean normal curvature to be obtained (subject only to floating point errors) under any arbitrary mesh refinement including large mesh asymmetries and asymmetric surfaces providing a numerical validation the mathematical proof of Hypothesis H_2 derived in Section 3. Therefore, any implementation of DDG and closely related methods for curvature computation in future numerical frameworks simulating multiphase flow should opt to use the total integral formulation such as Equation 3.5 for computing surface tension forces acting on interfaces.

Outlook 5

Outlook on future applications of three-phase DDG formulations

The results of this work provides a strong foundation to open many new avenues of research that can be investigated within the field of multiphase CFD, particle technology and thermodynamics.

5.1 Hybrid Euler-Lagrangian CFD-DEM methods

As discussed in this dissertation many multiphase CFD frameworks can incorporate the formulation discussed here and this has already successfully been implemented and will continue to be developed in the near future. Surface detection and computation of curvature was implemented in the Material Point Method (MPM) for a solver using the *taichi* library as a back-end of the gradient computations that can take advantage of powerful mixed precision TPU units for the triangulation step. This involves using the boundary computation algorithm described in Section 3.2.3. Once the surface is constructed the curvature can be compute using Equation 3.5 and the resulting forces on the fluids is added directly to momentum balance as discussed in Section 2.3.2. The parallelisation of the boundary computation and parallel higher precision curvature computation improved speed performance by 62% in numerical tests when compared to CPU implementations. This demonstrates the potential to keep the real time speeds of MPM, but coupled with a rigorous method for surface tension computation.

In collaboration with the Technical University of Munich (TUM) the DDG formulation developed here has been implemented in the ALPACA library which used the level-set method for interface tracking. At the time of writing validation is being done on this implementation which should allow for multiphase simulation of a wide range of systems including systems with pressure shock waves where the DDG formulation has been shown to be robust even under pressure high gradients.

5.2 Equilibrium structures from fluid-solid potential models

While dynamic simulation can provide valuable scientific insight into processes, there are many practical engineering applications where it is only desired to predict the final equilibrium of the system. For example, the structures of nanoparticulate films which determine the performance of the material in applications such as batteries, catalysts and sensors where the percolation and porosity of the micro- and mesoscale structure material determine its macroscopic properties. In this case it is far cheaper to directly compute the equilibrium structure of the fluid film under different conditions of humidity. Current research attempts to understand how to incorporate the wetting of the solid particles into the energy minimisation equation. Coupled with the force models already available for the particles (cf. Section A) the structural minima of large scale fluid-particle systems can be computed efficiently.

5.3 Shearing stress models in particle-particle liquid bridges for DEM simulations

As demonstrated in Section 4.2 the method developed here can be used to find the equilibrium surface of particle-particle liquid bridges in arbitrary configuration. Currently the method is being used to investigate models of surface tension forces in shearing bridges. Course grained models developed from the solutions of these bridges can be used in the Torsion balance of large scale DEM simulations as discussed in Section B.

Conclusions 6

Concluding remarks

In this work a new method for computing curvatures and integrated three-phase contact angles has been developed. The central hypotheses have been validated and a new practical tool was developed that can be implemented in a wide array of state-of-the art methods. Firstly, it was shown in Chapter 4 that the rigorous derivation of the curvature of interfaces allows for an accurate, or near-exact computation of curvature for simulation of three-phase systems. It was additionally shown that the trapezoidal integration error can be used to provide an estimate of continuity of the interface, which is useful for ensuring dynamic accuracy in three-phase simulations by controlling the refinement of a numerically captured fluid-fluid interface that is manifold.

Secondly, it was shown that the method for computing curvatures developed for Hypothesis 1 can be computed exactly for arbitrary fluid-fluid interfaces that are manifold. The extent to which an interface is manifold can be tracked by computing the topology of the simplicial complex using the methods developed for the *hyperct* library back-end of *ddgclib*. Because the method for computing curvatures are near-exact and mesh independent, a very large tolerance can be used in large scale simulation of complex three-phase materials. In principle it is only necessary to track that the interface is continuous. Therefore, the complex can be refined in such a way that the error matches the desired tracking of non-solid volume due to porosity and asperities in three-phase systems.

The method has been validated on several important fundamental test cases:

1. Capillary rise in a tube with comparison to analytical solutions.
2. Sessile microdroplet validation with experiment and comparison to the state-of-the-art Surface Evolver software.

3. Symmetric and asymmetric particle-particle liquid bridges.

It has been shown that *ddgclib* can be used for simulation of energy minimising systems to find the equilibrium solution of three-phase systems. This has immediate use for applications such as DEM parameterisation as well as gaining insights into the equilibria of multiphase systems. The methods developed in this work has also been shown to be implementable in a wide range of CFD methods, especially multiphase methods that rely on a sharp or Lagrangian interpretation of the interface. The near-exact accuracy and low computational cost presents a core advantage of implementing DDG in a framework. This further ensures that validation of other features does not rely on the accuracy of the curvature estimation and this allows engineers and physicists working in the field to validate other important multiphase models. Future work will focus on the implementation of these methods in new hybrid Euler-Lagrange solvers for simulation of multiphase systems using CFD-DEM methods.

Bibliography

Alencar, Adriano M, Arnab Majumdar, Zoltan Hantos, Sergey V Buldyrev, H Eugene Stanley, and Béla Suki (2005). „Crackles and instabilities during lung inflation“. In: *Physica A: Statistical Mechanics and its Applications* 357.1, pp. 18–26 (cit. on p. 84).

Anantharaman, Aditya, J Ruud van Ommen, and Jia Wei Chew (2015). „Minimum pickup velocity (U pu) of nanoparticles in gas–solid pneumatic conveying“. In: *Journal of Nanoparticle Research* 17, pp. 1–10 (cit. on p. 15).

Balakin, Boris V, Sergey Alyaev, Alex C Hoffmann, and Pawel Kosinski (2013). „Micromechanics of agglomeration forced by the capillary bridge: The restitution of momentum“. In: *AIChE Journal* 59.11, pp. 4045–4057 (cit. on p. 84).

Barber, C Bradford, David P Dobkin, and Hannu Huhdanpaa (1996). „The quickhull algorithm for convex hulls“. In: *ACM Transactions on Mathematical Software (TOMS)* 22.4, pp. 469–483 (cit. on pp. 44, 45).

Baric, V., L. C. Ciacchi, and L. Mädler (2019). „Compaction-induced restructuring of aggregated nanoparticle films using the discrete element method“. In: *Powder Technology* 342, pp. 773–779 (cit. on p. 16).

Bauer, Ulrich (2021). „Ripser: efficient computation of Vietoris–Rips persistence barcodes“. In: *Journal of Applied and Computational Topology* 5.3, pp. 391–423 (cit. on p. 45).

Bell, Nathan and Anil N Hirani (2012). „PyDEC: software and algorithms for discretization of exterior calculus“. In: *ACM Transactions on Mathematical Software (TOMS)* 39.1, pp. 1–41 (cit. on pp. 34, 43).

Boissonnat, Jean-Daniel and Clément Maria (2014). „The simplex tree: An efficient data structure for general simplicial complexes“. In: *Algorithmica* 70.3, pp. 406–427 (cit. on pp. 43, 44).

Brackbill, Jeremiah U, Douglas B Kothe, and Charles Zemach (1992). „A continuum method for modeling surface tension“. In: *Journal of computational physics* 100.2, pp. 335–354 (cit. on pp. 16, 17).

Brakke, Kenneth A (1992). „The surface evolver“. In: *Experimental mathematics* 1.2, pp. 141–165 (cit. on pp. 74, 79, 81).

Caenorhabditis elegans (neural) network dataset – KONECT (Jan. 2018) (cit. on p. 49).

Carrière, Mathieu, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda (2020). „PersLay: a neural network layer for persistence diagrams and new graph topological signatures“. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2786–2796 (cit. on p. 43).

Chazal, Frédéric (2017). „High-dimensional topological data analysis“. In: *Handbook of discrete and computational geometry*. Chapman and Hall/CRC, pp. 663–683 (cit. on p. 43).

Chen, Liangbiao (2013). „Using the generalized interpolation material point method for fluid-solid interactions induced by surface tension“. PhD thesis. University of Alaska Fairbanks (cit. on p. 13).

Crane, Keenan (2018). „Discrete differential geometry: An applied introduction“. In: *Notices of the AMS, Communication* 1153 (cit. on pp. 56, 155, 158).

De Loera, Jesus A., Elisha Peterson, and Francis Edward Su (2002). „A Polytopal Generalization of Sperner's Lemma“. In: *Journal of Combinatorial Theory, Series A* 100.1, pp. 1–26 (cit. on p. 43).

De Vaucorbeil, Alban, Vinh Phu Nguyen, Sina Sinaie, and Jian Ying Wu (2020). „Material point method after 25 years: Theory, implementation, and applications“. In: *Advances in applied mechanics* 53, pp. 185–398 (cit. on p. 13).

Delaunay, Ch. (1841). „Sur la surface de révolution dont la courbure moyenne est constante.“ fre. In: *Journal de Mathématiques Pures et Appliquées*, pp. 309–314 (cit. on p. 84).

Diez, Eugen, Paul Kieckhefen, Katja Meyer, Andreas Bück, Evangelos Tsotsas, and Stefan Heinrich (2019). „Particle dynamics in a multi-staged fluidized bed: Particle transport behavior on micro-scale by discrete particle modelling“. In: *Advanced powder technology* 30.10, pp. 2014–2031 (cit. on p. 15).

Do Carmo, Manfredo P (2016). *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications (cit. on pp. 21, 22).

Dörmann, M. and H. J. Schmid (2014). „Simulation of capillary bridges between nanoscale particles.“ In: *Langmuir* 30.4, pp. 1055–1062 (cit. on p. 9).

Dyer, Ramsay and Scott Schaefer (2009). „Circumcentric dual cells with negative area“. In: *Technical ReportTR2009-06, School of Computing Science, Simon Fraser University* (cit. on p. 59).

Edelsbrunner, Herbert and John Harer (2010). *Computational topology: an introduction*. American Mathematical Soc. (cit. on p. 43).

Edelsbrunner, Herbert, David Letscher, and Afra Zomorodian (2000). „Topological persistence and simplification“. In: *Proceedings 41st annual symposium on foundations of computer science*. IEEE, pp. 454–463 (cit. on p. 43).

Elcott, Sharif, Yiying Tong, Eva Kanso, Peter Schröder, and Mathieu Desbrun (2007). „Stable, circulation-preserving, simplicial fluids“. In: *ACM Transactions on Graphics (TOG)* 26.1, 4–es (cit. on pp. 34, 43).

Elliott, Janet AW (2020). „Gibbsian surface thermodynamics“. In: *The Journal of Physical Chemistry B* 124.48, pp. 10859–10878 (cit. on p. 157).

Endres, Stefan C, Carl Sandrock, and Walter W Focke (2018). „A simplicial homology algorithm for Lipschitz optimisation“. In: *Journal of Global Optimization* 72.2, pp. 181–217 (cit. on pp. 32, 37, 39, 43, 44).

Endres, Stefan Christian, Marc Avila, and Lutz Mädler (2022). „A discrete differential geometric formulation of multiphase surface interfaces for scalable multiphysics equilibrium simulations“. In: *Chemical Engineering Science* 257, p. 117681 (cit. on pp. 32, 66, 69, 70, 157).

Endres, Stefan Christian, Lucio Colombi Ciacchi, and Lutz Mädler (2021). „A review of contact force models between nanoparticles in agglomerates, aggregates, and films“. In: *Journal of Aerosol Science* 153, p. 105719 (cit. on pp. 7–10, 12, 123, 150).

Endres, Stefan Christian, Lucio Colombi Ciacchi, and Lutz Mädler (2021). „A review of contact force models between nanoparticles in agglomerates, aggregates, and films“. In: *Journal of Aerosol Science* 153, p. 105719 (cit. on p. 51).

Fabre, A., A. Clemente, F. Balas, et al. (2017). „Modeling the size distribution in a fluidized bed of nanopowder“. In: *Environmental Science: Nano* 4.3, pp. 670–678 (cit. on p. 15).

Fabre, A., S. Salameh, L. C. Ciacchi, M. T. Kreutzer, and J. R. van Ommen (2016). „Contact mechanics of highly porous oxide nanoparticle agglomerates“. In: *Journal of Nanoparticle Research* 18.7, pp. 1–13 (cit. on p. 15).

Fadnavis, Shreyas, Stefan Endres, Qiuting Wen, et al. (2021). „Bifurcated Topological Optimization for IVIM“. In: *Frontiers in neuroscience* 15 (cit. on p. 43).

Freedman, Daniel (2002). „Efficient simplicial reconstructions of manifolds from their samples“. In: *IEEE transactions on pattern analysis and machine intelligence* 24.10, pp. 1349–1357 (cit. on p. 43).

Gauss, Carl Friedrich (1877). *Principia generalia theoriae figurae fluidorum in statu aequilibrii*. Springer (cit. on p. 12).

Grinspun, Eitan, Mathieu Desbrun, Konrad Polthier, Peter Schröder, and Ari Stern (2006). „Discrete differential geometry: an applied introduction“. In: *ACM Siggraph Course* 7.1 (cit. on p. 59).

Hager, Alice, Christoph Kloss, and Christoph Goniva (2018). „Combining Open Source and Easy Access in the field of DEM and coupled CFD-DEM: LIGGGHTS®, CFDEM® coupling and CFDEM® workbench“. In: *Computer Aided Chemical Engineering*. Vol. 43. Elsevier, pp. 1699–1704 (cit. on p. 14).

Harlow, Francis H and J Eddie Welch (1965). „Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface“. In: *The physics of fluids* 8.12, pp. 2182–2189 (cit. on p. 16).

Hatcher, Allen (2002). *Algebraic topology*. Cambridge: Cambridge University Press, pp. xii+544 (cit. on pp. 29, 30).

Henle, M (1979). *A Combinatorial Introduction to Topology*. Unabriged Dover (1994) republication of the edition published by WH Greeman & Company, San Francisco, 1979 (cit. on pp. 19, 29–31, 33, 44).

Herminghaus, Stephan, Ciro Semprebon, and Martin Brinkmann (2019). „Capillary Interaction in Wet Granular Assemblies: Part 1“. In: *Particles in Contact: Micro Mechanics, Micro Process Dynamics and Particle Collective*, pp. 239–275 (cit. on p. 9).

Hirt, C.W. and B.D. Nichols (1981). „Volume of fluid (VOF) method for the dynamics of free boundaries“. In: *J Comp Phys* 39, pp. 201–225 (cit. on p. 14).

Hirt, Cyrill W, Anthony A Amsden, and JL Cook (1974). „An arbitrary Lagrangian-Eulerian computing method for all flow speeds“. In: *Journal of computational physics* 14.3, pp. 227–253 (cit. on p. 16).

Hofer, Christoph, Roland Kwitt, Marc Niethammer, and Mandar Dixit (2019). „Connectivity-optimized representation learning via persistent homology“. In: *International Conference on Machine Learning*. PMLR, pp. 2751–2760 (cit. on p. 43).

Hofer, Christoph, Roland Kwitt, Marc Niethammer, and Andreas Uhl (2017). „Deep learning with topological signatures“. In: *arXiv preprint arXiv:1707.04041* (cit. on p. 43).

Hu, Xiaoling, Li Fuxin, Dimitris Samaras, and Chao Chen (2019). „Topology-preserving deep image segmentation“. In: *arXiv preprint arXiv:1906.05404* (cit. on p. 43).

Jasak, Hrvoje (1996). „Error analysis and estimation for the finite volume method with applications to fluid flows.“ In: (cit. on p. 16).

Jin, W., C. R. Kleijn, and J. R. van Ommen (2017). „Simulation of atomic layer deposition on nanoparticle agglomerates“. In: *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* 35, p. 1 (cit. on p. 15).

Joshi, Milan and Dhananjay Joshi (2019). „A survey of Topological Data Analysis Methods for Big Data in Healthcare Intelligence“. In: *Int. J. Appl. Eng. Res* 14, pp. 584–588 (cit. on p. 43).

Khrulkov, Valentin and Ivan Oseledets (2018). „Geometry score: A method for comparing generative adversarial networks“. In: *International Conference on Machine Learning*. PMLR, pp. 2621–2629 (cit. on p. 43).

Kwitt, Roland, Stefan Huber, Marc Niethammer, Weili Lin, and Ulrich Bauer (2015). „Statistical topological data analysis-a kernel perspective“. In: *Advances in neural information processing systems*, pp. 3070–3078 (cit. on p. 43).

Laube, J., Dörmann, Schmid M., H. J., L. Mädler, and L. Colombi Ciacchi (2017). „Dependencies of the Adhesion Forces between TiO2 Nanoparticles on Size and Ambient Humidity.“ In: *Journal of Physical Chemistry C* 121.28, pp. 15294–15303 (cit. on p. 10).

Laube, Jens, Valentin Baric, Samir Salameh, Lutz Mädler, and Lucio Colombi Ciacchi (2018). „A new contact model for the discrete element method simulation of TiO 2 nanoparticle films under mechanical load“. In: *Granular Matter* 20.2, p. 28 (cit. on pp. 15, 150).

Lee, John M (2003). *Introduction to Smooth Manifolds* (cit. on p. 18).

Lee, Yongjin, Senja D Barthel, Paweł Dłotko, Seyed Mohamad Moosavi, Kathryn Hess, and Berend Smit (2018). „High-throughput screening approach for nanoporous materials genome using topological data analysis: application to zeolites“. In: *Journal of chemical theory and computation* 14.8, pp. 4427–4437 (cit. on p. 43).

Li, Guihong, Sumit K. Mandal, Umit Y. Ogras, and Radu Marculescu (2021). „FLASH: Fast Neural Architecture Search with Hardware Optimization“. In: *ACM Trans. Embed. Comput. Syst.* 20.5s (cit. on p. 43).

Liu, D., B. G. M. van Wachem, R. F. Mudde, X. Chen, and J. R. van Ommen (2016). „Characterization of fluidized nanoparticle agglomerates by using adhesive CFD-DEM simulation“. In: *Powder Technology* 304, pp. 198–207 (cit. on p. 15).

Liu, Hailong and Guoxin Cao (2016). „Effectiveness of the Young-Laplace equation at nanoscale“. In: *Scientific reports* 6.1, p. 23936 (cit. on p. 10).

Lütgehetmann, Daniel, Dejan Govc, Jason P Smith, and Ran Levi (2020). „Computing persistent homology of directed flag complexes“. In: *Algorithms* 13.1, p. 19 (cit. on pp. 45, 46).

MacDowell, Luis G (2017). „Capillary wave theory of adsorbed liquid films and the structure of the liquid-vapor interface“. In: *Physical Review E* 96.2, p. 022801 (cit. on p. 12).

MacDowell, Luis G, Pablo Llombart, Jorge Benet, Jose G Palanco, and Andres Guerrero-Martinez (2018). „Nanocapillarity and liquid bridge-mediated force between colloidal nanoparticles“. In: *ACS omega* 3.1, pp. 112–123 (cit. on p. 12).

Mädler, L and S Friedlander (2007). „Transport of Nanoparticles in Gases: Overview and Recent Advances.“ In: *Aerosol and Air Quality Research* 7.3, pp. 304–342 (cit. on p. 15).

Maria, Clément, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec (2014). „The gudhi library: Simplicial complexes and persistent homology“. In: *International Congress on Mathematical Software*. Springer, pp. 167–174 (cit. on p. 45).

Martín, De, Bouwman L., W. G., and J. R. van Ommen (2014). „Multidimensional nature of fluidized nanoparticle agglomerates“. In: *Langmuir* 30.42, pp. 12696–12702 (cit. on p. 15).

Meunier, Frédéric (2006). „Sperner labellings : A combinatorial approach“. In: 113, pp. 1462–1475 (cit. on p. 43).

Meyer, Mark, Mathieu Desbrun, Peter Schröder, and Alan H Barr (2003). „Discrete differential-geometry operators for triangulated 2-manifolds“. In: *Visualization and mathematics III*. Springer, pp. 35–57 (cit. on p. 34).

Misiulia, Dzmitry, Göran Lidén, and Sergiy Antonyuk (2021). „Evolution of turbulent swirling flow in a small-scale cyclone with increasing flow rate: A LES study“. In: *Flow, Turbulence and Combustion* 107, pp. 575–608 (cit. on pp. 15, 153).

Misiulia, Dzmitry, Praveen Kumar Nedumaran, and Sergiy Antonyuk (2023). „Effect of the discharging flap on particle separation in a cyclone“. In: *Chemical Engineering & Technology* (cit. on pp. 15, 153).

Misztal, Marek Krzysztof, Kenny Erleben, Adam Bargteil, et al. (2013). „Multiphase flow of immiscible fluids on unstructured moving meshes“. In: *IEEE transactions on visualization and computer graphics* 20.1, pp. 4–16 (cit. on p. 43).

Mogre, C., A. U. Thakurdesai, J. R. van Ommen, and S. Salameh (2017). „Long-term fluidization of titania nanoparticle agglomerates“. In: *Powder Technology* 316, pp. 441–445 (cit. on p. 15).

Mohamed, Mamdouh S, Anil N Hirani, and Ravi Samtaney (2016). „Comparison of discrete Hodge star operators for surfaces“. In: *Computer-Aided Design* 78, pp. 118–125 (cit. on pp. 34, 43).

Moor, Michael, Max Horn, Bastian Rieck, and Karsten Borgwardt (2020). „Topological autoencoders“. In: *International Conference on Machine Learning*. PMLR, pp. 7045–7054 (cit. on p. 43).

Mullen, Patrick, Keenan Crane, Dmitry Pavlov, Yiying Tong, and Mathieu Desbrun (2009). „Energy-preserving integrators for fluid animation“. In: *ACM Transactions on Graphics (TOG)* 28.3, pp. 1–8 (cit. on pp. 34, 43).

Murray, Brandon, Matthew J Fox, and Shankar Narayanan (2020). „Quantifying the evaporation rate of sessile droplets using a quartz crystal microbalance". In: *Journal of Applied Physics* 128.3 (cit. on p. 79).

Musin, Oleg R (2015). „Extensions of Sperner and Tucker ' s lemma". In: *Journal of Combinatorial Theory, Series A* 132, pp. 172–187 (cit. on p. 43).

Nguyen, Thien, Kstutis Karčiauskas, and Jörg Peters (2014). „A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk". In: *Axioms* 3.2, pp. 280–300 (cit. on p. 59).

Noh, William F and Paul Woodward (1976). „SLIC (simple line interface calculation)". In: *Proceedings of the fifth international conference on numerical methods in fluid dynamics June 28–July 2, 1976 Twente University, Enschede*. Springer, pp. 330–340 (cit. on p. 14).

Pampaloni, Francesco and Ernst HK Stelzer (2009). „Three-dimensional cell cultures in toxicology". In: *Biotechnology and Genetic Engineering Reviews* 26.1, pp. 117–138 (cit. on p. 84).

Paulavičius, Remigijus, Yaroslav D Sergeyev, Dmitri E Kvasov, and Julius Žilinskas (2014). „Globally-biased DISIMPL algorithm for expensive global optimization". In: *Journal of Global Optimization* 59.2-3, pp. 545–567 (cit. on p. 43).

Paulavičius, Remigijus and Julius Žilinskas (2014). *Simplicial global optimization*. Springer (cit. on p. 43).

Pei, C., C. Y. Wu, M. Adams, D. England, S. Byard, and H. Berchtold (2015). „Contact electrification and charge distribution on elongated particles in a vibrating container". In: *Chemical Engineering Science* 125, pp. 238–247 (cit. on p. 15).

Pei, C., C. Y. Wu, D. England, S. Byard, H. Berchtold, and M. Adams (2013). „Numerical analysis of contact electrification using DEM-CFD". In: *Powder Technology* 248, pp. 34–43 (cit. on p. 15).

Petrova, Tamara and RB Dooley (2014). „The International Association for the Properties of Water and Steam". In: *Revised Release on Surface Tension of Ordinary Water Substance. Moscow, Russia* (cit. on p. 79).

Pinkall, Ulrich and Konrad Polthier (1993). „Computing discrete minimal surfaces and their conjugates". In: *Experimental mathematics* 2.1, pp. 15–36 (cit. on p. 59).

Pun, Chi Seng, Kelin Xia, and Si Xian Lee (2018). „Persistent-Homology-based Machine Learning and its Applications–A Survey". In: *arXiv preprint arXiv:1811.00252* (cit. on p. 43).

Ramamurthy, Karthikeyan Natesan, Kush Varshney, and Krishnan Mody (2019). „Topological data analysis of decision boundaries with application to model selection". In: *International Conference on Machine Learning*. PMLR, pp. 5351–5360 (cit. on p. 43).

Rieck, Bastian, Christian Bock, and Karsten Borgwardt (2019). „A persistent weisfeiler-lehman procedure for graph classification". In: *International Conference on Machine Learning*. PMLR, pp. 5448–5458 (cit. on p. 43).

Rieck, Bastian, Matteo Togninalli, Christian Bock, et al. (2018). „Neural persistence: A complexity measure for deep neural networks using algebraic topology". In: *arXiv preprint arXiv:1812.09764* (cit. on p. 43).

Rieck, Bastian, Tristan Yates, Christian Bock, et al. (2020). „Uncovering the topology of time-varying fmri data using cubical persistence“. In: *arXiv preprint arXiv:2006.07882* (cit. on p. 43).

Roenby, Johan, Henrik Bredmose, and Hrvoje Jasak (2019). „IsoAdvector: Geometric VOF on general meshes“. In: *OpenFOAM®: Selected Papers of the 11th Workshop*. Springer, pp. 281–296 (cit. on p. 14).

Royer, Martin, Frédéric Chazal, Clément Levrard, Yuhei Umeda, and Yuichi Ike (2021). „ATOL: Measure vectorization for automatic topologically-oriented learning“. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1000–1008 (cit. on p. 43).

Salameh, S., J. Gómez-Hernández, A. Goulas, H. van Bui, and J. R. van Ommen (2017). „Advances in scalable gas-phase manufacturing and processing of nanostructured solids: A review“. In: *Particuology* 30, pp. 15–39 (cit. on p. 15).

Salikov, Vitalij, Sergiy Antonyuk, Stefan Heinrich, Vinayak S Sutkar, Niels G Deen, and JAM Kuipers (2015). „Characterization and CFD-DEM modelling of a prismatic spouted bed“. In: *Powder Technology* 270, pp. 622–636 (cit. on p. 15).

Schopf, S. O., A. Hartwig, U. Fritsching, and L. Mädler (2017). „Imbibition into Highly Porous Layers of Aggregated Particles“. In: *Transport in Porous Media* 119.1, pp. 119–141 (cit. on p. 16).

Schopf, S. O., S. Salameh, and L. Mädler (2013a). „Transfer of highly porous nanoparticle layers to various substrates through mechanical compression“. In: *Nanoscale* 5.9, pp. 3764–3772 (cit. on p. 15).

Schopf, Sven O, Samir Salameh, and Lutz Mädler (2013b). „Transfer of highly porous nanoparticle layers to various substrates through mechanical compression“. In: *Nanoscale* 5.9, pp. 3764–3772 (cit. on p. 150).

Schubert, H (1984). „Capillary forces-modeling and application in particulate technology“. In: *Powder Technology* 37.1, pp. 105–116 (cit. on p. 9).

Schubert, Heinrich (2012). *Handbuch der mechanischen Verfahrenstechnik*. John Wiley & Sons (cit. on p. 9).

Silva, Vin de and Gunnar Carlsson (2004). „Topological estimation using witness complexes“. In: *SPBG’04 Symposium on Point - Based Graphics 2004*. Ed. by Markus Gross, Hanspeter Pfister, Marc Alexa, and Szymon Rusinkiewicz. The Eurographics Association (cit. on p. 43).

Smith, Warren D (2000). „A lower bound for the simplexity of then-cube via hyperbolic volumes“. In: *European Journal of Combinatorics* 21.1, pp. 131–137 (cit. on p. 44).

Sutkar, Vinayak S, Niels G Deen, Amit V Patil, et al. (2016). „CFD–DEM model for coupled heat and mass transfer in a spout fluidized bed with liquid injection“. In: *Chemical Engineering Journal* 288, pp. 185–197 (cit. on p. 15).

Tong, Z., H. Kamiya, A. Yu, H. K. Chan, and R. Yang (2015). „Multi-scale modelling of powder dispersion in a carrier-based inhalation system“. In: *Pharmaceutical Research* 32.6, pp. 2086–2096 (cit. on p. 15).

Tong, Z. B., B. Zheng, R. Y. Yang, A. B. Yu, and H. K. Chan (2013). „CFD-DEM investigation of the dispersion mechanisms in commercial dry powder inhalers“. In: *Powder Technology* 240, pp. 19–24 (cit. on p. 15).

Townsend, Jacob, Cassie Putman Micucci, John H Hymel, Vasileios Maroulas, and Konstantinos D Vogiatzis (2020). „Representation of molecular structures with persistent homology for machine learning applications in chemistry“. In: *Nature communications* 11.1, pp. 1–9 (cit. on p. 43).

Wagner, Wolfgang and Andreas Pruß (2002). „The IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use“. In: *Journal of physical and chemical reference data* 31.2, pp. 387–535 (cit. on p. 79).

Walt, Stefan Van der, Johannes L Schönberger, Juan Nunez-Iglesias, et al. (2014). „scikit-image: image processing in Python“. In: *PeerJ* 2, e453 (cit. on p. 81).

Wardetzky, Max (2008). „Convergence of the cotangent formula: An overview“. In: *Discrete differential geometry*, pp. 275–286 (cit. on p. 70).

Watts, Duncan J. and Steven H. Strogatz (1998). „Collective Dynamics of 'Small-world' Networks“. In: *Nature* 393.1, pp. 440–442 (cit. on p. 49).

Wittel, Falk K, Roman Mani, Konstantin Melnikov, Filippo Bianchi, and Hans J Herrmann (2019). „Capillary interaction in wet granular assemblies: Part 2“. In: *Particles in Contact: Micro Mechanics, Micro Process Dynamics and Particle Collective*, pp. 277–309 (cit. on p. 9).

Yang, J., C.-Y. Wu, and M. Adams (2014). „Three-dimensional DEM–CFD analysis of air-flow-induced detachment of API particles from carrier particles in dry powder inhalers“. In: *Acta Pharmaceutica Sinica B* 4.1, pp. 52–59 (cit. on p. 15).

– (2015). „Numerical Modelling of Agglomeration and Deagglomeration in Dry Powder Inhalers: A Review“. In: *Current Pharmaceutical Design* 21.40, pp. 5915–5922 (cit. on p. 15).

Young, Thomas (1805). „An essay on the cohesion of fluids“. In: *Philosophical Transactions of the Royal Society of London* 95, 65–87 (cit. on p. 10).

Zhao, Qi and Yusu Wang (2019). „Learning metrics for persistence-based summaries and applications for graph classification“. In: *arXiv preprint arXiv:1904.12189* (cit. on p. 43).

Ziegler, Günter M, Peter Schröder, John M Sullivan, and Alexander I Bobenko (2008). *Discrete Differential Geometry (Oberwolfach seminars; v. 38)*. Birkhäuser Basel (cit. on p. 59).

Zong, Chuanming (2005). „What is known about unit cubes“. In: *Bulletin of the American Mathematical Society* 42.2, pp. 181–211 (cit. on p. 44).

List of Figures

List of Tables

A
Tables of nanoparticle contact models

In this appendix the tables of particle contact force models which were reviewed more extensively in Endres, Ciacchi, and Mädler (2021) is shown in order for this work to be self-contained. These tables are adapted almost verbatim from the article.

Table A.1.: Short range particle-particle repulsive forces

Model	Description	Parameters	Applications	Citation
Hertzian force model: $F_{\mathrm{nr}} = \frac{4}{3}E^*\sqrt{R^*}\delta^{\frac{3}{2}}$	Classical Hertzian force model	E^* - equivalent Young's modulus R^* - equivalent particle radius δ - particle interaction parameter	Simulation of silica aggregates of size 400nm	*Model:* (Mindlin & Deresiewicz, 1953; Morrisey, Thakur, & Ooi, 2014) *Applications:* (Schilde et al., 2012, 2014)
Hertz-Mindlin normal force model: $F_{nr} = +\frac{4E^*}{3R^*}a^3$	Hertz-Mindlin (repulsion component of J model)	a – contact radius	Conventional particle laden flows in the laminar regime.	Mindlin and Deresiewicz (1953)

Table A.1.: Short range particle-particle repulsive forces

Model	Description	Parameters	Applications	Citation
Water adsorbate Hertz-Mindlin force model: $F_{nr} = k_n \delta_n^{\frac{3}{2}} \sqrt{0.5\,(R + h_n)^*}$ *where* $(R + h_n)^* = \frac{2(R_i + h_n)(R_j + h_n)}{R_i + R_j + 2h_n}$	Adaption of Hertz stress model to include parameterised from MD simulations.	k_n - Hertzian contact stiffness δ_n- particle-particle overlap $(R + h_n)^*$ - the combined effective radius R_i – radius of particle i R_j – radius of particle j h_n – particle-particle separation distance at equilibrium. Related to the water adsorbate thickness h that adds up to each of the nominal particle radii $R_i$and R_j	Simulation of TiO_2 nanoparticle films subjected to mechanical stress.	(Laube et al., 2015, 2017, 2018b)

Table A.1.: Short range particle-particle repulsive forces

Model	Description	Parameters	Applications	Citation
Hertz modified for elasticity of non-adhesive nanopowders: $F_{\mathrm{nr}} = \frac{4}{3}\frac{E^*}{1-v^{*2}} R_{Tip}^{\frac{1}{2}} \left(s_0 - s\right)^{\frac{3}{2}}$	Variant of the Hertz model was used to model elasticity of various non-adhesive porous nanopowder agglomerates.	E^*- Young's modulus of the powder v^*- Poisson's ration of the powder R_{Tip}- Radius of the glass colloid, s_0 - colloid–sample contact distance s - penetration depth.	Used to model elasticity of various titania (TiO_2, 22 nm), alumina (Al_2O_3, 8 nm), and silica (SiO_2, 16 nm) nanopowder agglomerates.	(Fabre, et al., 2016)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
JKR cohesion model: $F_{na} = -4\sqrt{\pi\gamma E^*}a^{3/2}$ $\delta = \frac{a^2}{R^*} - \sqrt{4\pi\gamma a/E^*}$ *The equivalent parameters can be calculated from:* $\frac{1}{E^*} = \frac{\left(1-v_i^2\right)}{E_i} + \frac{\left(1-v_j^2\right)}{E_j}$ $\frac{1}{R^*} = \frac{1}{R_i} + \frac{1}{R_j}$	JKR (Johnson-Kendall-Robertson) model is a cohesion contact model which accounts for the influence of Van der Waals forces, many models present here have been adapted from JKR.	γ - surface energy E^* - equivalent Young's modulus R^* - equivalent particle radius δ - particle interaction parameter a – contact radius v_i, v_i– Poisson ratios of interacting particles i and j.	Used to model adhesive of microscopic agglomerates (5-200 μm. Used to study agglomerate breakage rates in Silica agglomerates 50-400 nm range	*Model:* (Johnson, et al., 1971) *Applications:* (DEM Solutions Ltd, 2014; Morrisey et al., 2014; Nguyen et al., 2014; Schilde et al., 2014)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Adhesion pull-off force: $F_{na} = -\frac{3}{2}\Gamma\pi R^*$ *Where* $\Gamma = \gamma_1 + \gamma_2 - \gamma_{12}$ $R^* = \frac{R_1 \times R_2}{R_1 + R_2}$	Simple model of adhesion pull-off force based on JKR (Johnson-Kendall-Robertson) theory	Γ - thermodynamic work of adhesion. R^* - effective radius between particles γ_1 , γ_2 – surface energies of particle i and particle j R_i, R_j– radii of particle i and particle j	Popular model used in most non-adhesive granular systems. Agglomerates built up from primary particles within a large diameter range of 9–11 μm. Applied in dry powder inhaler systems in CFD-DEM simulations	(Hertz, 1882; Johnson et al., 1971; Lian et al., 1998; Thornton et al., 1996; Yang et al., 2013, 2014, 2015a, 2015c, 2013a)
Combined capillary and solvation force model: $F_{na} = (F_{cap} + F_{solv}) \cdot \mathbf{n_{ij}}$ *The capillary and solvation forces can be computed from different models described here.*	Combined adhesive forces	$\mathbf{n_{ij}}$ – The normal vector between particles i and j	Simulation of TiO2 nanoparticle films subjected to mechanical stress.	(Laube et al., 2018a)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Solvation force model: $F_{solv} = F^0_{solv} \cos\left(\frac{2\pi(d-R_i-R_j)}{\sigma_{solv}}\right) e^{\frac{d-R_i-R_j}{\sigma_{solv}}}$	Solvation forces	F^0_{solv} – Semi-emperical parameter fitted to the oscillatory force peak σ_{solv} – decay length d – particle distance R_i – radius of particle i R_j – radius of particle j	Solvation forces between particles in the simulation of TiO_2 nanoparticle films subjected to mechanical stress.	*Model:* (Israelachvili, 2011) *Applications:* (Israelachvili, 2011; Laube et al., 2015, 2018a)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Capillary force model: $F_{cap} = \Delta P A_m + 2\gamma_L U_m$ $\Delta P = \frac{\gamma_L}{r_K}$ $r_K = \frac{\gamma_L V_M}{RT \ln\left(\frac{p}{P_0}\right)}$ *or* $r_K = \left(\frac{1}{r_m} - \frac{1}{l_m}\right)^{-1}$ $r_m = \frac{2R_l(1-\cos(\beta))+D}{2\cos(\Theta+\beta)}$ $l_m = R_1 \sin(\beta) - r_m\left[1 - \sin(\Theta + \beta)\right]$	Capillary force between spherical nanoparticles	ΔP – Laplacian pressure difference between the liquid phase of the meniscus and the surrounding gas phase γ_L - liquid/gas surface tension A_m- meniscus neck cross section U_m - meniscus neck circumference r_k - Kelvin radius Θ - contact angle β – filling angle R_l – Effective particle radius $= d/2 + h$ D – Effective particle-particle separation $D = \delta_n + 2h$ r_m- outer meniscus radius l_m- inner meniscus radius	Capillary forces between particles in the simulation of TiO_2 nanoparticle films subjected to mechanical stress.	(Laube et al., 2017)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Approximated capillary force model: $F_{cap} =$ $-\pi\gamma_{\mathrm{L}}R_l\sin(\beta)\left[2\sin(\Theta+\beta)+R_l\sin(\beta)\left(\frac{1}{r_{\mathrm{m}}}-\frac{1}{l_{\mathrm{m}}}\right)\right]$	Approximated capillary force between spherical nanoparticles	γ_L - liquid/gas surface tension U_m - meniscus neck circumference r_k - Kelvin radius Θ - contact angle β – filling angle R_l – Effective particle radius $= d/2\ +\ h$ D – Effective particle-particle separation $D = \delta_n + 2h$ r_m- outer meniscus radius l_m- inner meniscus radius		(Dörmann & Schmid, 2014; Laube et al., 2017, 2015; Pakarinen et al., 2005)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Capillary force model for identical particles: $F_{\text{cap}} = -\pi\gamma R\left(\cos(\theta+\beta)+\cos(\theta)\right)\left(1-\frac{D}{\sqrt{\frac{V}{0.5\pi R}+D^2}}\right)$	Classical capillary force approximation for two identical particles	R - particle radius θ - contact angle, β - position of the three-phase contact line, γ - surface tension, V - volume of the meniscus (assumed to be constant) D - interparticle distance	Simulation of monodisperse nanoparticle agglomerates in the gas phase	(Butt & Kappl, 2009; Salameh et al., 2017a)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Solvation force model for identical particles: $F_{\text{sol}} = 0.5RF^0_{solv}\frac{1}{\sqrt{\frac{1}{(2\pi\sigma_{solv})^2}+\frac{1}{h^2}}}e^{-D/\sigma_{solv}}\cos\left(\frac{2\pi D}{h}+\Phi\right)$	Solvation force adapted for two spherical particles	σ_{solv} - decay length h - layer thickness Both these values are assumed to be equal to the molecule size. Φ - phase shift F^0_{solv} - amplitude of the first maximum (fitting parameter, can also be obtained from MD simulations)	Simulation of monodisperse nanoparticle agglomerates in the gas phase	(Horn & Israelachvili, 1981; Laube et al., 2018b)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Electrostatic forces: $F_e = -\frac{1}{4\pi\epsilon_0\epsilon_r}\frac{q_iq_j}{d^2}n_{ij}$	Coulomb's law for modelling electrostatic forces between primary particles	$\mathbf{n}_{ij}$ - The normal vector between particles *i* and *j* ϵ_0 - permittivity of free space ϵ_r - relative permittivity of the medium q_i,q_j $-$ the charges on the two particles d - Interparticle distance	Applications involving electrostatically charged primary particles such as dry powder inhalers	(Yang et al., 2015b)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Adhesive model in fluidised beds (4 stages): $F_{na} = -F_a \cdot \mathbf{n}_{\mathrm{ij}} - \eta_n \mathbf{v_{ij}}$ *Approach stage with zero cohesive force:* $F_a = 0 \quad d > H_0$ *Loading stage:* $F_a = k_{n,\mathrm{load}} \delta_n - F_{\mathrm{vdw},0}$ *Unloading stage:* $F_a = \begin{cases} k_{n,\ unload} \left(\delta_n - \delta_p\right) - F_{vdw,\ 0}, \quad F_{Fa} \ \geq \ f_{ad,\ min} \\ \left(1 - x_1\right)\left(F_{vdw.\ 0} - F_{bond,\ 0}\right) + x_1 f_{ad,\ \ min}, F_a < f_{ad,\ min} \end{cases}$ *Detachment stage:* $F_a = \left(1 - \frac{d}{d_c}\right)\left(-F_{\mathrm{vdw},0} - F_{\mathrm{bond},0}\right) +$ $\frac{5}{9}\left(\frac{d}{d_c}\right)\left(-F_{vdw,0} - F_{bond,0}\right)$ *Parameter equations:* $\psi_p = 1 - k_{n,load}/k_{n,\ \mathrm{unload}}$ $\delta_p = \left(1 - k_{n,\ \mathrm{load}}/k_{n,\ \mathrm{unload}}\right)\delta_{\max} f_{ad,\min} = \left(\delta_0/D_{i,NP} + 1\right)\left(-F_{\mathrm{vdw},0} - F_{\mathrm{bond},0}\right)$	The contact model is a modification of Luding's model and can be parameterised directly using Atomic Force Microscope (AFM)	$\mathbf{n_{ij}}$ – The normal vector between particles i and *j* $\mathbf{v_{ij}}$ – velocity vector between particles *i* and *j* H_0 - minimal separation distance d – particle separation distance d_c – interaction breakoff distance $F_{\mathrm{vdw},0}$ – van der Waals force at the minimal separation distance $F_{bond,\,0}$ – bond force at minimum separation distance δ_n - normal deformation δ_p- permanent plastic deformation $k_{n,\mathrm{load}}$- loading stiffness	Simulation of TiO2 nanoparticle agglomerate fluidization	(Liu et al., 2016b)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
van der Waals attractive force approximation *For* $(d \geq z)$*:* $F_{na,\ vdW} = -\frac{A_H \cdot \frac{R_i}{2}}{12 \cdot d^2}$ *For* $(d < z)$*:* $F_{na,\ vdW} = -\frac{A_H \cdot \frac{R_i}{2}}{12 \cdot z^2}$	A model for the approximation of van der Waals attractive forces in dry aggregate particles	d – particle-particle contact distance z – fixed distance parameter (¬0.4nm A_H- Hamaker constant R_i – primary particle size	Simulation of Silica aggregates with sizes of 50-400 nm built up from primary particle. The van der Waals force can be either included as a separate term (R. Y. Yang, Yu, Choi, Coates, & Chan, 2008)	(Schilde et al., 2012, 2014)

Table A.2.: Long range particle-particle attractive forces

Model	Description	Parameters	Applications	Citation
Modified van der Waals attractive force approximation: $F_{na} = -\frac{\pi^2}{3}\frac{\left(nd_0^3\right)^2\varepsilon(D_i)^6}{(r+\alpha d_0)^3\left[(r+\alpha d_0)^2-{D_i}^2\right]^2}$	A model for the approximation of van der Waals attractive forces in dry aggregate particles	d_0 – size of intermolecular interaction ε - energy intermolecular interaction α - coefficient that determines the maximum adhesive bond force = 0.24 in applications d_0 - equilibrium distance between two atoms d - particle distance r - distance between the particles' centres D_i – diameter of particle *i*	Simulation of nanoparticle powder of sizes 10–100 nm in cold compaction. Simulation of 10nm particles in magnetic pulse compaction processes.	*Model:* (Boltachev & Volkov, 2010, 2011; Boltachev, Volkov, & Nagayev, 2011) *Applications:* (Boltachev et al., 2011, 2013, 2014, 2015)

Table A.3.: Friction and tangential models

Model	Description	Parameters	Applications	Citation
Hertz-Mindlin: $F_t\left(\delta_t\right) = -S_t\delta_t$ $S_t = 8G^{*\sqrt{R^*\delta_n}}$	Hertz-Mindlin tangential force model	δ_n - tangential overlap and S_t - tangential stiffness G^* - equivalent shear modulus		

Table A.3.: Friction and tangential models

Model	Description	Parameters	Applications	Citation
Sliding tangential friction force: $F_t = -\tau_S A_{ij} \left(\frac{\delta_t}{\lambda_t} - \left\lfloor \frac{\delta_t}{\lambda_t} \right\rfloor \right) t_{ij}$ $A_{lj} = \pi \left[R_i^2 - \frac{\left(R_i^2 - R_j^2 + d^2 \right)^2}{4d^2} \right]$ *Rolling friction torque:* $M_r = 0.5 \left(R + h_n \right)^* \tau_R A_{ij}$	Friction: Stick-slip behaviour model for the friction behaviour of surfaces with roughness in the atomic scale (of the order of 0.1 nm) Tangential: Reduced order slider model	Friction: τ_S - Sliding shear stiffness τ_R - Rolling shear stiffness A_{ij} - Contact area between overlapping particles t_{ij} -Tangential direction between particles i and j δ_t - the relative displacement of two particles along t_{ij} λ_t -period of displacement d - particle distance Tangential: τ_R - Rolling shear stiffness A_{ij} - Contact area between	Simulation of TiO_2 nanoparticle films subjected to mechanical stress.	Friction: (Laube et al., 2018a) Tangential: (Ai et al., 2011; Laube et al., 2018b)

Table A.3.: Friction and tangential models

Model	Description	Parameters	Applications	Citation
Modified Luding‘s model $F_t = \begin{cases} -k_t\delta_t - \eta_t v_{\text{ab,t}}, F_{ij,t} < \mu_f \left\|F_{ij,n}\right\| \\ -\mu_f \left\|F_{ij,n}\right\| \mathbf{n}_{ij}, F_{ij,t} \geq \mu_f \left\|F_{ij,n}\right\| \end{cases}$ *where the damping coeffient is given by* $\eta_{n(t)} = 2\sqrt{m_{ab}k_{n(t)}} \ln e_{vis} / \sqrt{\pi^2 + \ln^2 e_{\text{vis}}}$	Applicable to agglomerates. The contact model is a modification of Luding’s model and can be parameterised directly using Atomic Force Microscope (AFM)	$\mathbf{v}_{\text{ij, t}}$ - relative velocity between particle i and j $\mathbf{n}_{ij}$ – unit vector be between particle i and j η_t- damping coefficient along the tangential direction μ_f - friction coefficient e_v - coefficient of restitution due to the visco-elastic nature	Simulation of TiO_2 nanoparticle agglomerate fluidization	(Liu et al., 2016b)

Table A.3.: Friction and tangential models

Model	Description	Parameters	Applications	Citation
Linearized form of Cattaneo-Mindlin law: $F_t = \min\left\{\frac{4Ea\delta_t}{(2-v)(1+v)}; \mu F_{nr}; \pi a^2\sigma_b\right\}, \quad a = \frac{\sqrt{hD_i}}{2}$ *Moment balance using Reissner-Sagoci law:* $M_p\left(\theta_p\right) = \min\left\{\frac{8Ea^3\theta_p}{3(1+v)}; \mu M(a); \frac{\pi a^3}{2}\sigma_b\right\}$ *where* $M(a) = -2\pi \int \sigma_n(r)r^2dr$ *Contact elasticity of flexure:* $M_r\left(\theta_r\right) = \min\left\{\frac{4}{3}\frac{Ea^3}{1-v^2}\theta_r; \frac{1}{3}aF_{nr}\right\}$	Modified Cattaneo-Mindlin law using Reissner-Sagoci for the moment balance. Here F_{nr} is the “Rod model” governing repulsion when a particle i is not sintered to a particle j. M_r - rolling moment θ_r - turn angle of the contact area in the direction	D_i – Diameter of particle *i* E - Young modulus of the particles v - Poisson’s Ratio - friction constant σ_b - critical transverse stress (or the fracture stress), M_p - surface moment, θ_p - turn angle of the contact area	Simulation of nanoparticle powder of sizes 10–100 nm in cold compaction. Simulation of 10nm particles in magnetic pulse compaction processes.	*Model: Linearized form of Cattaneo-Mindlin law* (Agnolin & Roux, 2007; Yang et al., 2000) *Reissner-Sagoci law* (Reisner & Sagoci, 1944) *Applied model* (Boltachev et al., 2013) *Applications:* (Boltachev et al., 2011, 2013, 2015)

Table A.4.: Sinter-bridge models

Model	Description	Parameters	Applications	Citation

Table A.4.: Sinter-bridge models

Model	Description	Parameters	Applications	Citation
Hertz-Mindlin with Bonding contact model. *Normal and tangential force on particle:* $F_s = L^n \frac{E_B}{L_0} A_B$ $F_{S,t} = L^t \frac{E_B}{2L_0(1+\upsilon_B)} A_B$ *Moment on particle:* $M_n = \delta^n \frac{E_B}{2L_0(1+\upsilon_B)} J$ $M_t = \delta^t \frac{E_B}{L_0} I$ *Brittle fracture failure models:* $-\frac{F_\mathrm{n}}{A_\mathrm{B}} + \frac{\lvert M_\mathrm{t} \rvert}{I} R_\mathrm{B} \geq \sigma_\mathrm{B}$ $\frac{\lvert F_\mathrm{S,\,t} \rvert}{A_\mathrm{B}} + \frac{\lvert M_\mathrm{n} \rvert}{J} R_\mathrm{B} \geq \tau_\mathrm{B}$	The model of Potyondy and Cundall (2004) and adapted for sinter-bridges in aggregates (Amin, Li, Wu, Ding, & Xu, 2010).	L^n - normal displacement L^t - tangential displacement L_0 - rest length E_B - sinter bridge Young's Modulus A_B- bridge cross section, computed from bridge radius $A_B = \pi\, R_B^2$ υ_B - Poisson's ratio. I - moment of inertia J - polar moment of inertia $\delta^n -$ relative tangential angular displacements via moment of inertia I δ^t - relative tangential angular displacements via polar moment of inertia J. Brittle fracture failure models	Simulation of TiO_2 nanoparticle films subjected to mechanical stress. Simulation of aggregate breakups using the failure models was applied in multiscale simulation of the wet milling process with aggregates built up from primary particle sizes of 12 nm and 13 nm alumina nanoparticles.	*Model:* (Amin et al., 2010; Laube et al., 2018b) *Applications:* (Baric et al., 2019; Beinert, et al., 2018a, 2018b; Ostanin et al., 2013)

Table A.4.: Sinter-bridge models

Model	Description	Parameters	Applications	Citation
Rod model: $\frac{F_{nr}}{E{D_i}^2} = \frac{(h/D_i)^{3/2}}{3(1-v^2)}$ $- \frac{(\pi/4)(1-v)}{(1-2v)(1+v)} \left[\frac{h}{D_i} + \ln\left(1 - \frac{h}{D_i}\right)\right]$ *with* $h = d - r$	Modified Hertz law (so-called "rod model") used for repulsion between the particles up to that moment when a solid bridge appears	D_i – diameter of particle *i* E - Young modulus of the particles v - Poisson's ratio	Simulation of nanoparticle powder of sizes 10-100 nm in cold compaction. Simulation of 10nm particles in magnetic pulse compaction processes.	*Model:* (Boltachev & Volkov, 2010; Boltachev & Volkov, 2011; Boltachev, Volkov, & Nagayev, 2011) *Applications:* (Boltachev et al., 2011, 2013, 2014, 2015)

CFD-DEM models

B

B.1 Mass and momentum conservation equations

The mass and momentum conservation equations are given by

$$\frac{\partial \alpha_f \rho_f}{\partial t} + \nabla \cdot (\alpha_f \rho_f \mathbf{u_f}) = 0$$

$$\frac{\partial (\alpha_f \rho_f \mathbf{u_f})}{\partial t} + \nabla \cdot (\alpha_f \rho_f \mathbf{u_f} \mathbf{u_f}) = -\alpha_f \nabla p - \mathrm{K_{sf}} (\mathbf{u_f} - \mathbf{u_s}) + \nabla \cdot (\alpha_f \tau) + \alpha_f \rho_f \mathbf{g} + \mathbf{f} \tag{B.1}$$

$$\frac{\partial}{\partial t} (\varepsilon_g \rho_g) + \nabla \, . \, (\varepsilon_g \rho_g u_g \,) = \, 0 \tag{B.2}$$

$$\frac{\partial}{\partial t} (\varepsilon_g \rho_g u_g) + \nabla \, . \, (\varepsilon_g \rho_g u_g u_g \,) = \, -\varepsilon_g \nabla \rho_g - \nabla \, . \, (\varepsilon_g \tau_g) + S_p + \, \varepsilon_g \rho_g \, \mathbf{g} \tag{B.3}$$

Drag force model for modelling the fluidisation of aggregate nanoparticles

The following model can be used as the fluid drag component acting on the particles (Liu et al., 2016b):

$$F_d = \frac{V_a f_d}{(1 - \varepsilon_g)} (u_g - v_p) \tag{B.4}$$

The drag force coefficient is given by

$$f_d = \frac{3}{4} C_D \frac{\rho \, |u_g - \overline{v_p}| \, (1 - \varepsilon_g)}{d_p} {\varepsilon_g}^{-2.7} \tag{B.5}$$

The parameters for this model are described in Table 8. The drag force is an important component in fine grinding and dispersing processes such as wet milling, which can change the morphology of the aggregate structures. Morphology can be measured quantitatively by the fractal (or Hausdorff) dimension D_f, which describes how the number of primary particles scales with the characteristic radius of the aggregate R_g through the power law expression $N_p \sim {R_g}^{D_f}$. The fractal dimension ranges from $D_f \rightarrow 3$ for compact, nearly spherical agglomerates to $D_f \rightarrow 1$ for chain-like structures (Friedlander, 2000).

It is well known that the aggregate morphology can change during a wet milling processes (Jeon, Thajudeen, & Hogan, 2015). Importantly, this change influences the drag force acting on the particle. In two studies on multi-scale simulation of fine grinding and dispersing processes (Beinert et al., 2018a, 2018b) the authors also related the drag coefficient to the fractal dimension. The aggregates have a final fractal dimension $D_{f,\,end}$ which can change due to deagglomeration from shear forces during the simulation and where a value of $D_{f,end} = 3$ corresponds to a spherical approximation of the agglomerate structure:

$$C_{D,simulated} = \frac{2 \cdot F_d}{\rho_f \cdot v_f^2 \cdot A_{agg}} \tag{B.6}$$

$$C_{D,analytical(Stokes)} = \frac{24}{Re} \tag{B.7}$$

At lower fractal dimensions the following relations were calculated in the study:

$$C_{D,simulated}\left(D_{f,end} = 1.55\right) = \frac{544}{Re^{0.34}}$$

$$C_{D,simulated}\left(D_{f,end} = 2.15\right) = \frac{491}{Re^{0.32}}$$

$$C_{D,simulated}\left(D_{f,end} = 2.75\right) = \frac{238}{Re^{0.48}}$$

Table 8

CFD coupling and drag force coefficients

Model	Description	Parameters	Applications	Citation
Mass and momentum conservation equations of fluid phase $\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g\right) + \nabla\ .\ \left(\varepsilon_g \rho_g u_g\ \right) =\ 0$ $\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g u_g\right) + \nabla\ .\ \left(\varepsilon_g \rho_g u_g u_g\ \right) =$ $-\varepsilon_g \nabla \rho_g - \nabla\ .\ \left(\varepsilon_g \tau_g\right) + S_p +\ \varepsilon_g \rho_g\ \mathbf{g}$ *where* $\varepsilon_g = 1 -\ \sum_{n=1}^{N_p} \frac{v_{pn}}{v_{cell}}$ $S_p =$ $-\frac{1}{V_{cell}} -\ \sum_{n=1}^{N_p} \frac{v_\alpha C_{WY}}{(1\ -\varepsilon_g)}\left(u_g - v_p\right)\ .\ \xi$ *Wen & Yu drag coefficient* $C_{WY} = \frac{3}{4} C_D \frac{\rho\lvert\mathbf{u_g} - \bar{\mathbf{v}}_p\rvert(1-\varepsilon_g)}{d_p} \varepsilon_g - 2.7$	CFD-DEM coupling	ε_g - bed voidage ρ_g- particle or agglomerate density u_g - fluidizing gas velocity ξ – drag force scale factor β_c – drag coefficient	Modelling processes such as fluidised bed reactor production of nanoparticle powders. Modelling of deagglomeration in dry powder inhaler systems.	*CFD-DEM model* (Liu et al., 2016) *Wen & Yu drag coefficient* (van Wachem et al., 2001)

Model	Description	Parameters	Applications	Citation
Felice friction coefficient $f_{\mathrm{d}} = \frac{1}{2} C_{\mathrm{Di}} \rho_{\mathrm{a}} \frac{\pi D_{\mathrm{i}}^2}{4} \varepsilon^2 \left\| u - v_i \right\| (u - v_i) \varepsilon^{-(\chi+1)}$		C_{Di} - the fluid drag coefficient the fluid drag coefficient D_i - diameter of the particle i ρ_a – density of the surrounding fluid u - velocity of the surrounding fluid	Applied to dry powder inhaler systems.	Model: (Di Felice, 1994) Applications: (Pei et al., 2015; Yang et al., 2015c)

B.2 CFD-DEM simulation for macroscopic applications

Two selected popular software framewords are CFDEM and MultiFLow:

CFDEM. The CFDEM coupling of LIGGGHTS and OpenFOAM offers a parallel resolved software (Goniva, et al., 2012; Hager, et al., 2014). It has been used to study multiscale models of fine grinding and dispersing processes in (Beinert, et al., 2018).

MultiFlow. The adhesive CFD-DEM model was implemented by Liu et al. (2016) in MultiFlow (van Wachem & Denner, 2012) to simulate the fluidization of nanoparticle agglomerates. The software uses standard CFD-DEM coupling techniques (Liu, Bu, & Chen, 2013; Mallouppas & van Wachem, 2013; Motlagh, et al., 2014; van Wachem, et al., 2001; Zhu, Zhou, Yang, & Yu, 2007) with the Wen and Yu drag force model (Wen & Yu, 1966).

B.3 Molecular dynamics

The use of MD software has primarily found applications in parameter estimation and model development, although the numerical simulation techniques for coarse-grained particles are largely similar to atomistic and molecular simulations.

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator). The popular LAMMPS software (Plimpton, 1997) was used by Laube et al. (2018) to compute e.g. the damping coefficient γ_n and other parameters of their DEM model. In the study several MD runs were performed starting with the particles at a distance larger than their equilibrium separation. In the simulation the water layers were first relaxed to form a stable meniscus and then letting the system spontaneously relax to equilibrium at constant temperature.

The development of models and parameters for capillary forces in TiO_2 nanoparticles was also conducted using the LAMMPS software (Laube et al., 2017, 2015)

B.4 Applications using state of the art CFD(-DEM) methods

In this section some practical applications of the nanoparticle contact models is summarised especially with respect to the kind of valuable insights that can be gained from such simulations. The early examples are taken verbatim from Endres, Ciacchi, and Mädler, 2021 while the later examples update some newer applications in the field since the publication.

Practical applications of particle-particle contact models

The models described in Section 2 have already led to insight in a number of practical applications as well as deepened our knowledge of these materials and the processes used to manufacture them.

Film compaction processes. In the studies by **Baric2019**; Laube et al. (2018) the DEM model was used to study structural changes during compression of TiO_2 nanoparticle films synthesised by flame-spray pyrolysis (Schopf, Salameh, and Mädler, 2013b). The model demonstrated the ability of the sintered TiO_2 aggregates to rearrange via mutual detachment, rolling or sliding events dictated by non-covalent, humidity-dependent interactions, which are known to be crucial to predict the correct response to compaction. The simulations allow for the dynamic calculation of sinter bridge breakage rates and aggregate size distribution, which could be validated with experimental results. It was found that the elastic deformation of aggregates was crucial to prevent fragmentation of stable sinter bridges at this size scale of the aggregates even during harsh film handling such as compaction at pressures of several MPa. The model allows for key insights in predicting the change in functional film properties such percolation, pore structure, heat transfer and electric conductivity.

Additive manufacturing. Yoshida et al. (2016) used the DEM with van der Waals-Hamaker model to predict particle flowability improvements when nanoparticles (20nm) are added to the main particles (400nm) during compressed packing. An analysis of the effects of adhesive forces and changing degree of rotations of admixed particles found that rotation of the admixed particles was not the main reason for the improvement in flowability. Rather, this peak tendency was dictated by the Hamaker constant of the admixed particles.

Fluidised beds. Bahramian & Grace (2017) studied the mechanical properties and interparticle cohesive forces of the TiO_2 nanoparticles in different fluidization regimes. Understanding the agglomeration process of nanoparticles was crucial to help decrease their adhesion during fluidization. The Young's modulus and hardness of the agglomerates were determined by nanoindentation to infer the morphology of aggregates on the fluidization process (van Ommen et al., 2012).

In (Liu et al., 2016b) the authors developed a CFD-DEM model for the simulation and characterisation of TiO_2 nanoparticle agglomerates during fluidization. The same model led to valuable insights into the fluidization process (Liu et al., 2016a):

- As the nanoparticle agglomerates move around in the bed, they break and recombine repeatedly. Gas channelling is observed in case of strong agglomeration.
- A strong correlation is observed between the dispersion coefficient and the agglomerate breakage rate.
- The evolution of agglomerates could be visualised, and the agglomerate breakage is compared for different cases.
    - Increasing fluidizing gas velocity or reducing particle adhesion can increase agglomerate breakage rate.
    - There are different contacts in an agglomerate, i.e. weak contacts, strong contacts, and sometimes permanent contacts, which result in different stabilities of agglomerates.

Modelling the mechanical properties of Silica aggregates. In (Schilde et al., 2014) DEM simulations were performed with the previously determined micromechanical properties of aggregates or agglomerates, which are the main products of large-scale production of nanostructured materials. The study used the standard model of Mindlin with the previously developed contact model (Schilde & Kwade, 2012) and an attractive van der Waals interaction force. Insight from the studies determined that calculated radial forces within the aggregates were significant for the crack formation and propagation. The data could be compared to the theoretical model of Schönert (2004).

Modelling agglomeration and deagglomeration in dry powder inhalers. In dry powder inhaler systems, agglomeration and deagglomeration are important in two stages. Namely, (i) during the inhalation process drug particles are aggregated into agglomerates or mixed with large carrier particles; next (ii) the agglomerates disintegrate into small fractions or the drug particles detach from the carrier particle for the delivery of drugs to the lungs. The cohesive and adhesive forces need to be sufficiently strong to aggregate and transport the particles, but weak enough to be overcome by the removal forces generated during the de-agglomeration process (de Boer, Chan, & Price, 2012; Stegemann et al., 2013; Yang et al., 2015a; Zhou & Morton, 2012). According to (Kaialy et al., 2014) it is rather difficult to identify the influence of a single factor on dry powder inhaler performance due to the complexity of dry powder inhaler system. This complexity is similar to that found to influence the forces of aerosol synthesised materials as discussed in Section 2.1.2. In dry powder inhaler systems it is known from several studies (Yang et al., 2015a) that simulations should include effects of particle size (Guenette et al., 2009), particle shape (Kaialy et al., 2011), particle surface roughness (Kaialy, Ticehurst, & Nokhodchi, 2012), particle concentration (Young et al., 2005), material properties (Steckel & Bolzen, 2004), storage conditions (Jashnani, Byron, & Dalby, 1995), surface energy (Cline & Dalby, 2002), density and porosity (Kaialy & Nokhodchi, 2012), crystal form (Shur et al., 2012), flowability (Rabbani & Seville, 2005), the type of ternary additives (Kaialy & Nokhodchi, 2013).

The inclusion of electrification phenomena (Pei et al., 2013) in CFD-DEM simulations was demonstrated to have considerable effects on the dry powder inhaler performance. In (Yang et al., 2013, 2014) it was assumed that van der Waals forces dominated the agglomeration process in mixing. The adhesion between particles was considered using the JKR theory (Johnson et al., 1971). In a later study (Yang et al., 2015b) the effect of electrostatic forces on the mixing performance was studied. In contrast to the earlier proposed mechanisms for the mixing process, the latter study indicated that both long-range and short-range adhesive forces resulted in different mixing behaviours.

Modelling the mechanical properties of aggregates in fine grinding and dispersing processes. In two studies (Beinert, et al., 2018a, 2018b) of aggregates built up from primary particle sizes of 12 nm and 13 nm, alumina nanoparticles were simulated in the wet mill process. In this study the breakage rate in a macroscopic PBE was computed by directly simulating a representative volume element to determine the number of fragments and broken solid bonds given by the micro scale CFD-DEM simulation. The study allowed the authors to predict the drag coefficient and the number of broken aggregate bonds that occurred due to stress between the grinding beads. The authors also concluded that the fractal dimension is insufficient to describe both the effect of the aggregate structure on the drag coefficient and the fracture during mechanical compression between grinding media. These conclusions emphasise the need for the use of a more fundamental CFD-DEM model in this process.

Modelling the mechanical properties of aggregates in compaction processes. In Boltachev et al. (2013) verification of the model was performed on the compaction of alumina-based nanopowders and 1% Nd:Y_2O_3 nanopowders (yttrium oxide doped by 1% of neodymium). One of the key findings from the study is that the quantitative analysis of easily aggregating nanopowders, in this case particles with an adsorbates-free surface, require stronger interparticle bonds of chemical nature. In Boltachev et al. (2014) the authors performed a purely theoretical study of magnetic pulse compaction processes with particles smaller than 10nm. It was found that high pressures are reached that exceed the initial "magnetic pressure" several times due to the inertial effects under the radial pressing of conductive shells. A new dimensionless number was derived from the process dynamics which can be used to optimise the design within the most effective value range established in the study.

Particle Separation in a Cyclone. Misiulia, Lidén, and Antonyuk (2021) developed a simulation of the vortex behaviour and pressure losses in small-scale cyclones with turbulent conditions simulated using the LES method. A follow up study in Misiulia, Nedumaran, and Antonyuk (2023) simulated a cyclone system which led to insights that could improve flapper design.

C

Other DDG formulations from literature

Discrete curvature estimates of a range of definitions taken from literature

A collection of curvatures known from literature is shown in Table C.1. The results of plotting the minimum symmetric discretisation on a spherical cap is shown in Figure C.1. The vector and scalar curvatures are plotted over a range of contact angles Θ_C and compared to the exact Gaussian K and exact Mean H curvatures. As illustrated in Table C.1, for all curvature estimates, the errors for the discrete Gaussian K and the discrete Mean H curvatures are expected to be low close to $\Theta_C = 90^\circ$, where there is no curvature at all, and the errors are expected to be high as Θ_C approaches 0° and 180°. To enhance the symmetry, the sign of the concave curvatures has been reversed. It can be observed in Table C.1 that, similar to the droplet test case, the dual normalised mean normal curvature once again provides the best approximation for the normal curvature, being nearly precisely equal to H for the entire range of Θ_C.

Vector curvatures:	
Gaussian normal curvature:	$K\boldsymbol{N}dA = \frac{1}{2}\sum_{ij \in S\mathrm{St}(i)} \frac{\varphi_{ij}}{\ell_{ij}}\left(\boldsymbol{f_j} - \boldsymbol{f_i}\right)$
Mean normal curvature:	$H\boldsymbol{N}dA = \frac{1}{2}\sum_{ij \in St(i)} \left(\cot\alpha_{ij} + \cot\beta_{ij}\right)\left(\boldsymbol{f_i} - \boldsymbol{f_j}\right)$
Surface area gradient:	$\boldsymbol{N}dA = \frac{1}{6}\sum_{ijk \in \mathrm{St}(i)} \boldsymbol{f_j} \times \boldsymbol{f_k}$
Scalar curvatures:	
Gaussian curvature (angle defect):	$\Omega_i = 2\pi - \sum_{ijk}\theta_i^{jk}$
Mean curvature:	$H_{ij} = \frac{1}{2}\ell_{ij}\varphi_{ij}$

Table C.1.: Definitions of discrete curvature (from Crane (2018)).

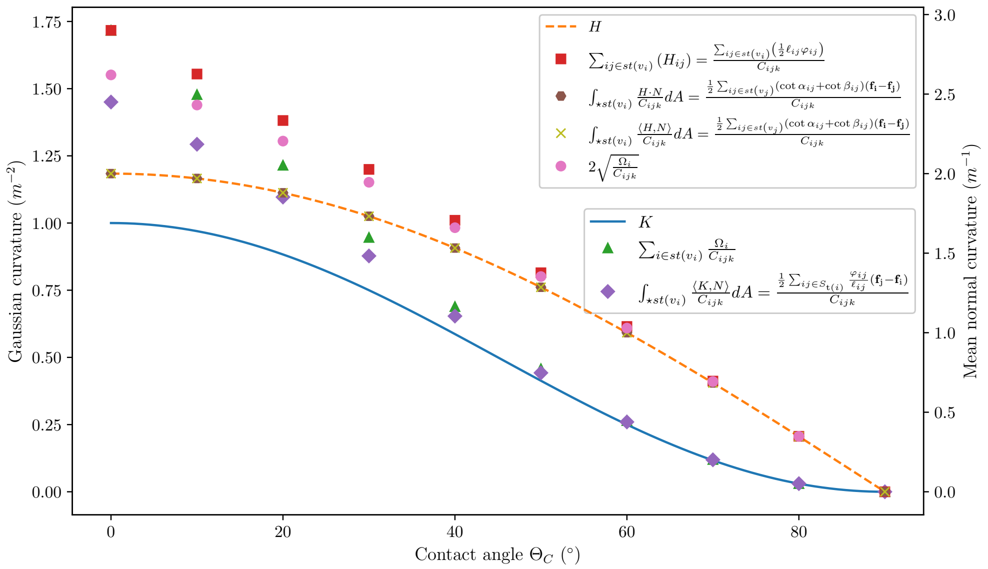

Figure C.1.: Varying contact angle Θ_c at $a = 1.0, n = 5$. Note that the approximations for normalised mean normal curvature are precisely on the smooth value H. Note that none of the notions of discrete Gaussian curvature are close to smooth value at high curvatures (or low contact angles), however, it is possible to get good point-wise approximations for the Gaussian curvature using the implicitly defined notions shown in Chapter 4.

D Interface energy minimisation definitions and demonstration

The method described in this appendix is taken almost verbatim from Endres, Avila, and Mädler (2022). It is demonstrated how Equation 3.5 can be used to directly minimise the energy of multiphase systems under certain assumptions. In such a model it is assumed that viscous inertial forces are negligible and it is further assumed that there is no exchange of heat mass between phases (i.e. thermodynamic equilibrium between phases). This allows us to decouple the effects from numerical methods in the discretisation of the momentum balance (Navier-Stokes and other continuum equations) with the formulation presented here, although it should be noted that assumptions should not be considered an accurate dynamic simulation due to the absence of the momentum balance. A mixture thermodynamic definition of surface tension (Elliott, 2020) of an arbitrary phase j is defined as:

$$\gamma^j = \left(\frac{\partial U^j}{\partial A^j}\right)_{S^j, N_i^j}, \tag{D.1}$$

when we integrate this on an isotrope at constant composition we find

$$U^j = \int \gamma^j dA \tag{D.2}$$

which relates the (intrinsic) internal energy of a system and the change in (local) area of the interface, lowering the local surface area lowers in the intrinsic internal energy of the system. This also has a direct interpretation on the driving force $F = -\nabla U$ (i.e. the force due to surface tension is equal to the gradient of the internal energy surface) or in differential forms $F = -dU^j = -\gamma^j dA$. The Young-Laplace Equation 2.1 is indirectly connected to the idea that surface tension tends to minimise energy of the system by minimising the local surface area of the gas-liquid interface under constraints leading to natural phenomena such as droplet formation. This fundamental idea of minimising local surface area is called "mean curvature flow", or sometimes called "H2 flow" in the context of discrete differential geometry and is about computing gradients of the surface energy by the discretising the differential 2-form dA. From this we can find the force acting on a vertex point to lower the energy of a simplicial complex by pushing the vertices down the gradient. Therefore, the term "equilibrium" is used, however,isentropic dynamics can be derived from the discretisation that is a precise physical relationship only under those conditions. We briefly review the concept of mean curvature flow in the context of discrete differential geometry. Mean curvature energy in the context of discrete exterior calculus is defined by (Crane, 2018):

$$E(f) = \int_M dA \tag{D.3}$$

Where the right hand side of Equation 2.10 can be related to Equation D.3 only under the assumption of constant surface tension between phases γ. In differential form this can be described as

$$\delta E = 2HNdA \tag{D.4}$$

And the change of vertex positions over some period τ is

$$\frac{d}{d\tau} f = -2HN \tag{D.5}$$

We can additionally specify a force vector $\boldsymbol{F}$ a point on the surface to balance the physical mechanical equilibrium:

$$\frac{d}{d\tau} f = -2HN + \boldsymbol{F} \tag{D.6}$$

The discretised differential at a vertex $\boldsymbol{f_i^k}$ is given by the explicit update formula

$$\boldsymbol{f_i^{k+1}} = \boldsymbol{f_i^k} - \frac{\tau}{2} \sum_{ij \in S_{\mathrm{I}}(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \left(\boldsymbol{f_j^k} - \boldsymbol{f_i^k} \right) + \boldsymbol{F_i} \tag{D.7}$$

Similarly, we update boundary vertices according to Equation 3.12. Where the τ parameter becomes a specified constant. A more stable formulation of the energy minimisation of any surface can be described in terms of its Willmore or H2 flow. The discrete Willmore energy is defined by:

$$E_{\text{discrete}} = \sum_{i \in V} (HN)_i^2 / A_i \tag{D.8}$$

In terms of the inner product of the Laplacian:

$$\begin{aligned} \nabla E_W(f) &= \frac{1}{4} \langle\langle \Delta f, \Delta f \rangle\rangle = \frac{1}{4} \left\langle \left\langle \Delta^2 f, f \right\rangle \right\rangle \\ \dot{f} &= -\nabla E_W(f) = -\frac{1}{2} \Delta^2 f \end{aligned} \tag{D.9}$$

The period is defined by:

$$\frac{d}{d\tau} = -\nabla_f E \tag{D.10}$$

We solve this equation subject to mass conversation constraints by computing the density from the equation of state and preserving the expected volume at a given droplet pressure. Figure D.1 demonstrates a general algorithmic flow diagram of the computations performed in the case studies in Section 4.

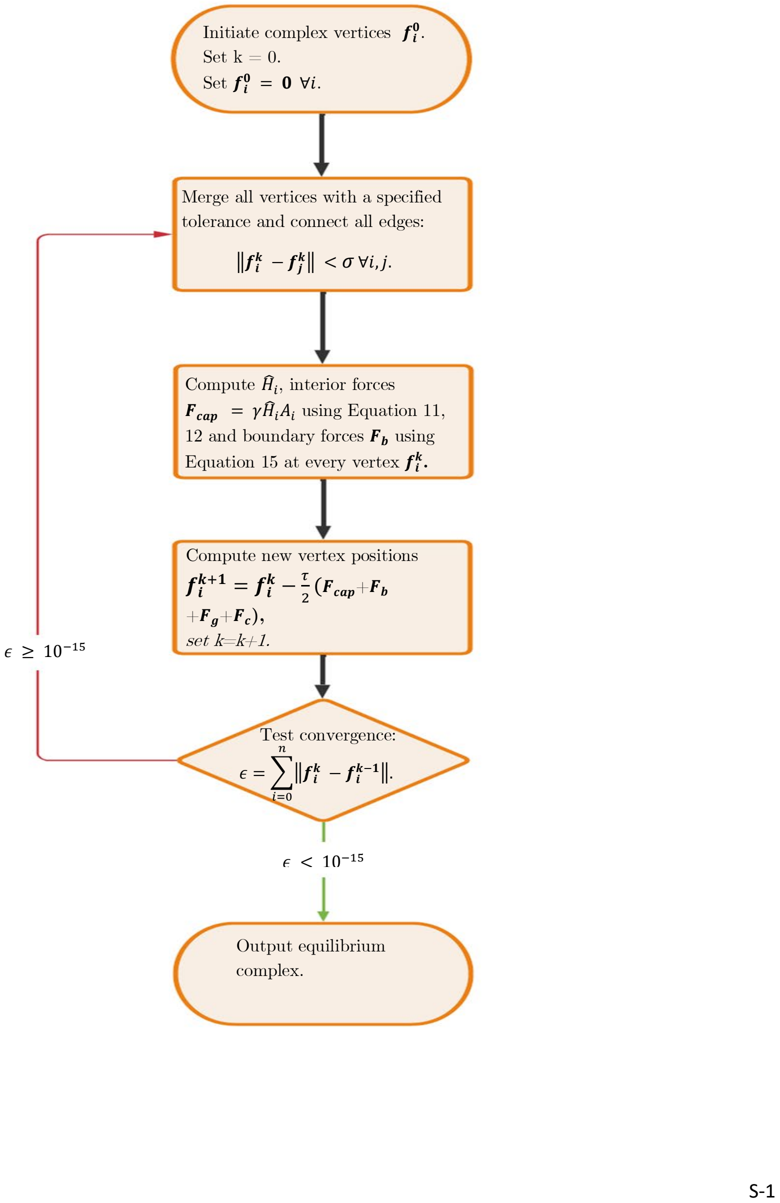
Initiate complex vertices $\boldsymbol{f_i^0}$.
Set k = 0.
Set $\boldsymbol{f_i^0} = \boldsymbol{0}\ \forall i$.
Merge all vertices with a specified tolerance and connect all edges:
$\|\boldsymbol{f_i^k} - \boldsymbol{f_j^k}\| < \sigma\ \forall i, j.$
Compute $\widehat{H}_i$, interior forces $\boldsymbol{F_{cap}} = \gamma \widehat{H}_i A_i$ using Equation 11, 12 and boundary forces $\boldsymbol{F_b}$ using Equation 15 at every vertex $\boldsymbol{f_i^k}$.
Compute new vertex positions
$\boldsymbol{f_i^{k+1}} = \boldsymbol{f_i^k} - \frac{\tau}{2}(\boldsymbol{F_{cap}} + \boldsymbol{F_b}$
$+ \boldsymbol{F_g} + \boldsymbol{F_c}),$
set k=k+1.
$\epsilon \geq 10^{-15}$
Test convergence:
$\epsilon = \sum_{i=0}^{n} \|\boldsymbol{f_i^k} - \boldsymbol{f_i^{k-1}}\|.$
$\epsilon < 10^{-15}$
Output equilibrium complex.
S-1

Figure D.1.: Algorithmic flow diagram of all calculation steps involved in minimizing the energy of an interface. Note that σ must be set to a value appropriate for the simulation domain. $\boldsymbol{F_g} = \rho g h$ is the gravity force and $\boldsymbol{F}_c$ are compressive forces acting on the vertex $\boldsymbol{f_i}$.