Université de Montréal

Mechanical Characterization of Rigid Discrete Interlocking Materials

par

Charles Gingras

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

Orientation Imagerie

August 1, 2021

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Mechanical Characterization of Rigid Discrete Interlocking Materials

présenté par

Charles Gingras

a été évalué par un jury composé des personnes suivantes :

Mikhail Bessmeltsev
(président-rapporteur)

Bernhard Thomaszewski
(directeur de recherche)

Pierre Poulin
(membre du jury)

Résumé

Les matériaux discrets entrecroisés (DIM) rigides sont une classe de matériaux qui se distinguent par la manière unique par laquelle ils se déforment: les DIMs sont composés d'éléments (connectés par entrecroisements) qui peuvent se déplacer librement à l'intérieur d'une amplitude définie par les contacts avec leurs éléments voisins. Ceci donne une réponse biphasique aux déformations unique à ces structures où soit aucune résistance n'est fournie à une déformation, soit un arrêt complet à la déformation se présente. Il n'est pas clair comment l'ensemble de paramètres discrets et continus décrivant un DIM influence ce comportement biphasique. De plus, nous ne possédons pas les outils pour le charactériser correctement. Dans le but d'élucider ce comportement, nous présentons une méthode qui s'inspire de techniques d'homogénisation qui peut détecter les contacts physiques entre éléments composés de tores. En définissant une énergie adéquate, nous pouvons minimiser les intersections entre éléments tout en déformant le DIM d'une façon arbitraire en utilisant des techniques d'optimisation standardes. Nous explorons les déformations auxquelles des arrangements planaires de DIMs peuvent être assujettis et investiguons comment le couplage de contraintes dans deux directions orthogonales influence ces déformations. Nos résultats permettent de mieux comprendre comment différents paramètres décrivant un DIM influence ces déformations.

Mots clés: matériaux discrets entrecroisés, surfaces implicites, optimisation, déformation, arrangements planaires

Abstract

Rigid discrete interlocking materials (DIMs) are a class of materials that distinguish themselves by the unique way in which they deform: in DIMs, elements (connected through interlocking) can move freely within a range defined through contacts with neighbouring elements. This results in a biphasic deformation behaviour unique to these structures where no resistance is provided to deformation or a hard stop to deformation is met. It is yet unclear how the set of discrete and continuous parameters describing a DIM influences this biphasic behaviour. Likewise, we lack tools to properly characterize it. To that effect, we present a method which takes inspiration from homogenization and handles contacts by leveraging the definition of implicit surfaces, specifically tori, making up our elements. By defining an adequate energy function, we can minimize intersection between elements while deforming the DIM in an arbitrary way using standard optimization approaches. We explore the deformations that planar sheets of DIM can be subjected to and investigate how the coupling of constraints in two orthogonal directions affects these deformations. Our results give insights on how the tuning of various parameters describing the DIM affects these deformations.

Keywords: discrete interlocking materials, implicit surfaces, optimization, deformation, planar sheets

Contents

List of figures

List of abbreviations

DIM	Discrete interlocking material
MSM	Mass-spring model
FEM	Finite element method
KPS	Kirchhoff-Plateau surfaces
MRE	Multi-ring element
DOF	Degree of freedom

Remerciements

J'aimerais exprimer ma sincère gratitude envers les personnes m'ayant aidé dans la réalisation de ce mémoire.

D'abord, merci à mon superviseur Bernhard Thomaszewski pour l'opportunité de faire cette recherche. Sa pédagogie ainsi que sa patience ont été grandement appréciées à travers ce projet. Son sens de l'humour m'a aidé à garder un esprit positif quand certains problèmes plus coriaces m'ont occupé pour quelques mois d'affilée.

De plus, un grand merci au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), au Fond de recherche Québec nature et technologies (FRQNT) et aux Fonds de Bourses Banque Nationale pour leur support financier à travers mon parcours à la maîtrise.

Enfin, j'aimerais remercier ma famille et mes proches pour leur support au fil des années.

Chapter 1

Introduction

Discrete Interlocking Materials (DIM) are materials made of individual elements that are interlocked together in a repeating fashion, constraining each other's motion in three dimensional space. DIMs stand out among other materials by their biphasic behaviour: deformation in a given direction will be largely unresisted until a fully extended configuration is reached in that direction. At this point, further deformation is met with the full stiffness of the material making up the elements and restoring forces will be generated to counter the deformation. The transition point between inelastic and elastic regimes will vary based on the direction of deformation, the element dimensions, and the topology of interlocking. Moreover, due to DIMs' discrete and inherently redundant nature, failure of individual elements will generally remain local and not propagate in a catastrophic manner. Chainmail is a prime example of a DIM where the elements are more often than not rigid. Rigid DIMs distinguish themselves from other DIMs by the hard stop in deformation that is met once they are fully extended, i.e., in the elastic regime. These properties of DIMs make them attractive prospects for certain applications: in robotics, for instance, DIMs can be useful for manipulation and picking tasks, enabling end-effectors that provide compliance during approach and strength during transport. For soft robots, the tunable anisotropy can be used to shape pneumatically-actuated limbs that would otherwise expand in an isotropic way. To enable such applications, we must first establish a formal way to characterize DIMs. In this thesis, we are more specifically concerned with understanding where in the configuration space of a DIM its inelastic regime takes place. For the sake of simplicity, we assume contacts are frictionless and focus on rigid DIMs. With that goal in mind, the simulation of such materials presents two main challenges.

The first challenge pertains to the large number of elements present in a DIM: a typical piece of chainmail can easily be composed of hundreds of elements, if not more. Approaches using rigid body mechanics are ill-suited for the task of design-space exploration due to the large degrees of freedom associated with DIMs. Typical optimizations such as coarsening

of the elements' meshes can be adopted, but that will be at the expense of accuracy. The freezing of bodies with low kinetic energy is a common and well researched heuristic to minimize computation times in rigid body simulations [**29**]. More sophisticated versions of it which leverage bodies stacking into messy piles [**15**] or bodies in contact with low relative velocities [**7**] would seem useful in the context of DIMs. However, we are interested in the deformation of such materials and their mechanical properties. These protocols would mostly come into effect at the end of simulations, where these materials have already left the inelastic regime, or near it, thus being of little help in the characterization task.

The other challenge involves the interlocking characteristic of DIMs. It is important that the interlocking relationships between different pairs of elements are preserved throughout the simulation. Failure to enforce such rules would render the simulation of a DIM essentially useless as elements could move through one another, a behaviour quite unlike the physical reality we are attempting to reproduce.

Keeping those challenges in mind, we decide to adopt an energy-based method that leverages the structure of elements composed of implicit surfaces, specifically tori. We introduce a score function that can accurately determine the presence of overlap between two tori allowing the detection of contacts between elements at any time step. We also derive an equation whose evaluation determines whether or not two elements interlock. By defining an energy whose minimization induces deformation in the DIM while avoiding intersection between elements, we can use optimization routines such as Newton's method to explore the deformation space of the DIM while ensuring that topology is preserved at every time step. In particular, we investigate how the coupling of constraints applied in orthogonal directions affects the deformation limit a DIM may be subjected to. Lastly, to study the macromechanical behaviour of infinite tilings of DIMs, we take an approach based on homogenization where the DIM is a tiling of a unit cell and depends entirely on the cell's parameters. Our contributions include:

- An accurate torus-to-torus score function to detect contacts and intersections;
- An accurate way to detect interlocking between a pair of tori;
- A scheme to simulate and analyze deformation in rigid DIMs with little trade-off in accuracy.

Chapter 2

Related work

Work done specifically on rigid DIMs is, as a whole, quite sparse. Despite chainmail existing for thousands of years, the use of DIMs as a mechanical metamaterial has been largely unexplored. Engel and Liu [**12**] describe a fabrication process to create microscopic chainmail for smart textile. However, much of their investigation focused on electrical properties of the material: mechanical properties were ignored except when it came to the material's load bearing ability. Slightly more in line with our objectives, Caraglio et al. [**4**] explore how interlinked chains of polymers behaved under mechanical stretching by using Langevin dynamics simulations. They explore how varying interlocking configurations deform as the chains are pulled apart from one another using constant forces and how the interlocking affects the length of the stretched out system. By applying a constant velocity to the elements instead, they also were able to analyze the force the rings were subjected to as they became more stretched out. In the context of soft robotics, Ransley et al. [**27**] present a concept study of a chain of rigid elements which, when applied with a voltage, contracts under actuation continuously from a curved drape profile to a rigid cantilever.

In addition to macromechanical characterization, via simulation, a more application-oriented but central aspect of metamaterial design is the ability to understand the possibilities, and limitations, of a space of materials. Within visual computing, the exploration of high-dimensional design spaces has been intensively studied, whether it be through physical simulation in the context of clothing [**34**], flexible molds [**39, 20**], and furniture design [**33**] or through physical contraints in the context of surface shapes [**8, 38**].

As mentionned previously, DIMs behave unlike any other material, but they do share traits with some. On a macroscopic scale, sheets of rigid interlocking elements can be roughly thought of as sheets of inelastic cloth, especially when they have left their inelastic regime: neither can be stretched and both show similar behaviour to out-of-plane motion, namely in how they can both drape over surfaces. As a matter of fact, chainmail was once worn as a form of protection and, even nowadays, designers come up with pieces of clothing made

Fig. 2.1. The use of chainmail as garments. Left [**13**] and right [**28**].

up of interlocking elements. For instance, *Nervous System* [**28**] applied their *Kinematics* system to chainmail-like textile structures which allowed them to fold their designs to obtain smaller configuration and thus optimize 3D-printing. Other artists such as Danit Peleg [**24**] are also making a push to popularize the use of 3D-printed garments in general. Since we are interested in planar arrangements of DIMs, planar and sheet materials also bare relevance, especially those that are structured. Metamaterials share similarities to chainmail in that both are made up of individual elements at the microscopic level, usually arranged in some form of a unit cell, which themselves describe the macroscopic deformation the system of elements can exhibit by how they are tiled. Hence, in the next sections, we shall cover some of the work that has been done in relation to cloth, planar materials and assemblies, and metamaterials and interlocking materials.

2.1. Cloth

The simulation of cloth is a well researched problem in the graphics and animation community. The standard approach involves simulating the cloth as a simple sheet of triangles where the edges are simulated as springs. Collisions between edge-edge and face-vertex pairs are predicted during a time step and resolved by applying impulses. The position of elements at each time step is determined by either explicit or implicit time integration [**2**].

However, this approach is not always adequate as it ignores the underlying microscopic structure of the cloth which can affect the mechanical properties of the cloth. Take for instance yarn-based fabric: the manner in which the yarn is knitted has a significant impact on the microscopic and macroscopic behaviours of the fabric, something that a simple triangle mesh sheet cannot accurately replicate. In woven fabrics, the yarns are nearly immobile meaning that the overall yarn structure is subject to very little deformation. Meanwhile, the yarns of knitted fabric can deform and slide much more freely. Changes in the small-scale structure of the fabric thus happen readily leading to a highly extensible cloth at the

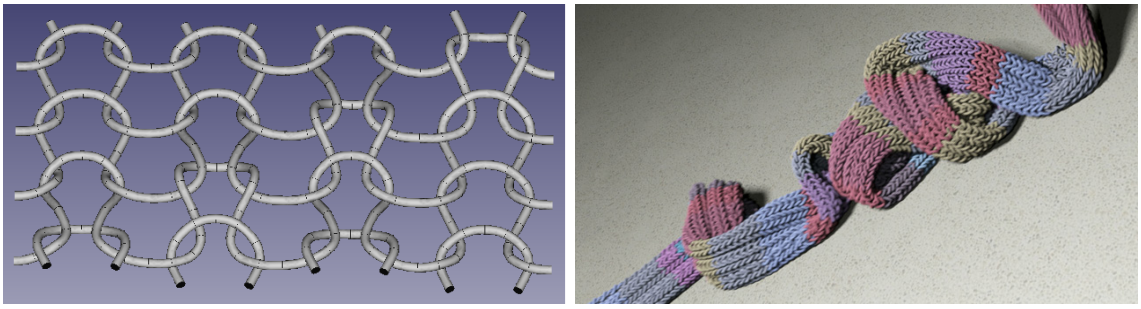


Fig. 2.2. An example yarn structure [**35**] and yarn-level simulation [**16**].

macroscopic scale. In their work, Kaldor et al. [**16**] address this discrepancy between both types of cloth and propose a solution to better simulate knitted fabrics: a model where each yarn is approximated by a B-spline curve. They can then model the yarn's time evolution using the equations of motion of constrained Lagrangian dynamics. However, despite the possibility of parallelizing some of the steps (e.g., contact force evaluations), the algorithm scales poorly. Cirio et al. [**6**] improve upon this aspect by using a compact representation of yarn-based cloth using persistent contacts. Their representation is based on the observation that yarn threads have a select few points of persistent contact: these points can thus be represented by nodes with some restriction on their motion based on the yarn threads that reach them. Yarns can then simply be represented by straight segments. Using this method, their simulations run nearly an order of magnitude faster than previous work.

Given the time stepping nature of simulations, it is possible for the loop-loop topology of knitted cloth to erroneously change after a time step. Naturally, this will lead to incorrect results as the topology is intrinsically linked to the macroscopic behaviour of the cloth. The use of large time steps to reduce computation time of a simulation only further exacerbates the issue. Loop-loop topology can be more formally described using linking numbers, a way of counting the oriented linkages between two closed curves. Qu et al. [**26**] analyze different methods of computing the linkage numbers between spline curves. These methods can be applied to knitted fabric, chainmail, or any type of loopy material. The methods differ in how they compute the linkage number, but all employ the same overall strategy:

- Determine which loops potentially intertwine based on whether or not their respective boundary volumes overlap or not;
- Discretize the spline curves into line segments while ensuring the process is link homotopic;
- Compute the linking number of potentially linked pairs of loops.

From their results, they note that the counting crossings and the Barnes-Hut methods are the more efficient ones they have tested.

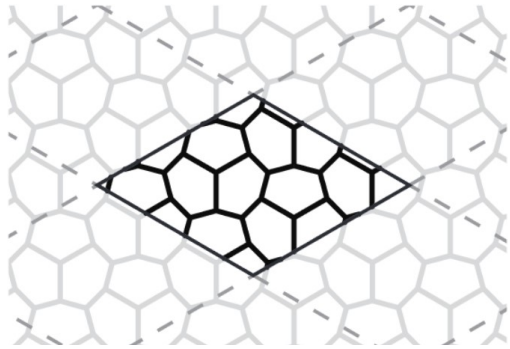


Fig. 2.3. Unit cell and tiling of a planar rod network [**31**].

2.2. Planar sheets and assemblies

Some planar materials, unlike cloth, do not exhibit nearly as much freedom of motion in the direction orthogonal to the plane due to their resistance to out of plane bending. Such materials thus require a different set of methods to be properly simulated and analyzed. Lu et al. [**19**] devise a theoretical framework of nonlinear continuum mechanics for two-dimensional graphene sheets under in-plane and bending deformations. They use their method to analyze the behaviour of single-atomic-layer graphene sheets under uniaxial stretch, cylindrical bending, and buckling under molecular mechanics simulation. Geers et al. [**14**] address the main principles needed to construct a computational homogenization scheme to upscale thin multi-layered sheets towards macroscopic shells. Schumacher et al. [**31**] use a Kirchhoff rod simulation to explore the properties of planar rod networks of varying designs to characterize their Young's modulus, Poisson's ratio, and bending stiffness. They are also able to derive structures with desired mechanical properties with a simple gradient-based optimization approach. For instance, they successfully obtain a structure with a target Young's modulus profile. We most notably adopt their use of homogenization to subject a tileable unit cell to states of deformation under the influence of boundary conditions to preserve the tileability of the mesoscopic scale leading to deformations on the macroscopic scale.

Kirchhoff-Plateau surfaces (KFS) are flexible structures made of elastic rods and stretched textile membranes that are assembled into a planar state and deploy into complex three-dimensional shapes. Designing such a structure in the goal of obtaining a specific shape is by no means trivial. To ease the design process, Pérez et al. [**25**] propose a user-guided, computer-assisted approach where the user is in charge of designing the structure and shape of the KPS while having access to editing and visualization tools to simplify the task. These helper tools are supported by their computational model that represents rods as Catmull-Rom splines and models membranes as a St. Venant-Kirchhoff material to accurately predict equilibrium configurations of the KPS and how a change of parameters influences those configurations. Similarly, Skouras et al. [**32**] develop an interactive tool for the design of

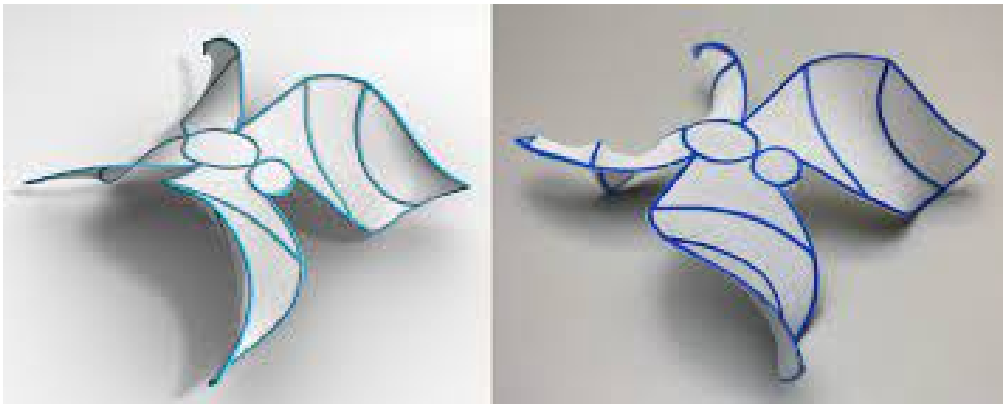


Fig. 2.4. Simulation (left) and physical experiment (right) of KFS [**25**].

three-dimensional shapes from two-dimensional interlocking elements that bend, but do not stretch. Although a single type of element is used, they can be assembled in a number of different ways allowing for a variety of shapes to arise. While designing, a user only needs to work with basic shapes that can be merged or extruded. By leveraging a half-edge data structure combined with a penalty approach, they can render the appearance of the final product once it is assembled from the flat elements and also automatically come up with an order of instructions to make assembly easier.

Lensgraf et al. [**17**] study the kinematic behaviour of planar assemblies of puzzle pieces connected through loose joints. They define a linearized distance function between vertices of the polygons that make up the bodies in the chain and collision points. This distance function, in turn, is used to derive a constraint on the system to define a linear program in which the target of optimization is a configuration-space direction in which motion is possible. This direction is then used in an Euler step to move the system in the given direction. This is done iteratively until the system reaches an equilibrium. They show that their approach can be used to explore joint tolerance of systems by parameterizing the tightness of joints and performing binary search. They also successfully use their algorithm to simulate flock formation of robots and show that the algorithm may be used for assembly (and disassembly) of linked elements. They note that due to their geometric-based approach, the motion of the elements in the system may not be the most realistic, but it allows fast and interesting optimizations due to the linear constraints describing a space of possible motions.

2.3. Structured materials

The design and analysis of structured materials has gained a rise in popularity in the past years due to manufacturing technologies such as 3D printing becoming more readily available. Likewise, metamaterials and interlocking materials are also gaining in popularity as their unique properties are becoming more well-known. Even simple convex building blocks such

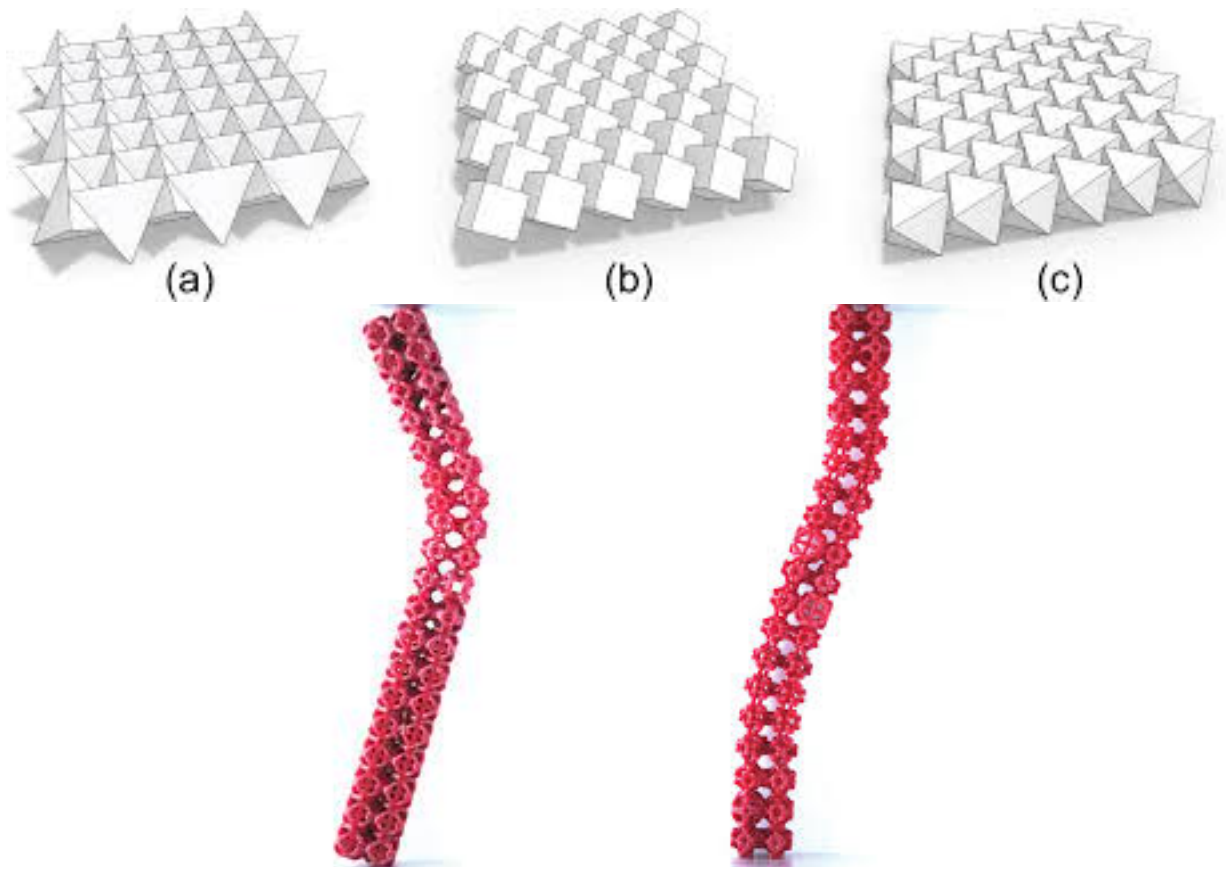


Fig. 2.5. Top: three topologically interlocked assemblies of tetrahedra, cubes, and octahedra respectively [**36**]. Bottom: two rods with different arrangements of volumetric textures leading to different behaviours under pressure [**23**].

as tetrahedra can display a wealth of characteristics when assembled that would otherwise not be present in a monolithic block of the same substance. Despite being interlocked, the building blocks can still slide, rotate, or separate to some degree, thus providing many tunable deformation mechanisms and properties. Moreover, this extra range of motion allows the building blocks to dissipate energy through frictional sliding, often making these kinds of materials more impact resistant than their monolithic counterparts. However, improvements in impact resistance and energy absorption (i.e., toughness) are usually accompanied by losses in material strength. Mirkhalaf et al. [**21**] systematically explore topologically interlocked panels made of convex ceramic blocks and identify an architecture based on octahedral blocks that displayed much higher toughness (50-fold), but also higher strength ($\sim$20%). They also observe that the monolithic material tends to fail in a much more catastrophic and brittle manner, as opposed to the interlocking material which fails in a much more localized fashion. However, they note that the exploration of the large design space of possible architectures is rather difficult since their (physical) experiments are lengthy, which brings us to numerical models.

Wang et al. [**36**] study topological interlocking assemblies made of convex building blocks. Given an input surface and a two-dimensional tessellation, their method derives a manner in which the surface can be reproduced by interlocking blocks shaped in accordance to the tessellation. By using a global interlocking test and static equilibrium analysis, their method also optimizes the structure such that it can withstand loads in a greater set of directions, making them more stable. Wang et al. [**37**] develop a graph model to describe the interlocking nature of interlocking assemblies. With their model, they can determine if an object may be disassembled into its constituent parts or not. Furthermore, they come up with a design framework that builds upon the graph model that, given a full input model, iteratively splits off parts for disassembly in a tree traversal process. Given an assembly and its parts, it can also determine how they should interlock to be a proper interlocking assembly (i.e., once assembled, only one of the parts can move w.r.t. to the others).

Although their method does not involve disjoint building blocks, Andreassen et al. [**1**] develop an optimization routine to minimize the Poisson ratio of materials assembled from a single unit cell tiled in all directions at periodic intervals. Their approach leverages first-order FEM by adding a design variable to each element which determines if the element is void or not. Through a standard topology optimization approach, the unit cell is then essentially hollowed out from iteration to iteration, leading to a change in the macroscopic behaviour of the tiled material. Schumacher et al. [**30**] develop a method to make the design of 3D printed metamaterials more accessible: they begin by designing families of substructures that exhibit desired properties with the additional benefit that these substructures may be interpolated to smoothly vary the materials' properties. Through the use of a global optimization algorithm, they can then select from these substructures to create objects with specific shapes and behaviours. Similarly, Panetta et al. [**23**] explore how objects with complex material properties could be assembled from basic volumetric textures. Their goal is to create objects that display heterogenous properties, e.g., a rod that is much more susceptible to bending at a specific point along its axis (see bottom of Figure 2.5). By defining a family of pattern topologies, they then perform a shape optimization so that these topologies may match specific elasticity parameters. The textures can then be used to fabricate objects manually or automatically possessing pre-specified behaviour with a local-global optimization. The complex material properties are thus attained by selecting the correct combination of textures throughout the object, which differs from other work where typically a single unit cell (i.e., texture) is selected and its macroscopic behaviour when tiled is analyzed. In later work, Panetta et al. [**22**] address the issue that such microstructures may have, namely the high stress that softer topologies may experience at specific joints, often leading to fracture. By deriving the exact solution to the worst-case stress analysis problem and introducing their own parametric shape model, they achieve robust minimization of

maximal stress in microstructures without the side-effect of altering the microstructures' macroscopic behaviour.

Bickel et al. [**3**] tackle the design problem of fabricating objects with specific behaviour to deformation by subsequently layering different materials. Their main contribution is a goal-based design approach that replicates deformation behaviour at the mesoscale by means of combinatorial optimization. By measuring the deformation of real world materials and then using those measurements as a reference point for their method, they can design objects with specific responses to deformation and fabricate them using a 3D multi-material printer. This is made possible by their optimization routine which uses a branch-and-bound with clustering approach to select an ideal combination of materials to reproduce the set of specified deformations. In such iterative design algorithms, the use of FEM is a major bottleneck. Even with the use of mesh coarsening and model reduction, such methods require expensive precomputation phases that themselves require *a priori* knowledge of an object's geometry and material composition. Naturally, for design tasks, this knowledge is not usually known. Chen et al. [**5**] propose Data-Driven FEM, a new simulation methodology to address this issue. The key feature of their method is the use of a custom metamaterial database learned in an offline fashion that is used to perform fast coarsening of meshes at runtime leading to significant speed gains (up to 2 orders of magnitude) by reducing the time of precomputation stages.

Chapter 3

Distance between elements

Our method is an energy-based approach that attempts to reach a compromise between the deformation of a DIM and the minimization of intersections between elements, all the while maintaining the DIM's topology. To appropriately penalize element interpenetration, we must first be able to recognize it. To achieve this, we leverage the definition of a torus to our advantage:

Definition 3.0.1 (Torus and related definitions)**.** *A torus T is a geometrical shape whose surface can be defined as the set of points lying at a distance r from a circle C. The circle C has radius R, normal N and center c. For our purposes, $R > r$ will always hold. The major radius of T is R and the minor radius of T is r.*

Using this definition, we devise a method to compute intersections between pairs of tori that builds upon the distance between their respective circles C. Furthermore, we also derive an equation which, when solved, answers the question of whether or not two tori interlock. A circle in 3D space is a function of an angle yielding points. Hence, before deriving the solution to the circle-to-circle distance problem, we quickly review how to derive the distance from a point to a circle.

3.1. Distance between a circle and a point

The distance d between a circle C (of center c and normal N) and a point P is rather trivial to compute. Consider the plane defined by c and N. Project P onto this plane to obtain Q. Then the closest point K on C to Q is

$$K = c + r(Q - c)/|Q - c|.$$

The distance between C and P is then

$$d = |P - K|.$$

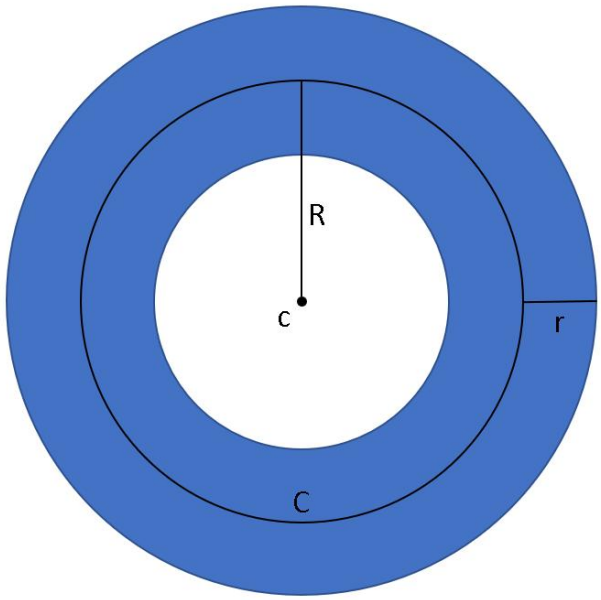


Fig. 3.1. A torus with circle C, center c, major radius R, and minor radius r.

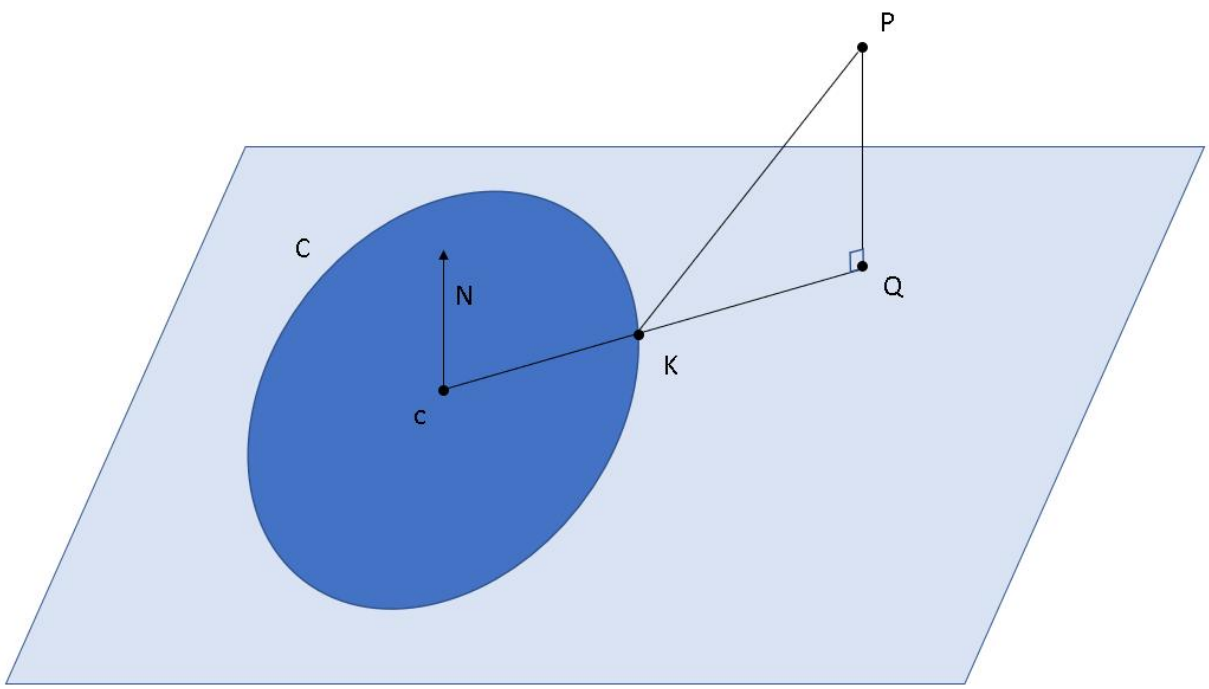


Fig. 3.2. The distance between a circle C and a point P is the distance between points P and K.

By defining $\Delta = P - c$, we can formulate d as a function of C's parameters and P [**11**]

$$\begin{aligned} |P-K|^2 &= |P-Q|^2 + |Q-K|^2 && (3.1.1)\\ &= (N\cdot\Delta)^2 + (|Q-c|^2 - R)^2 \\ &= (N\cdot\Delta)^2 + (\sqrt{|\Delta|^2 - (N\cdot\Delta)^2} - R)^2 \\ &= (N\cdot\Delta)^2 + (|N\times\Delta| - R)^2. && (3.1.2)\end{aligned}$$

3.2. Distance between a circle and a circle

To compute the shortest distance between two circles C_1 and C_2, we can set P to be a function of an angle θ whose image spans the set of points defined by C_2 [**11**]. To accomplish this, we define vectors U and V s.t. the set $\{U, V, N_2\}$ is right handed and orthonormal. The definition of $P(\theta)$ is then

$$P(\theta) = c_2 + R_2\Big(\cos\theta U + \sin\theta V\Big) \tag{3.2.1}$$

and the definition of $\Delta(\theta)$ is

$$\Delta(\theta) = c_2 - c_1 + R_2\Big(\cos\theta U + \sin\theta V\Big). \tag{3.2.2}$$

We can substitute Δ in Equation 3.1.2 for Equation 3.2.2 to then solve for θ by differentiating w.r.t. θ, setting to zero and solving for θ.

$$\Delta'(\theta) = R_2\Big(-\sin\theta U + \cos\theta V\Big) \tag{3.2.3}$$

$$\begin{aligned} g &= \frac{d}{d\theta}|P - K|^2 \\ &= 2(N_1 \cdot \Delta(\theta))(N_1 \cdot \Delta'(\theta)) + 2(|N_1 \times \Delta(\theta)| - R_1)\frac{N_1 \times \Delta(\theta) \cdot N_1 \times \Delta'(\theta)}{|N_1 \times \Delta(\theta)|} \\ &= 0 \end{aligned} \tag{3.2.4}$$

After some rearrangements (see Appendix A), we obtain a polynomial of the term $\cos\theta$ which can be solved for by using root-finding algorithms. The possible values of θ derived through $\cos\theta$ are plugged into Equation 3.2.2 to obtain a set of points, one of which (or possibly more) will have minimal distance to C_1.

3.3. Detecting overlap between pairs of tori

We can define a score function between a pair of tori T_1 and T_2 from the distance of their respective circles C_1, C_2 allowing to detect whether or not they overlap.

Theorem 3.3.1 (Overlap between two tori)**.** *Given tori T_1 and T_2 with respective circles C_1, C_2, the value d_c represents the minimal distance from C_1 to C_2. Assume this distance is from points p_1 to p_2. The line segment connecting p_1 and p_2 goes through point t_2 on T_2. Take the value*

$$d = d_c - r_1 - r_2.$$

When $d \geq 0$, there is no overlap between T_1 and T_2. When $d < 0$, overlap does occur.

Proof. We begin by proving the first claim by contradiction: assume that despite $d \geq 0$, overlap does occur. If overlap occurs, a point t_2 on T_2 is at a distance $d_t < r_1$ from C_1. Since

t_2 is at a distance of r_2 from C_2, $d'_c = d_t + r_2$ is a distance from C_1 to C_2. We thus have

$$d'_c = d_t + r_2 < r_1 + r_2$$
$$\equiv d'_c - r_1 - r_2 < 0$$

and since d_c is minimal

$$d = d_c - r_1 - r_2 \leq d'_c - r_1 - r_2 < 0$$

which contradicts the fact that $d \geq 0$. Since this contradiction arises from our assumption that overlap does occur, no overlap must occur.

Now we prove the second claim. If $d < 0$, then $d_c < r_1 + r_2$. Therefore, t_2 is at a distance less than r_1 from C_1, therefore t_2 is inside T_1, therefore T_1 and T_2 overlap. □

Theorem 3.3.1 states how to determine whether or not overlap occurs. However, there might be multiple pairs of points at which overlap is occurring, i.e., different sections of the tori. Both these issues are addressed by simply taking into account all the different values of θ obtained by solving for the roots of the polynomial derived from Equation 3.2.4 and considering all scores which turn out negative between the tori. Since the obtained scores are either local minima or local maxima (we consider all extrema of the distance function between C_1 and C_2) this is also a valid way to compute overlap:

Proposition 3.3.2 (Local overlap of tori)**.** *Given tori T_1 and T_2 with respective circles C_1, C_2, the value d_c represents a locally minimal distance from C_1 to C_2. Assume this distance is from points p_1 to p_2. The line segment connecting p_1 and p_2 goes through point t_2 on T_2. Take the value*

$$d = d_c - r_1 - r_2.$$

When $d \geq 0$, there is no local overlap between T_1 and T_2. When $d < 0$, local overlap does occur.

Proof. Same as the proof of Theorem 3.3.1 considering the locally shortest distance instead of the globally shortest one. □

3.4. Maintaining the interlocking relationship between tori

It is important that throughout a simulation, our method maintains the interlocking relationship between elements.

Definition 3.4.1 (Neighbour elements)**.** *Two elements are said to be neighbours if and only if they interlock.*

At the same time, we do not want tori that were not neighbours at the start of the simulation to interlock after a step of the simulation. To prevent any changes in the neighbour

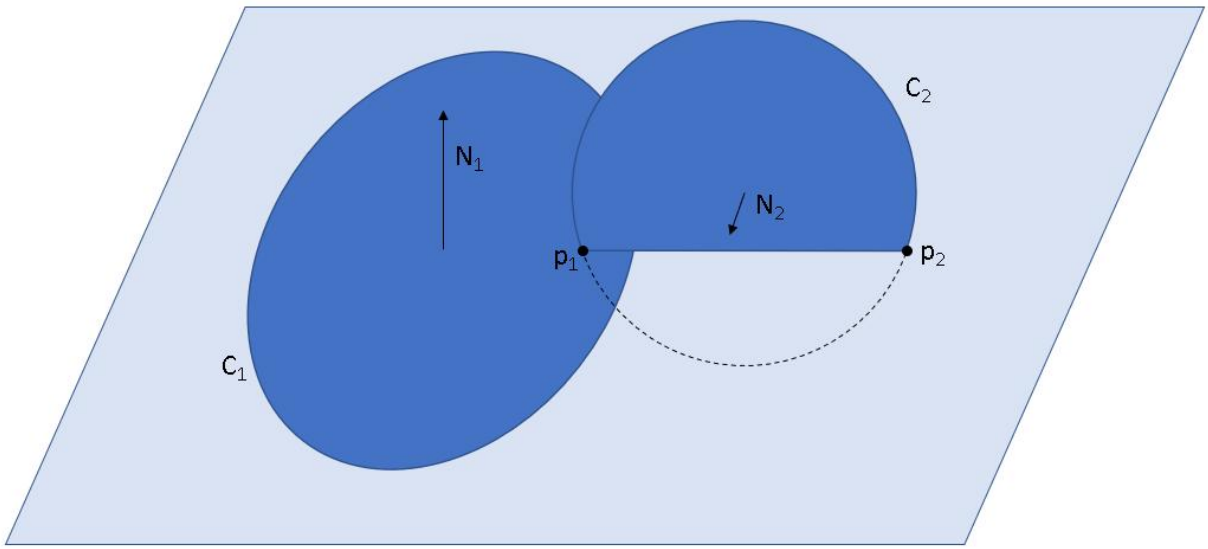


Fig. 3.3. Two interlocking circles C_1 and C_2. The dotted portion of C_2 is the arc behind the plane on which C_1 lies from our point of view.

relationship, or lack thereof, between pairs of tori, we need a more formal description of what it means for two elements to interlock:

Definition 3.4.2 (Interlocking)**.** *Given two tori T_1 and T_2, T_1 and T_2 are said to be interlocking if the circle C_2 of T_2 has two points p_1, p_2 lying on the plane defined by center c_1 and normal N_1. Furthermore, either of p_1 or p_2 needs to be at a distance less than R_1 from c_1 while the other must be at a distance greater than R_1 from c_1. Since the interlocking relationship is mutual, this naturally also applies from T_2 to T_1.*

Such points p will satisfy the equation

$$N_1 \cdot (p - c_1) = 0. \tag{3.4.1}$$

We can solve for p using a similar approach to the one used in Equation 3.1.2

$$N_1 \cdot \Big(p(\theta) - c_1\Big) = N_1 \cdot \Big(c_2 + R_2 \cos\theta U + R_2 \sin\theta V - c_1\Big) = N_1 \cdot \Delta(\theta) = 0. \tag{3.4.2}$$

Taking the square and then differentiating w.r.t. θ

$$\frac{d}{d\theta}\Big(N_1 \cdot \Delta(\theta)\Big)^2 = \Big(N_1 \cdot \Delta(\theta)\Big)\Big(N_1 \cdot \Delta'(\theta)\Big) = 0. \tag{3.4.3}$$

Solving for θ in Equation 3.4.3 will yield multiple points, some of which may be solutions to Equation 3.4.2. If none are found, then the corresponding tori do not interlock. If some are found, then additional checks must be done to ensure that the distance conditions of Definition 3.4.2 are met.

To solve Equation 3.4.3, let $\gamma = \cos\theta$, $\sigma = \sin\theta$, and $D = c_2 - c_1$.

$$
\begin{aligned}
\Big(N_1 \cdot \Delta(\theta)\Big)\Big(N_1 \cdot \Delta'(\theta)\Big) =& \gamma(N_1 \cdot D)(N_1 \cdot R_2 V) \\
&+ \gamma^2(N_1 \cdot R_2 U)(N_1 \cdot R_2 V) + (1-\gamma^2)(-N_1 \cdot R_2 U)(N_1 \cdot R_2 V) \\
&- \sigma[(N_1 \cdot D)(N_1 \cdot R_2 U) - \gamma(N_1 \cdot R_2 V)(N_1 \cdot R_2 V) + \gamma(N_1 \cdot R_2 U)(N_1 \cdot R_2 U)] \\
&= 0
\end{aligned}
\tag{3.4.4}
$$

First we rearrange the terms

$$
\begin{aligned}
&\gamma(N_1 \cdot D)(N_1 \cdot R_2 V) + \gamma^2(N_1 \cdot R_2 U)(N_1 \cdot R_2 V) + (1-\gamma^2)(-N_1 \cdot R_2 U)(N_1 \cdot R_2 V) \\
&= \sigma[(N_1 \cdot D)(N_1 \cdot R_2 U) - \gamma(N_1 \cdot R_2 V)(N_1 \cdot R_2 V) + \gamma(N_1 \cdot R_2 U)(N_1 \cdot R_2 U)].
\end{aligned}
\tag{3.4.5}
$$

Then we square both sides

$$
\begin{aligned}
&\Big(\gamma(N_1 \cdot D)(N_1 \cdot R_2 V) + \gamma^2(N_1 \cdot R_2 U)(N_1 \cdot R_2 V) + (1-\gamma^2)(-N_1 \cdot R_2 U)(N_1 \cdot R_2 V)\Big)^2 \\
&= \sigma^2[(N_1 \cdot D)(N_1 \cdot R_2 U) - \gamma(N_1 \cdot R_2 V)(N_1 \cdot R_2 V) + \gamma(N_1 \cdot R_2 U)(N_1 \cdot R_2 U)]^2 \\
&= (1-\gamma^2)[(N_1 \cdot D)(N_1 \cdot R_2 U) - \gamma(N_1 \cdot R_2 V)(N_1 \cdot R_2 V) + \gamma(N_1 \cdot R_2 U)(N_1 \cdot R_2 U)]^2 \\
&\equiv p_1 = \sigma^2 p_2 = (1-\gamma^2)p_2.
\end{aligned}
\tag{3.4.6}
$$

Finally, Equation 3.4.6 can be rearranged into a polynomial of γ.

$$
p_1 - (1-\gamma^2)p_2 = 0.
$$

We can solve for the roots of this polynomial using a root finding algorithm and subsequently solve for θ.

We are now equipped with the tools to measure intersection between elements and penalize it. Likewise, we can determine whether or not two elements interlock after a time step and adjust the step size if topology violations arise. In the next chapter, we shall cover how our simulation method works.

Chapter 4

System energy, gradient, and optimization

In this chapter, we present the different components of our energy function and how they are computed. The two main contributors of the energy are a term penalizing negative scores (i.e., intersection) between pairs of elements and a force inducing a desired deformation in the DIM. In the goal of simulating infinite DIMs, we adopt an approach based on homogenization involving a repeatedly tiled unit cell; this unit cell must meet certain conditions for the tiling to remain valid, which introduces a third term to our energy. We use Newton's method to minimize the energy. Hence, we also cover how we parameterize the DIM under simulation as well as how the gradient and Hessian are derived based on this parameterization.

4.1. System energy

We define an energy for the system under simulation that we then wish to minimize. There are three components to this energy:

- energy of the contacts between elements;
- energy to maintain the periodic boundary conditions;
- energy based on an external force.

We now detail how each component is defined.

4.1.1. Energy from contacts

The goal of this energy (E_c) is to penalize interpenetration between elements: contacts between elements are allowed, but we wish to limit the extent to which an element can move into another. In Section 3.3, we discussed how to compute a score between two individual tori. As a reminder, the method yields a set of values that are all extrema of the score function between both tori (see Proposition 3.3.2): some of the values are locally minimal while others are locally maximal. In our case, we are interested in the ones that are negative.

A straightforward approach to penalize these would be to compute the energy from contacts as the score d squared, scaled by some arbitrary factor.

$$E_c = \begin{cases} \alpha d^2, & \text{if } d < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4.1.1}$$

However, this formulation is not C^2 continuous and since we wish to use Newton's method for optimization (as we will explain in Section 4.3), we look for a more suitable alternative. In their work, Li et al. [**18**] use the following barrier function

$$b(\delta, \hat{\delta}) = \begin{cases} -(\delta - \hat{\delta})^2 \ln \frac{\delta}{\hat{\delta}}, & \text{if } 0 < \delta < \hat{\delta} \\ 0, & \text{otherwise} \end{cases} \tag{4.1.2}$$

where $\hat{\delta}$ is the minimum distance that will not be penalized between two objects and δ is the actual distance between them. Looking at Proposition 3.3.2, our torus-to-torus score function is based on the distance between the tori's respective circles' C and their respective minor radii r. We can modify the above barrier function to fit our needs: set $\hat{\delta} = r_1 + r_2$ since the case where $d_c = r_1 + r_2$ is the case where contact happens between the two tori, but no interpenetration does. As for δ, it is simply equivalent to d_c in Proposition 3.3.2.

$$\begin{aligned} E_c &= \begin{cases} -\alpha(d_c - r_1 - r_2)^2 \ln \frac{d_c}{r_1 + r_2}, & \text{if } 0 < d_c < r_1 + r_2 \\ 0, & \text{otherwise} \end{cases} \\ &= \begin{cases} -\alpha d^2 \ln \frac{d_c}{r_1 + r_2}, & \text{if } d < 0 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.1.3}$$

4.1.2. Additional contacts

Overlap between two elements' volumes is itself a volume, yet we are penalizing it with what is more akin to a distance measurement. It is a valid way to penalize overlap since as long as the score between elements is positive, no overlap occurs. It leaves a bit to be desired when overlap does occur though. Say two elements T_1 and T_2 mutually overlap with points P_1 and P_2 being their respective closest points on their respective circles. Consider the axis A passing through center c_2 and P_2. We could rotate T_2 about A while maintaining P_1 and P_2 as the closest points. This results in the score between T_1 and T_2 to remain the same, but the overlap volume likely does not. In our tests, this could be observed by having neighbour elements lie in contact in orientations that were suboptimal.

We resolve this issue by sampling additional points about P_2 as Drumwright et al. [**9**] suggested to resolve instabilities in their solver. This extra set of points does not lie on A

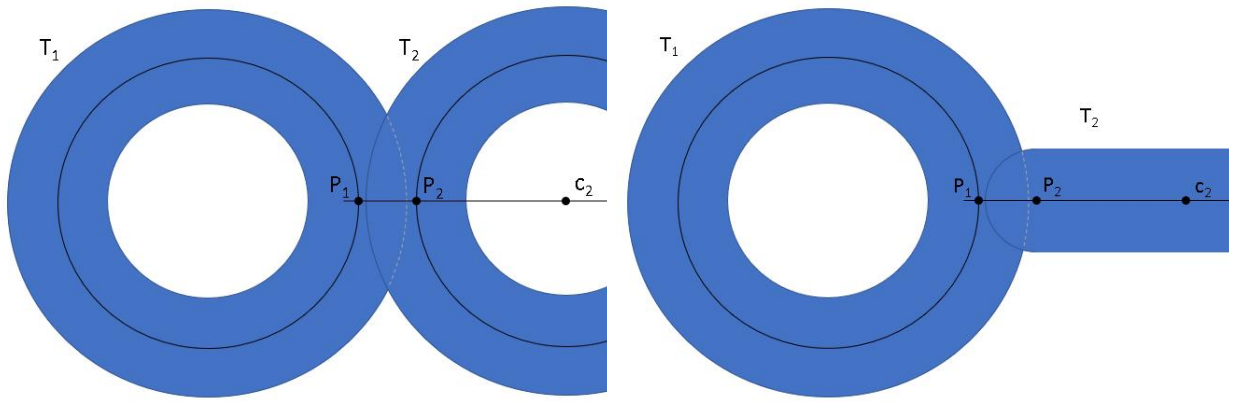


Fig. 4.1. By rotating T_2 about the axis going through c_2 and P_2, the overlap volume between T_1 and T_2 changes, but the distance between P_1 and P_2 remains the same.

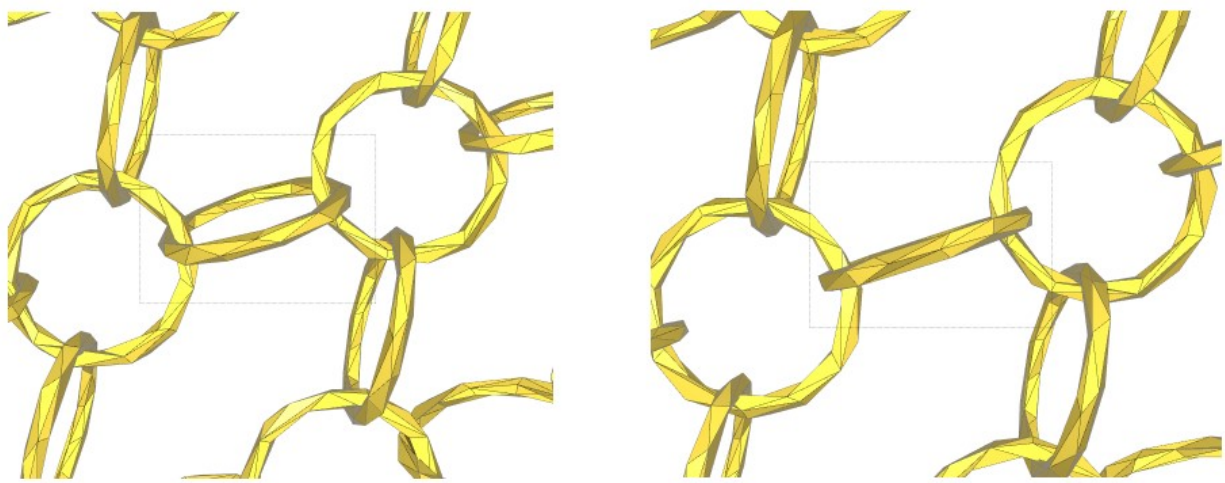


Fig. 4.2. On the left, we can see that the element is not aligned perpendicularly to its neighbours which increases the volume of intersection between them. On the right, we can see that once we incorporate non-minimal points into the computation of contact energy, the elements reach an alignment that is much more adequate to minimize interpenetration.

and hence proofs the method against the above described scenario. We shall refer to such points as *non-minimal points* from now on.

Definition 4.1.1 (Non-minimal points)**.** *A pair of non-minimal points p_1, p_2 are points (on circles C_1, C_2 respectively) whose score is not minimal, but still yield interpenetration of their respective elements T_1, T_2.*

Non-minimal points are chosen by sampling at fixed intervals along C_2. Thus, given that $P_2 = c_2 + R_2 \cos\theta U + R_2 \sin\theta V$ then a non-minimal point P_2' would be

$$P_2' = c_2 + R_2 \cos(\theta + \delta)U + R_2 \sin(\theta + \delta)V. \tag{4.1.4}$$

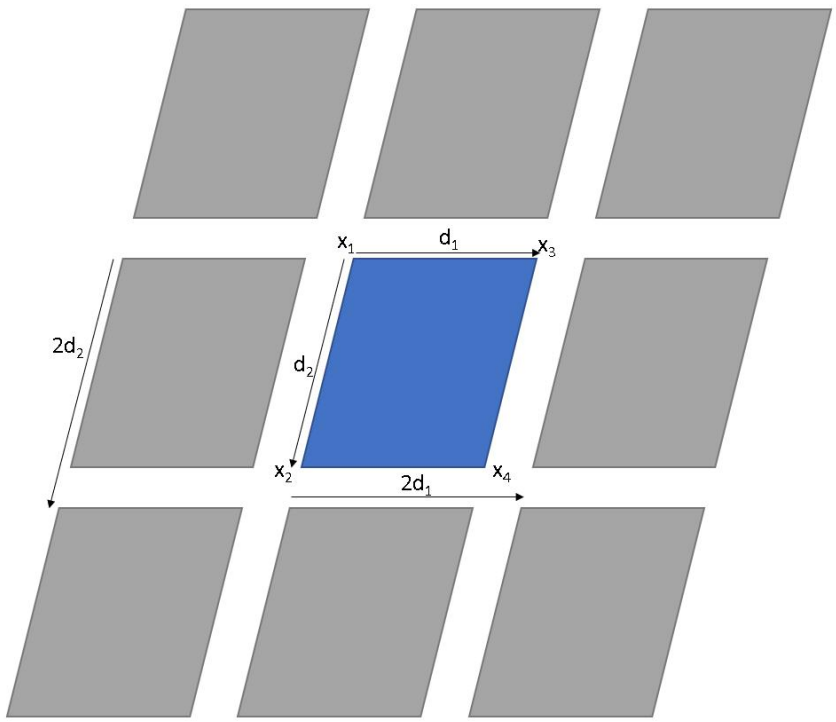


Fig. 4.3. The main unit cell (in blue) is used to determine the position of all other elements by setting the position of other cells based on d_1 and d_2. We simulate a layer of unit cells (in gray) neighbouring the main unit cell to more accurately approximate the behaviour of elements far away from the boundary of an arbitrarily large sheet of discrete interlocking elements.

The distance from P_2' to C_1 is computed as in Equation 3.1.2. This distance is then penalized as in Equation 4.1.3. Note that since P_2' is not minimal, Proposition 3.3.2 does not apply to it. However, this is sufficient for our goal of resolving suboptimal orientations.

4.1.3. Periodic boundary conditions

Ultimately, we are interested in the mechanical behaviour of sheets of discrete interlocking material on a macroscopic level. Simulating a small sheet of material would not give satisfactory results since elements on the boundary of the sheet are much less constrained than those closer to the center meaning the obtained results would not be representative of those obtained on a much larger sheet. On the flip side, simulating a large sheet quickly becomes computationally expensive as contacts increase quadratically with the number of elements and so do the validity checks for neighbours (see Section 3.4). To decouple a sheet's size from its macroscopic behaviour, we turn to periodic boundary conditions.

The overall idea is to simulate a single unit cell comprised of few elements which is then padded with an external layer of identical cells all around to act as the boundary of the sheet (see Figure 4.3). Elements on the boundary are thus entirely dependent on those

in the center of the sheet plus a translation term which is also derived from the unit cell. In turn, the original cell at the center is being stretched by its neighbouring cells. This relationship between cells on the boundary and the one in the center of the sheet gives a good approximation of what we would observe in a large sheet of material away from the boundary: multiple neighbouring cells pulling on each other while being in nearly identical configurations.

To ensure that the unit cell can be tiled in a valid manner, an energy E_b is introduced. Specifically, given a unit cell whose four corners are $\{x_1, x_2, x_3, x_4\}$, we want the following to hold

$$x_4 = x_1 + (x_2 - x_1) + (x_3 - x_1). \tag{4.1.5}$$

We thus simply define the energy contributed by the unit cell as

$$E_b = \Big| x_4 - \Big(x_1 + (x_2 - x_1) + (x_3 - x_1) \Big) \Big|^2 = \Big| x_4 + x_1 - x_2 - x_3 \Big|^2. \tag{4.1.6}$$

4.1.4. Energy from forces

Lastly, an energy E_f is also defined based on the specific deformation that we are interested in. We either define this as an energy acting on some or all elements as gravity would

$$E_f = \beta x \tag{4.1.7}$$

where x is the position of the element.

Alternatively, this energy can also be defined based on a target position x_T that we wish for an element to reach (or try to reach)

$$E_f = \beta |x - x_T|^2. \tag{4.1.8}$$

4.2. System parameterization

An element T will have a set of six degrees of freedom (DOFs) on which it depends: three for position and three for rotation. Its positional DOFs are simply its center c whereas the three rotational DOFs are Euler angles $\{\theta_x, \theta_y, \theta_z\}$.

Definition 4.2.1. *An element T depends on the set of DOFs q s.t.*

$$q = \{c_x, c_y, c_z, \theta_x, \theta_y, \theta_z\}$$

Euler angles are prone to lead to gimbal lock where different axes of rotation align with one another, but this does not occur when the angles of rotation are small. Thus, we opt

to have the rotational DOFs represent the rotation to be applied at the current step while the overall rotation R relative to the initial orientation of T up to the current state is kept separate. At each step, the values obtained through optimization for $\{\theta_x, \theta_y, \theta_z\}$ are then used to update R,

$$R := R_y R_x R_z R$$

where R_a is the rotation matrix about axis $a \in \{x, y, z\}$ by angle θ_a. Lastly, we apply rotation before translation.

Given a point p_i on T in its initial configuration, we can compute the point's current position p as

$$p = R_y R_x R_z R p_i + c. \tag{4.2.1}$$

Obtaining p_i from p is quite simple as the inverse of a rotation matrix is its transpose.

$$p_i = (R_y R_x R_z R)^T (p - c) \tag{4.2.2}$$

However, while computing the gradients, we are rarely interested in p_i, but in Rp_i instead.

$$p_c = Rp_i = (R_y R_x R_z)^T (p - c) \tag{4.2.3}$$

Using this parameterization makes computing the gradient of the distance between two points much easier and cleaner as we shall see in the next section.

4.3. Optimization

To minimize the energy of our system, we use Newton's method. This requires computing the gradient of the energy and its Hessian w.r.t. the DOFs defined in Section 4.2. The energy components E_f and E_b only depend on the positional DOFs c of elements and are rather trivial to derive. Hence, we shall not go into detail in regards to how we derive their contribution to the gradient. We begin by working out the definition of the gradients of the contact component of the energy, continue by looking at a few special cases where the gradient evaluation is more involved, and finish off by explaining how we obtain the Hessian.

4.3.1. Gradient of the contact energy

Given D, the square of the distance between two points on two elements' circles is

$$D = d_c^2 = |P_1 - P_2|^2 = (P_1 - P_2)_x^2 + (P_1 - P_2)_y^2 + (P_1 - P_2)_z^2 = D_x^2 + D_y^2 + D_z^2. \tag{4.3.1}$$

We can rewrite the equation of d in Theorem 3.3.1 as

$$d_t = \sqrt{D} - r_1 - r_2 = \sqrt{D} - r.$$

From Equation 4.1.3, if $d < 0$, then we have

$$E_c = -\alpha d_t^2 \ln \frac{\sqrt{D}}{r}$$

$$\frac{d}{dq}E = -\alpha d_t \frac{1}{\sqrt{D}} \frac{d}{dq} D \ln \frac{\sqrt{D}}{r} - \alpha d_t^2 \frac{1}{2D} \frac{d}{dq} D. \tag{4.3.2}$$

4.3.2. Gradient of the squared distance between two points

We wish to compute the gradient of D as defined in Equation 4.3.1. The derivation of this form of D involves fewer terms and is generally easier to handle than the form in Equation 3.1.2. First we further define

$$P_1 = R_{1y} R_{1x} R_{1z} p_{1c} + c_1$$

$$P_2 = R_{2y} R_{2x} R_{2z} p_{2c} + c_2$$

using Equations 4.2.1 and 4.2.3.

Points P_1 and P_2 were obtained by solving for θ as explained in Section 3.2. This θ also has a dependency on the set of DOFs q_1 and q_2 and we must thus account for it in the gradient. Hence the gradient of D has the following form

$$\frac{d}{dq}D = \frac{\partial D}{\partial q} + \frac{\partial D}{\partial \theta}\frac{\partial \theta}{\partial q}. \tag{4.3.3}$$

The term $\frac{\partial D}{\partial \theta}$ will be zero for points obtained directly from solving Equation 3.2.4 since $g = \frac{\partial D}{\partial \theta} = 0$ by definition at such points. However, for non-minimal points as described in Section 4.1.4, that is not the case. We thus need to compute the term $\frac{\partial \theta}{\partial q}$. There is no closed form expression of θ w.r.t. the set of DOFs q so we use sensitivity analysis to compute it.

We wish for $g = \frac{\partial D}{\partial \theta}$ to remain the same as q changes

$$\frac{d}{dq}g = \frac{\partial g}{\partial q} + \frac{\partial g}{\partial \theta}\frac{\partial \theta}{\partial q} = 0 \tag{4.3.4}$$

and from here we isolate $\frac{\partial \theta}{\partial q}$

$$\frac{\partial \theta}{\partial q} = -\frac{\partial g}{\partial q}\left(\frac{\partial g}{\partial \theta}\right)^{-1}. \tag{4.3.5}$$

For non-minimal points, the angle $\theta' = \theta + \delta$ has a linear relationship with θ.

$$\frac{\partial g}{\partial \theta'}\frac{\partial \theta'}{\partial q} = \frac{\partial g}{\partial \theta'}\frac{\partial \theta'}{\partial \theta}\frac{\partial \theta}{\partial q} = \frac{\partial g}{\partial \theta'}\frac{\partial \theta}{\partial q} \tag{4.3.6}$$

Hence, we only need to compute $\frac{\partial \theta}{\partial q}$ once per overlap volume. The value $\frac{\partial g}{\partial \theta}$ needs to be computed on a per-point basis, but it is rather cheap to do so. Refer to Appendix B for the full derivation of D.

4.3.3. Multi-ring elements

Some of the elements we are interested in are not composed of a single ring (tori), but of multiple. We shall call these multi-ring elements.

Definition 4.3.1 (Multi-ring elements)**.** *A multi-ring element (MRE) T is an element composed of multiple rings (tori) of possibly different sizes. We will refer to these rings as sub-elements. A sub-element T_s will be shifted by a fixed offset d from T s.t. its center is $c_s = c + Rd$, where R is the rotation T has been subjected to. Consequently, all sub-elements in T are subjected to the translations and rotations applied to T.*

The energy contribution of MREs can be computed in the same way when it comes to the components E_b and E_f, i.e., simply use the MRE's positional DOFs. For the contact energy E_c however, the approach is more involved: each sub-element of an MRE needs to be checked against all sub-elements of another MRE to detect contacts and overlaps between both MREs. Then, the contributions to the gradient need to be added together correctly since MREs will be displaced as a single unit at the optimization step.

Given an MRE T, consider one of its sub-elements T_s. Any point on the surface of T_s may be described using Equation 4.2.1: $p_s = R_{sy}R_{sx}R_{sz}R_m p_{si} + c_s$, where $R_m = RR_s$, the matrix product of the rotation R applied to T and the initial rotation R_s applied to T_s w.r.t. the frame of T. Contact energy E_c can be computed normally between sub-elements with respect to DOFs $q_s = [c_{sx}, c_{sy}, c_{sz}, \theta_{sx}, \theta_{sy}, \theta_{sz}]^T$. We have the following relationship between q and q_s

$$\begin{aligned}
\frac{\partial c_{sa}}{\partial c_a} &= 1 \\
\frac{\partial c_{sa}}{\partial \theta_b} &= \left(\frac{\partial}{\partial \theta_b} R_y R_x R_z R d\right)_a \\
\frac{\partial \theta_{sb}}{\partial c_a} &= 0 \\
\frac{\partial \theta_{sb}}{\partial \theta_b} &= 1
\end{aligned}$$

for $c_a \in \{c_x, c_y, c_z\}$ and $\theta_b \in \{\theta_x, \theta_y, \theta_z\}$.

By the Chain Rule, we have

$$\frac{\partial}{\partial q}E_c = \sum_s \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ (\frac{\partial}{\partial \theta_x} R_y R_x R_z R d)_x & (\frac{\partial}{\partial \theta_x} R_y R_x R_z R d)_y & (\frac{\partial}{\partial \theta_x} R_y R_x R_z R d)_z & 1 & 0 & 0 \\ (\frac{\partial}{\partial \theta_y} R_y R_x R_z R d)_x & (\frac{\partial}{\partial \theta_y} R_y R_x R_z R d)_y & (\frac{\partial}{\partial \theta_y} R_y R_x R_z R d)_z & 0 & 1 & 0 \\ (\frac{\partial}{\partial \theta_z} R_y R_x R_z R d)_x & (\frac{\partial}{\partial \theta_z} R_y R_x R_z R d)_y & (\frac{\partial}{\partial \theta_z} R_y R_x R_z R d)_z & 0 & 0 & 1 \end{bmatrix} \frac{\partial}{\partial q_s}E_c. \tag{4.3.7}$$

With Equation 4.3.7, we can easily compute the MRE's gradient contribution $\frac{\partial}{\partial q}E_c$ by computing the gradient contribution $\frac{\partial}{\partial q_s}E_c$ of each of its sub-elements.

The use of MREs can lead to cases where a neighbour check as described in Section 3.4 becomes insufficient to maintain the initial neighbour element relationships. Consider an MRE shaped like a gyroscope: a neihbouring element interlocking with a single of the MRE's sub-elements can be positioned in four ways, relative to the other sub-elements, while still maintaining the interlocking relationship. Fortunately, we can easily determine in which position the element lies: after solving for the point p on the inside of the MRE, as described in Definition 3.4.2, verify that p is on the correct side of the planes defined by the other sub-elements. Evaluating the dot product of the vector $p-c$ and the normals of the sub-elements allows us to determine on which side of their planes p resides. The signs of the dot products need to remain the same at every step of the simulation to preserve the DIM's topology.

4.3.4. Periodic boundary conditions

As described in Section 4.1.3, all elements in a sheet depend on the elements of a single, main unit cell. The unit cells adjacent to the main one are translated copies of the main unit cell. Rotational DOFs will be identical between an element in a copied unit cell and its corresponding element in the main unit cell. Positional DOFs of an element in a copied cell will depend linearly on some, if not all, elements in the main cell.

Given the gradient of the energy with respect to the DOFs of the sheet q_S, we can then obtain the gradient of the energy with respect to the DOFs of the main unit cell q_C

$$\frac{\partial E}{\partial q_C} = \frac{\partial E}{\partial q_S}\frac{\partial q_S}{\partial q_C}. \tag{4.3.8}$$

4.3.5. Hessian computation

Our approach to optimization is to use Newton's method to minimize the energy of the system

$$H dq = -\frac{\partial}{\partial q_C}E.$$

At each step, we move in the direction of dq. While $dq \cdot \frac{\partial}{\partial q_C} E$ is non-negative, we regularize the system by adding γI to H, where γ is a scalar that increases with each attempt, before resolving for dq once more.

We observed that using simple gradient descent to reach optimality was much slower than using Newton's method, even when taking into account the time to compute the Hessian. Moreover, gradient descent tends to reach points of zero gradient that visually do not appear to be minimal.

We compute the Hessian using a finite difference approximation

$$\frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{g_i(x + \epsilon e_j) - g_i(x + \epsilon e_j)}{4\epsilon} + \frac{g_j(x + \epsilon e_i) - g_j(x + \epsilon e_i)}{4\epsilon} \tag{4.3.9}$$

where g_i is the value of the gradient at position i, e_i is the unit vector with position i set to 1 and the rest 0, and ϵ is a small positive scalar. We use $\epsilon = 1e - 7$.

From time to time, the optimization routine will converge to saddle points of the energy function where the gradient is of magnitude 0, but the energy is not minimal. For instance, consider the case where two elements are being pressed into one another with perfectly antiparallel updates. Shifting one slightly in a direction perpendicular to this motion would then allow the two elements to keep moving unhindered.

To solve this, we adopt Duenser et al.'s approach [**10**]: if a point of equilibrium is reached, we test the Hessian of the system's energy for indefiniteness by evaluating its eigenvalues. If any eigenvalue is negative, we perturb the system of elements by shifting all DOFs in the direction of the most negative eigenvalue's eigenvector repeatedly until the Hessian has no more negative eigenvalues. Then we proceed with the minimization once more.

Algorithm 4.3.2. Simulation algorithm for sheets of discrete interlocking elements

---

Initialize elements
Initialize forces
While !converged
    Compute energy E_{cur} and gradient $grad$
    Compute Hessian H approximation
    Do
        Solve for dq in $Hdq = -grad$
        $H = H + \epsilon I$
        $\epsilon = 2\epsilon$
    While $dq \cdot grad \geq 0$
    Do
        $q = q + \delta dq$
        $\delta = \delta/2$
        Update elements that are not part of the main unit cell
        Compute new energy E_{new}
    While ($E_{new} > E_{cur}$ **or Neighbour constraint not satisfied)**
    if grad == 0 and H indefinite
        Perturb system

Chapter 5

Macromechanical characterization

Our goal is to explore the impact of topology and element shape in the macromechanical deformation limits of a DIM. Hence, we require a manner to induce deformation in a DIM as well as a method to evaluate the degree of deformation that is observed in said DIM. In this chapter, we explain our approach in doing so: we set up a number of planar deformation experiments to explore the configuration space of various designs of rigid DIMs. At the beginning of each experiment, the corner elements of the sheet of DIM are given position constraints as described in Section 4.1.4. Our minimization routine is run until a point of zero gradient is reached where the Hessian is not indefinite, at which point we measure the deformation of the sheet.

5.1. Constraint definition

The deformation constraints are set up in a way such as to stretch the sheet out in a given direction D that is parallel to the sheet. The extent to which a DIM can be stretched in the direction of D is generally dependent on the elements' relative positions to one another in the in-plane direction perpendicular to D. We shall label this second direction D_{perp}. The constraints put on the four corners of the sheet (sheets are typically of rectangle or parallelogram shape) depend on both D and D_{perp}.

Given the center of mass C_m of the sheet (the average position of all elements), we define a vector v_i for each corner element $\{e_1, e_2, e_3, e_4\}$ pointing from C_m to the corresponding element's starting position. The components of v_i parallel to D and D_{perp} are then derived and scaled by pre-defined scalars to yield the element's target position

$$p_t = C_m + (1 + l_1)\frac{v_i \cdot D}{|D|^2}D + (1 + l_2)\frac{v_i \cdot D_{perp}}{|D_{perp}|^2}D_{perp} \tag{5.1.1}$$

where l_1 and l_2 determine how far in a direction the sheet shall be stretched.

A given target position is not always attainable. We wish that in that case, our minimization routine finds a configuration where neither direction D or D_{perp} is disproportionately

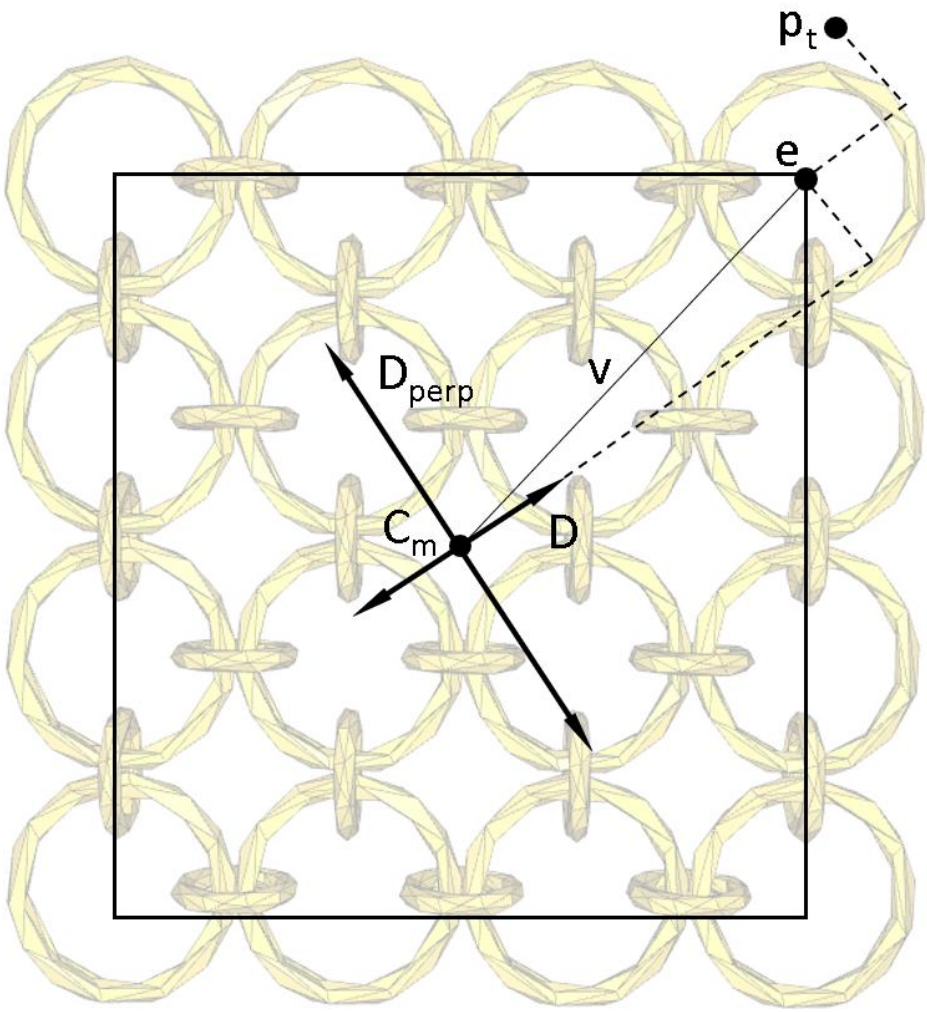


Fig. 5.1. Target position p_t of element e.

penalized with respect to the other. For a given value of l_2, l_1 is incrementally increased until any of the corner elements cannot be positioned within a certain threshold distance of its target position when the simulation ends. Elements are reset to their starting configuration whenever l_1 is incremented.

5.2. Deformation measurement

We evaluate sheet deformation by adopting the usual deformation measurement used in FEM and applying it to two-dimensional space. Given a quadrilateral, define its undeformed corners as $\bar{\mathbf{x}} = (\bar{x}, \bar{y})^T$, its deformed corners as $\mathbf{x}(\bar{\mathbf{x}}) = (x(\bar{x}, \bar{y}), y(\bar{x}, \bar{y}))^T$, and the corners' displacement vectors as $\mathbf{u}(\bar{\mathbf{x}}) = (u(\bar{x}, \bar{y}), v(\bar{x}, \bar{y}))^T$.

We compute the matrix

$$\mathbf{F} = \nabla \mathbf{u} + \mathbf{I} = \begin{bmatrix} \frac{\partial u}{\partial \bar{x}} & \frac{\partial u}{\partial \bar{y}} \\ \frac{\partial v}{\partial \bar{x}} & \frac{\partial v}{\partial \bar{y}} \end{bmatrix} + \mathbf{I} \tag{5.2.1}$$

The eigenvalues of $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ describe the deformation to which the sheet was subjected relative to its initial state (note that we report deformation as $\lambda - 1$ in Chapter 6). Since we always have $l_1 \geq l_2$ in our experiments, the larger eigenvalue usually pertains to the principal deformation direction D whereas the smaller one to the perpendicular direction D_{perp}. When this is the case, we usually have

$$\lambda_1 - 1 \sim l_1$$

$$\lambda_2 - 1 \sim l_2$$

where $\lambda_1 > \lambda_2$. In some cases, deformation along D is more constrained than along D_{perp} and the larger eigenvalue describes the deformation along D_{perp} instead. Hence, the above similarities will not hold.

We wish to stress the fact that this deformation is relative to the initial state of the sheet of DIM. DIMs do not have a true rest state: there are infinitely many valid configurations in which a DIM can lie at rest while free from the influence of external forces. Hence, the measurements we obtain through our experiments are inherently dependent on the initial configurations we arbitrarily defined. To keep our results meaningful, we maintain the relative position of elements the same when we tune parameters of specific designs.

Chapter 6

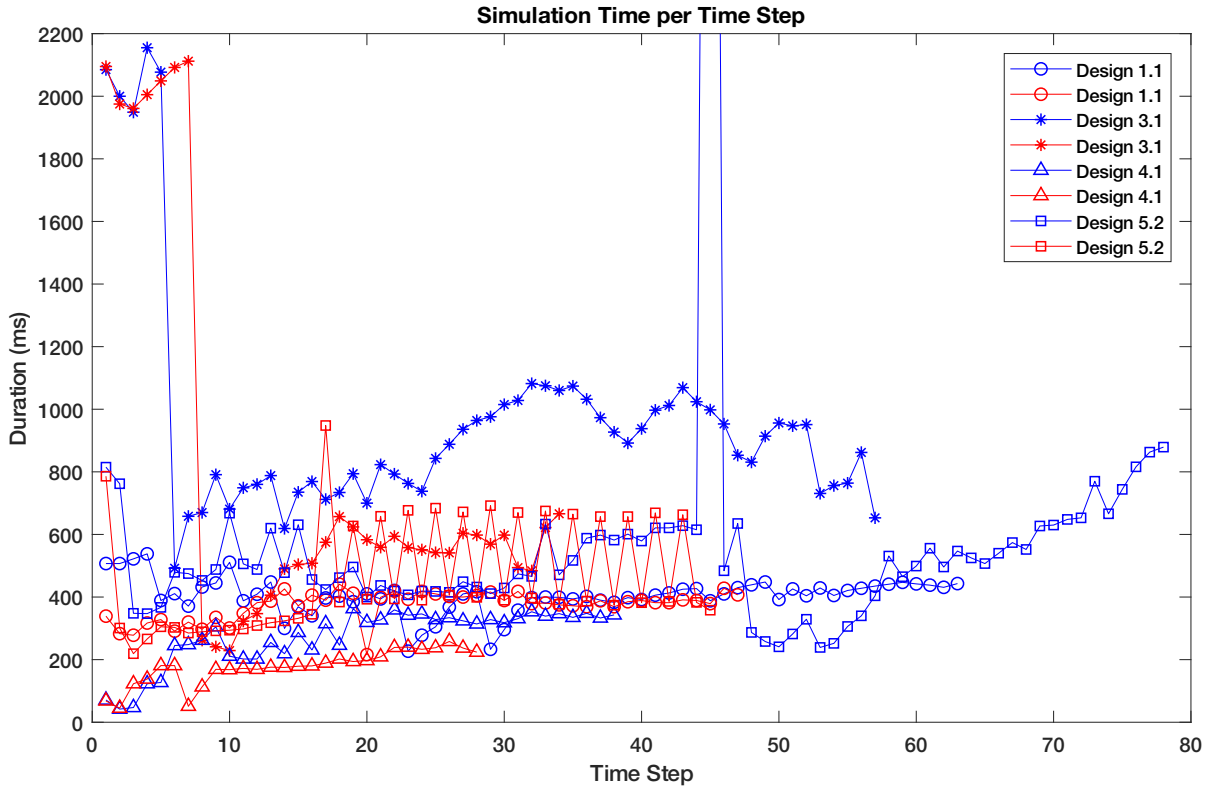
Results

We wish to study the mechanical behaviour of DIMs with an emphasis on the anisotropy they exhibit to deformations as well as the coupling that they possess between different deformation directions. To that effect, we subject a number of different DIM designs to deformations in a direction D while also constraining the relative deformation in the in-plane perpendicular direction D_{perp}. For brevity, we refer to deformation in the main direction D simply as deformation. We expect to see a decrease in deformation as the relative deformation along D_{perp} increases due to reduced shifting and stacking of elements. Similarly, an increase in minor radius should lead to a decrease in deformation since thicker rings have less space to move relative to each other. Conversely, an increase in major radius should allow greater deformations.

6.1. Simulation speed

Our simulations were run on a laptop using an *Intel(R) Core(TM) i7-4720HQ CPU* processor. Optimizing for speed was not our main concern, hence, we did not pursue speed gains other than basic optimizations to avoid unnecessary computations when elements are obviously too far apart to make contact. Parallelizing of the evaluation of intersections and interlocking is likely possible by having multiple threads be responsible for different pairs of elements at the same time.

Refer to Figure 6.1 for time measurements of simulation length, where we pull a DIM in a given direction until we converge to a minimum. Topologies that are more densely packed (e.g., designs 3.1 and 5.2) tend to lead to smaller updates (in terms of magnitude) at each time step, thus requiring more steps for full extension, while each step is itself more expensive to compute due to the higher number of proximate pairs of tori. This also occurs when the sheet is being compressed (i.e., $l_2 \leq 0$) as can be observed in the line plot of Figure 6.1 by the blue lines spanning more time steps than their red counterparts.



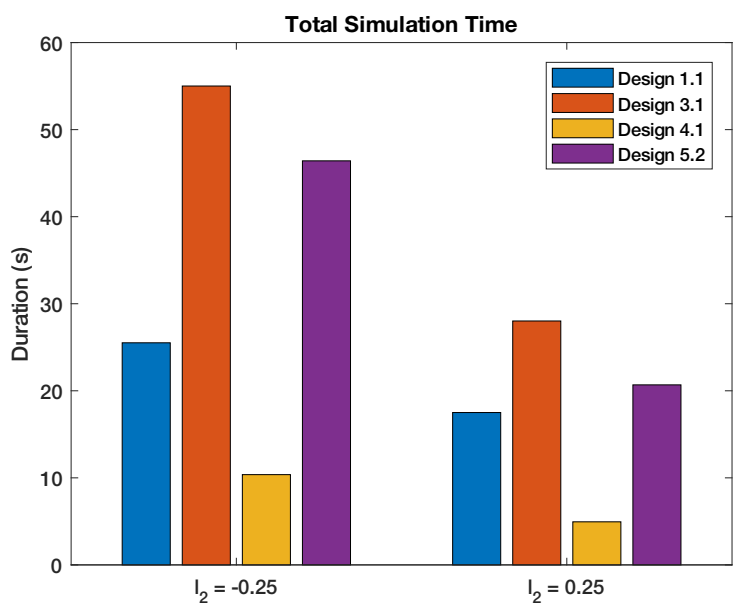


Fig. 6.1. The blue lines are for tests with $l_2 = -0.25$ whereas the red lines are for tests with $l_2 = 0.25$. Some of the designs take slower steps at the beginning where the initial configuration contains intersecting elements for consistency across designs. Note the spike in time for design 5.2 in the line plot, indicating that a step could not be taken and a perturbation of the system had to take place. These performance tests were performed in the same direction for all cases.

6.2. DIM designs

We now present our different designs:

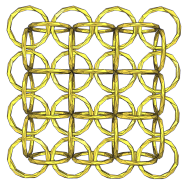
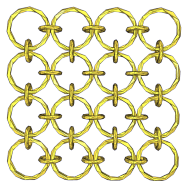
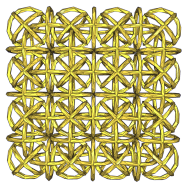
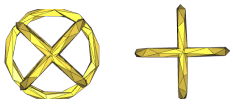
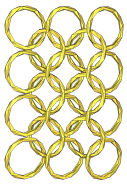
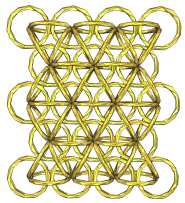
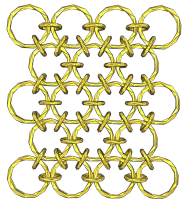
Name	Sheet	Details	Elements
Design 1.1		Ring elements are connected in a cartesian grid fashion. Possesses the same topology as design 1.2.	Single ring elements
Design 1.2		Ring elements are connected in a cartesian grid fashion. Possesses the same topology as design 1.1.	Single ring elements
Design 1.3		MREs are connected in a cartesian grid fashion.	Mixture of 
Design 2		Ring elements are connected diagonally.	Single ring elements
Design 3.1		Ring elements are connected along three directions. Possesses the same topology as design 3.2.	Single ring elements
Design 3.2		Ring elements are connected along three directions. Possesses the same topology as design 3.1.	Single ring elements

Fig. 6.2. Table of our DIM designs (first half)

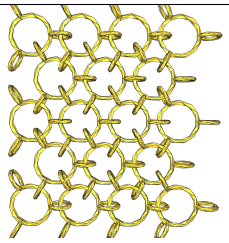
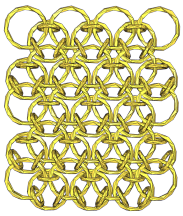
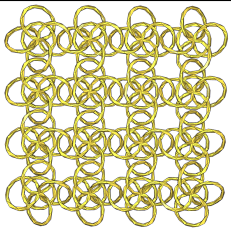
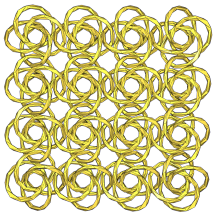
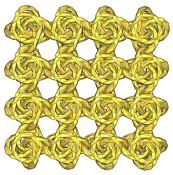
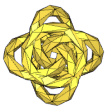
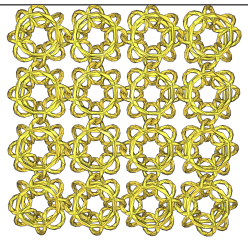
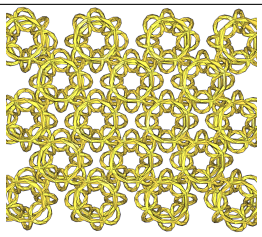
Name	Sheet	Details	Elements
Design 3.3		MREs are connected along three directions.	
Design 3.4		Mixture of single ring elements and MREs connected along three directions.	Single ring elements and 
Design 4.1		MREs are connected in a cartesian grid fashion.	
Design 4.2		MREs are connected in a cartesian grid fashion.	
Design 4.3		MREs are connected in a cartesian grid fashion.	
Design 5.1		MREs are connected in a cartesian grid fashion.	
Design 5.2		MREs are connected diagonally.	

Fig. 6.3. Table of our DIM designs (second half).

6.3. Planar deformations

As described in Section 5, we subject sheets of rigid DIMs to planar deformations and measure the extent to which it can be deformed in a particular direction as described in Section 5.2. By varying design parameters, namely the minor radius of elements and, when sensible, the major radius as well (see designs 1.1 vs 1.2 and designs 3.1 vs 3.2 in Figure 6.2), we explore how they influence the space of possible deformations. Scaling equally both minor and major radii of all elements in a sheet will lead to the exact same deformation as the unscaled version of the sheet as long as relative positions of elements are also scaled accordingly. The deformation measurements reported are obtained by taking $\lambda_1 - 1$ where λ_1 is the largest eigenvalue of C as described in Section 5.2.

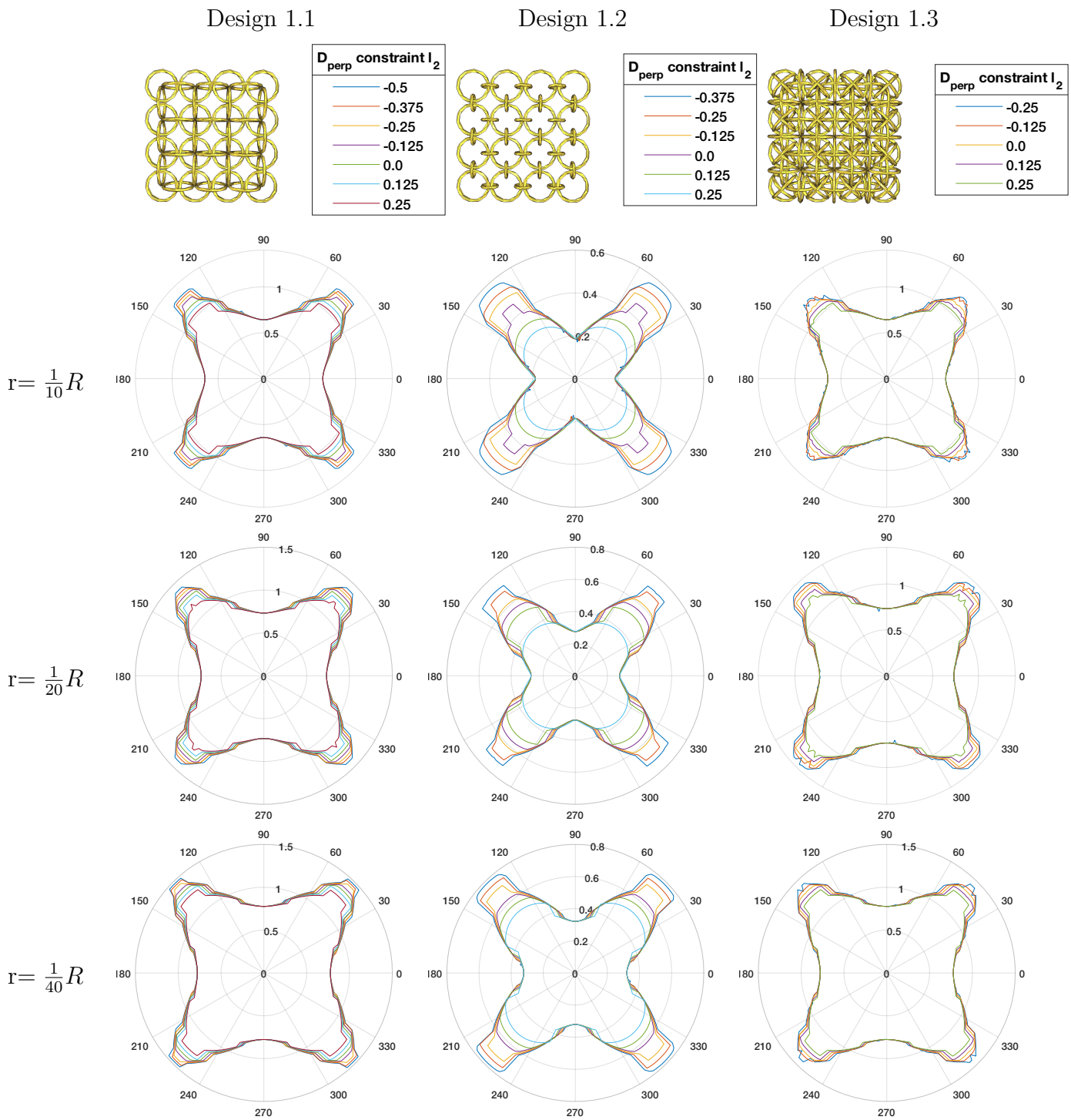


Fig. 6.4. Deformations of designs 1.1, 1.2 and 1.3 in different directions at varying values of l_2.

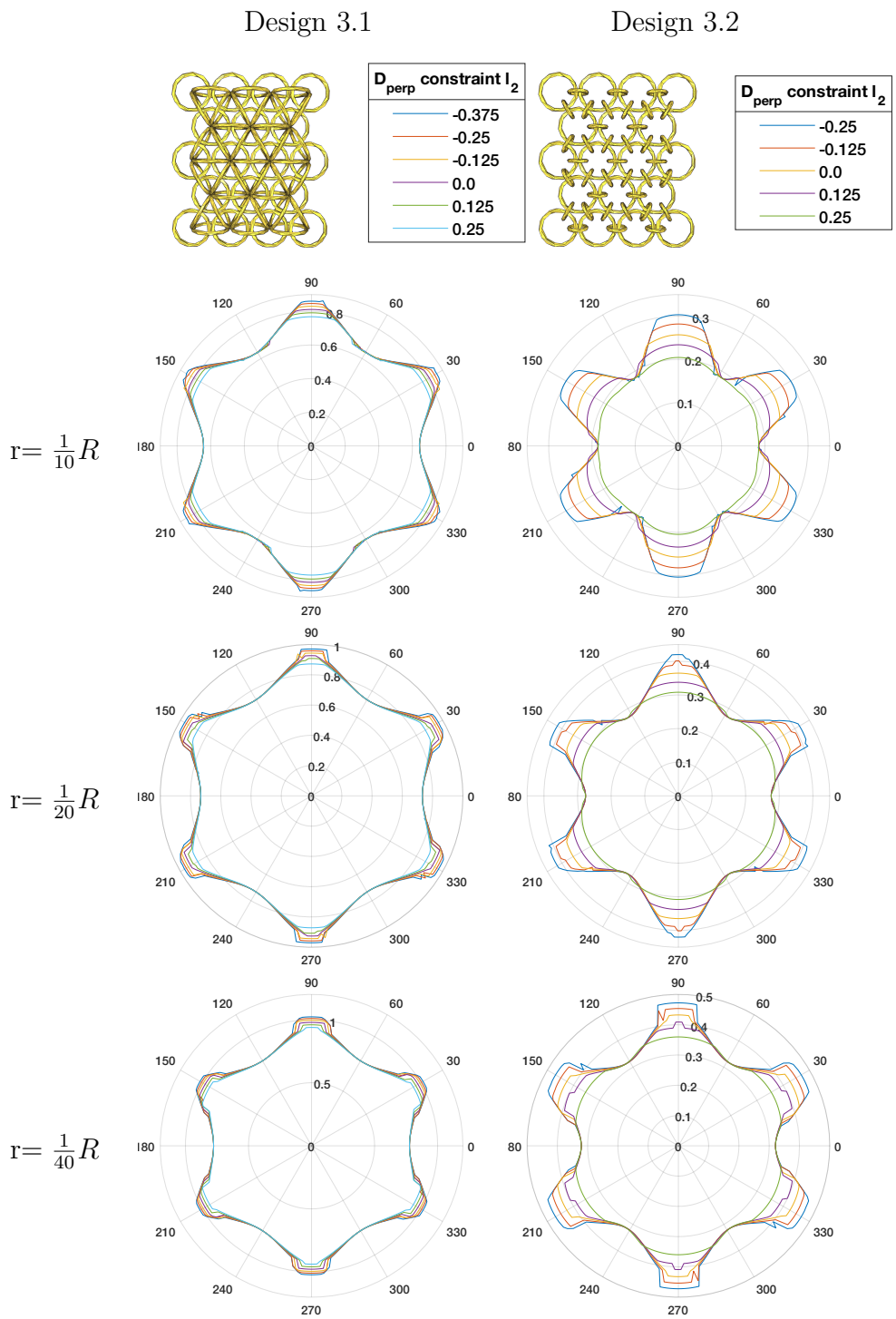


Fig. 6.5. Deformations of designs 3.1 and 3.2 in different directions at varying values of l_2.

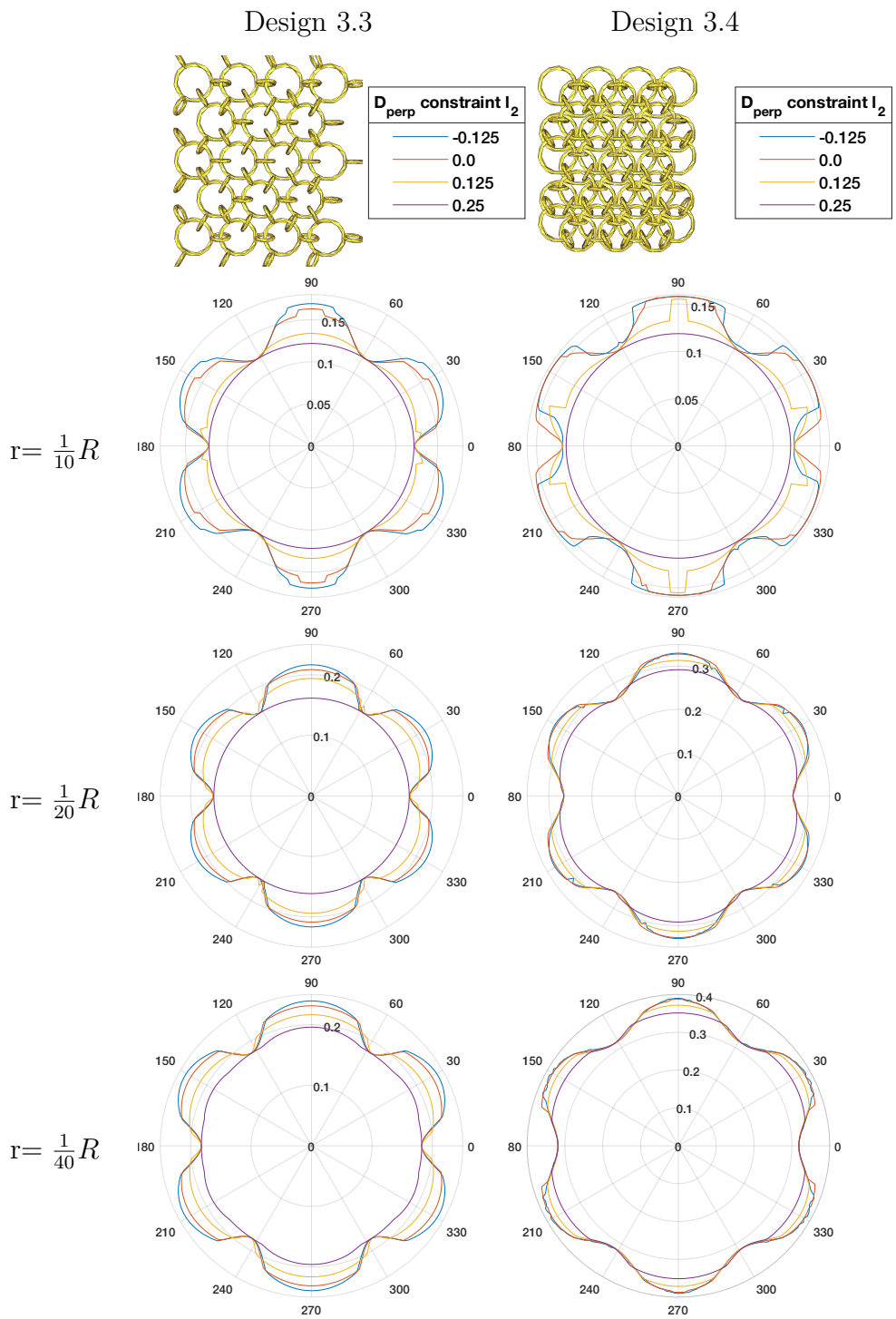


Fig. 6.6. Deformations of designs 3.3 and 3.4 in different directions at varying values of l_2.

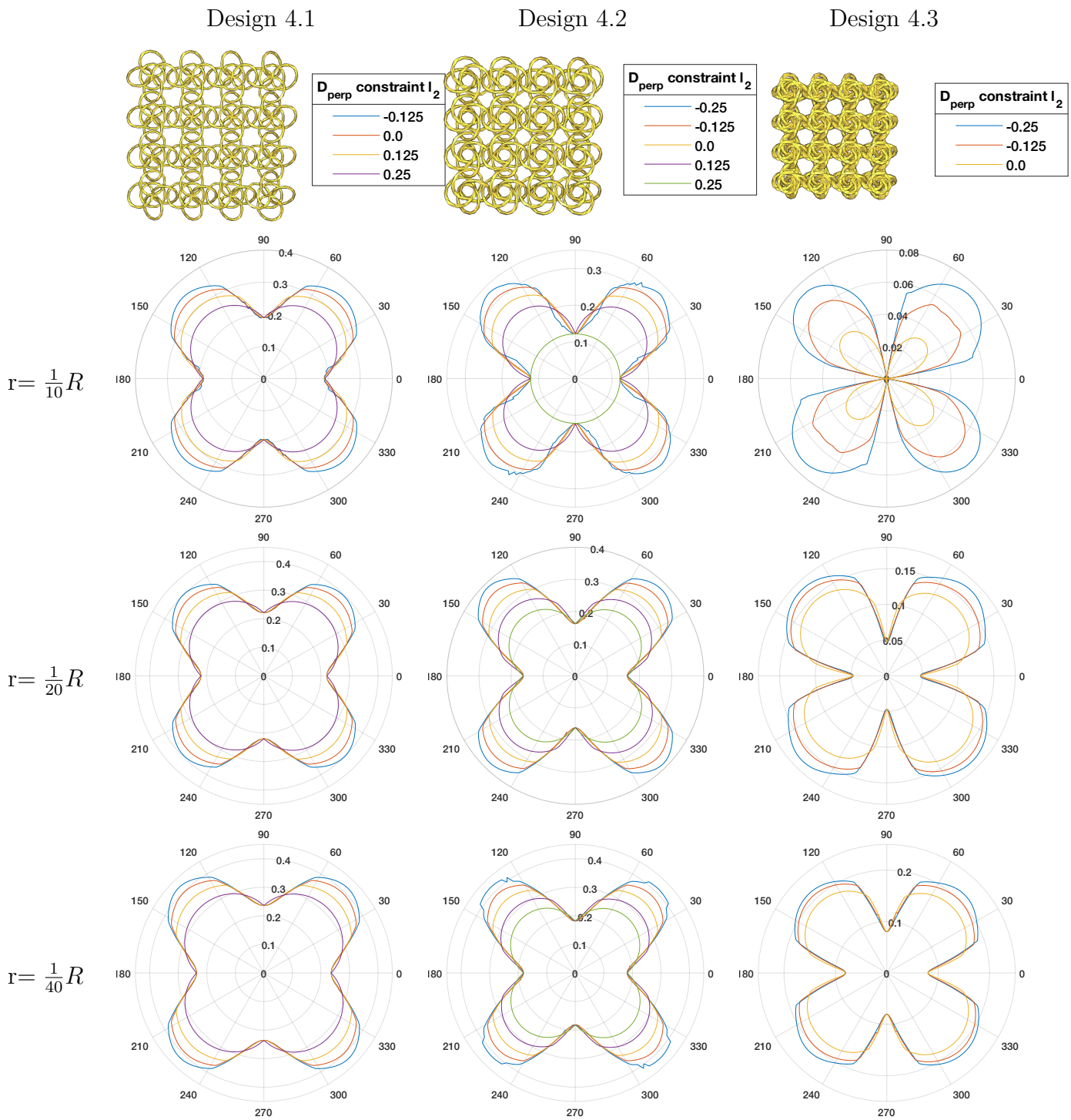


Fig. 6.7. Deformations of designs 4.1, 4.2, and 4.3 in different directions at varying values of l_2.

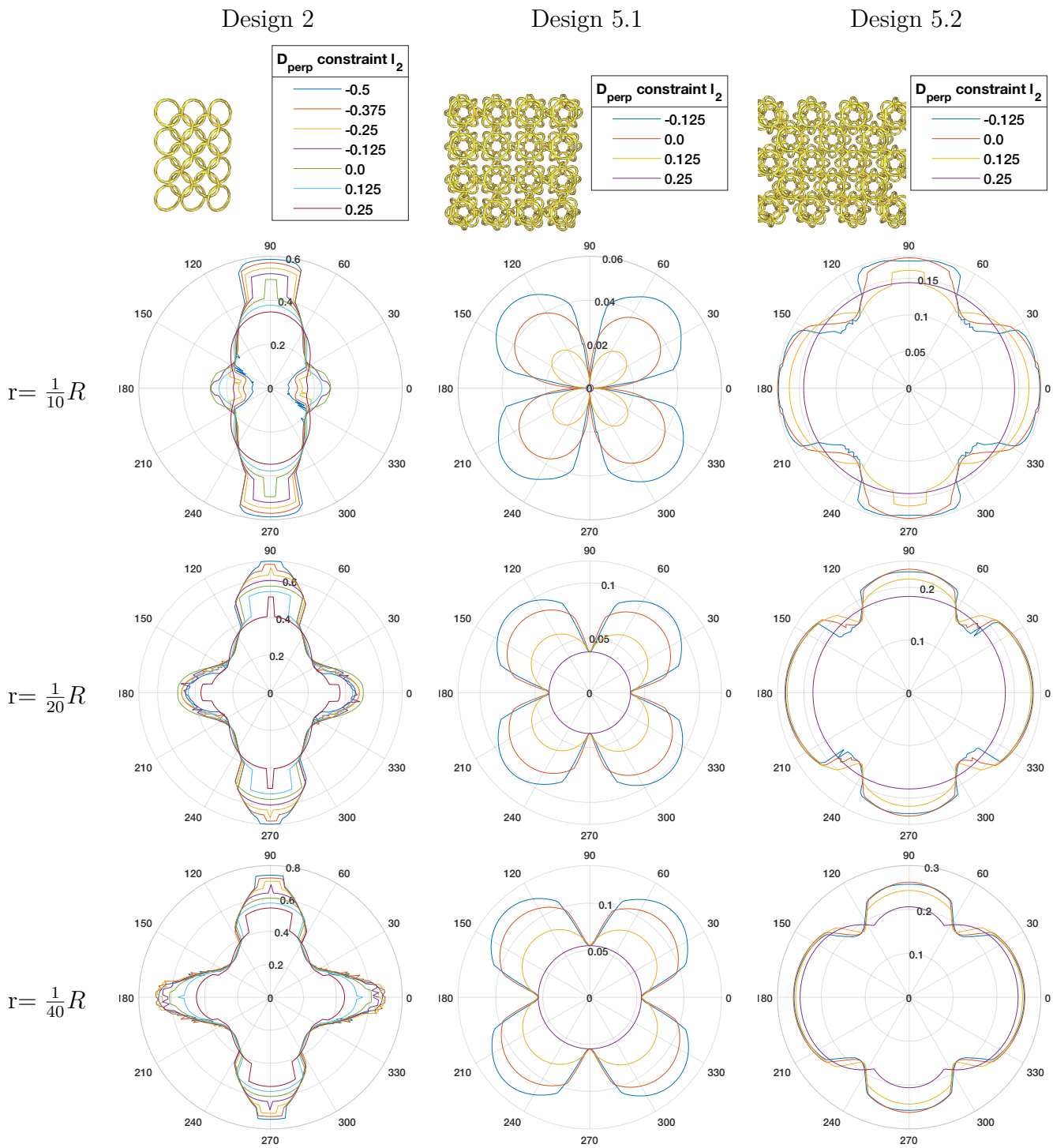


Fig. 6.8. Deformations of designs 2, 5.1, and 5.2 in different directions at varying values of l_2.

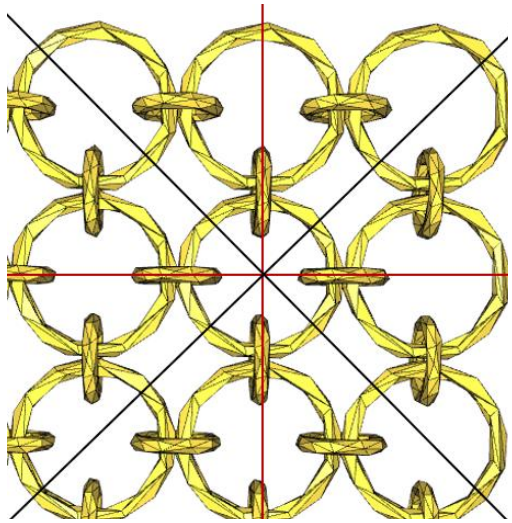


Fig. 6.9. The interlocking directions, described by the red lines, are defined by the direction in which neighbours interlock with each other. These directions usually exhibit minimum deformation. The black lines indicate the directions of maximum deformation and are typically equidistant from the nearest interlocking directions.

From our set of experiments, we can make the following observations:

- Deformation is typically minimal when the deformation direction aligns with the direction in which neighbours align. Conversely, deformation is typically maximal when the deformation direction is furthest away (in terms of angle) from aligning with any interlocking direction (see Figure 6.9). This latter case can be explained by the fact that elements are free to shift in space such as to align with their non-interlocking neighbours thus allowing greater deformation ranges. In the former case though, deformation in the given direction becomes limited by a select subset of the sheet's elements. As they are tensed and come into contact with their neighbours, no amount of shifting from adjacent elements will allow greater deformation.
- Decreasing the minor radius of elements relative to their major radius, unsurprisingly, increases the extent to which a DIM can deform. By being thinner, the elements have more room for displacement before coming into contact with their surrounding elements. This effect is compounded as we deform in directions further and further away from the interlocking direction of the sheet (see previous observation). This compounding effect can be observed by comparing the differences between the minimum and maximum deformations over all directions when we alter the minor radius. Figure 6.10 clearly demonstrates this: a smaller minor radius leads to consistently greater difference between the minimal and maximal deformations, with design 2 being somewhat of an exception. Moreover, it can also be observed by smoother transitions from minimum to maximum deformation as we decrease the minor radius. This is particularly obvious in designs 1.2 and 4.3 where the curve of the

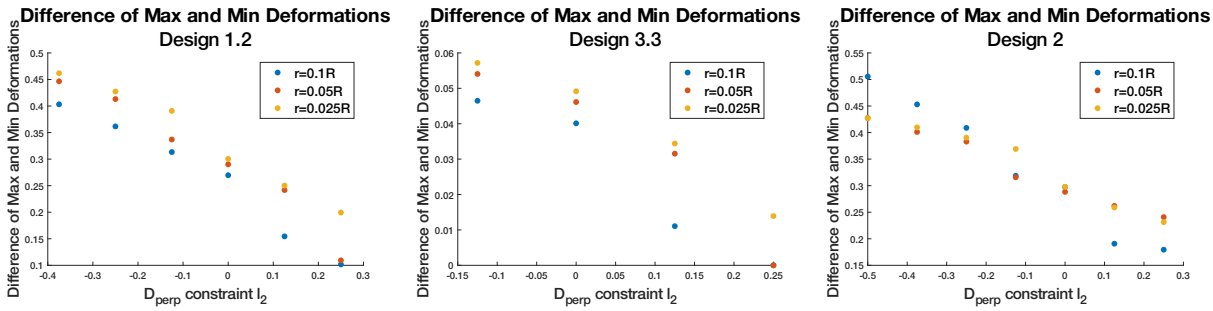


Fig. 6.10. Effect of minor radius on the difference between the maximum and minimum deformations. Design 2 distinguishes itself by a high amount of element stacking at small l_2 values which significantly reduces the minimum deformations observed, leading to larger differences than at smaller relative minor radii.

deformation graphs in directions 0° and 90° are visibly less sharp as the minor radius decreases. Lastly, the minor radius can also play a major role in the possible deformation space by determining whether or not certain elements can align and overlap: as elements become thicker, this phenomenon might become impossible which could lead to a decrease in the displacement range as we see in design 2.

- Compression of the sheet in the direction orthogonal to the deformation direction typically allows greater deformation. This builds upon the first observation: the movement of elements induced by the compression typically is equivalent to the shifting that these elements need to go through to allow a greater deformation. The greater the compression, the more the elements may shift into a more beneficial configuration for deformation. In a few rare cases though, compression will instead limit deformation by excessively stacking rings which then leaves very little room for the elements to move in the desired direction. Design 2 is a perfect example of this (see Figures 6.8 and 6.10) where the deformation in the direction range of $(-15°, 15°)$ becomes much more significant as the minor radius decreases, especially at lower values of l_2.
- Altering the major radius of a subset of elements does not change the core behaviour of an interlocking pattern, i.e. directions of maximum and minimum deformations remain the same. In designs 1.1 and 3.1, we halved the major radius of all elements not aligned with the sheet's plane to obtain designs 1.2 and 3.2 (see Figures 6.4 and 6.5) and observed this. Increasing the major radius will lead to an increase in deformation in all directions. We can observe a trend in the difference between the minimum and maximum deformations observed over all directions: as we increase l_2, this difference decreases at roughly the same rate despite the different element sizes (see Figure 6.11).

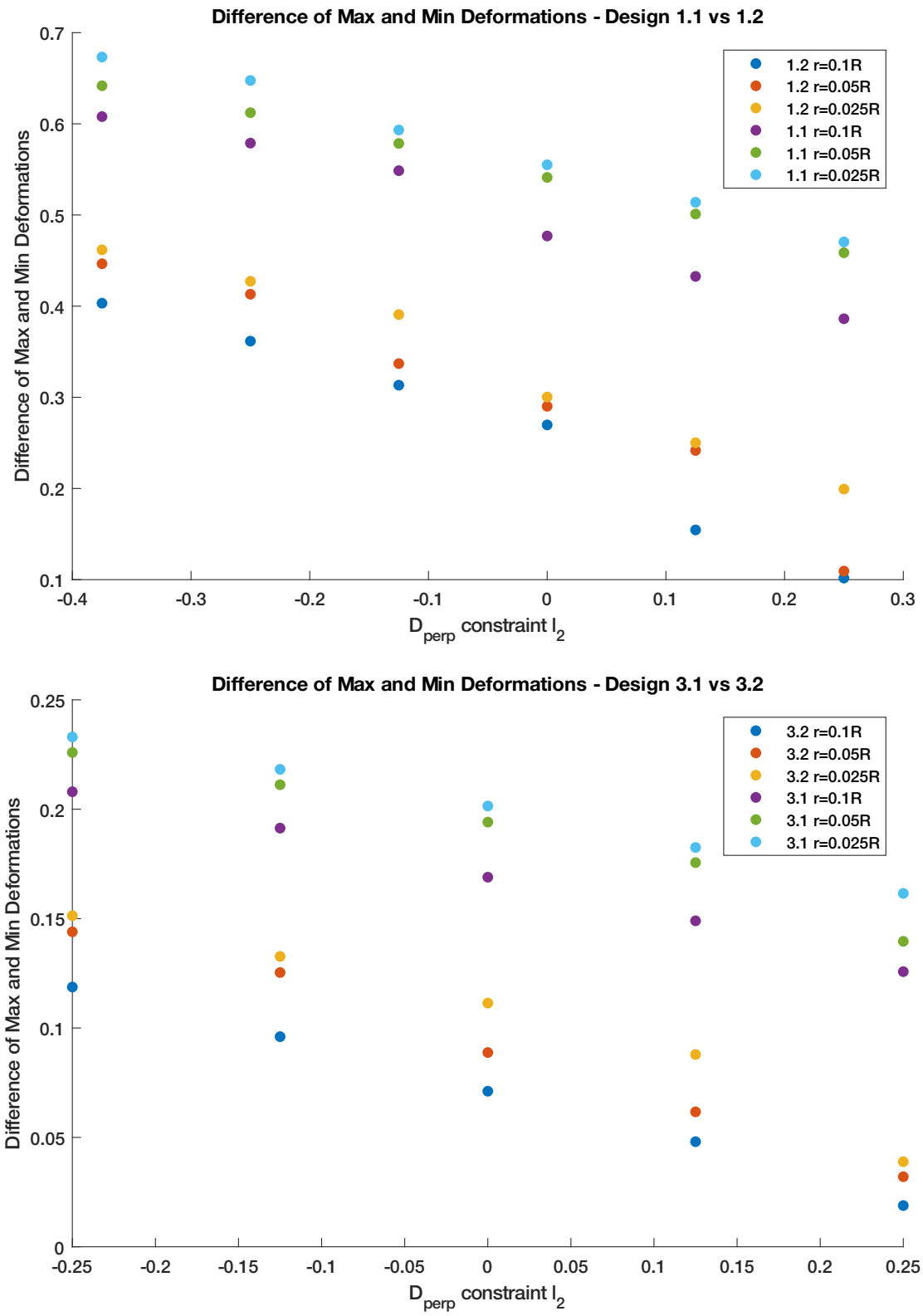


Fig. 6.11. Effect of altering the major radius of a subset of elements in the DIM sheet (see designs 1.1 vs 1.2 and 3.1 vs 3.2). The magnitude of the difference is larger with a larger R, but the decrease of said difference remains roughly the same as l_2 increases.

6.4. Sheet shape

We adopt a unit cell implementation to simulate and to study infinite tilings of DIMs. Contrary to our expectations, the shape of the unit cell and its tiling still influence the sheet's deformation behaviour.

As we can observe in Figure 6.12, the deformation peak in the direction of 30° is much wider than the other peaks in the directions of 90° and 150°. When we arrange the elements in a more rectangular fashion such as in Figures 6.5 and 6.6 however, the designs do not appear to favor any of the three deformation directions over the others. It is unclear to us why this occurs.

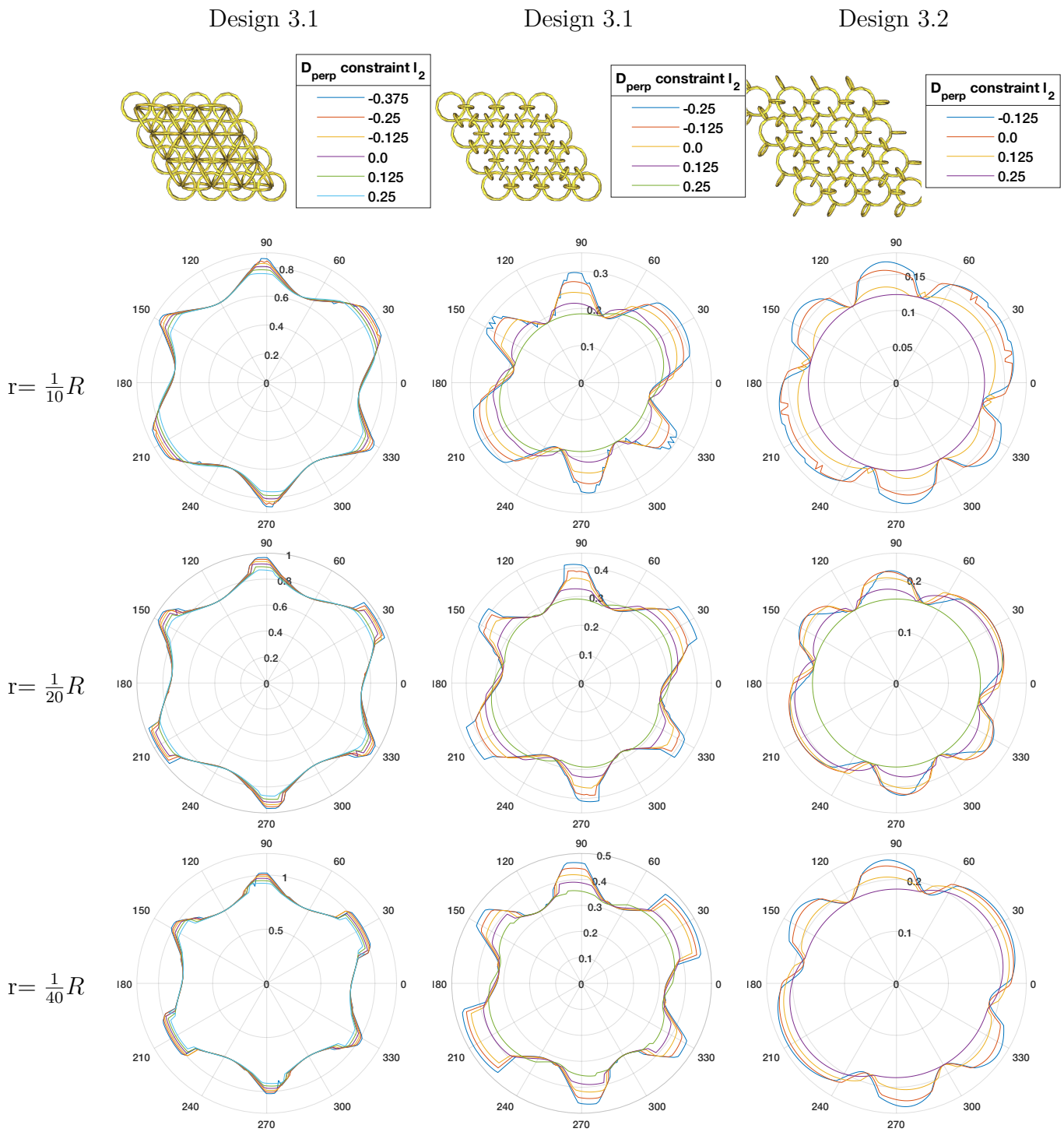


Fig. 6.12. Deformations of designs 3.1, 3.2, and 3.3 in different directions at varying values of l_2 when using a different configuration of elements, but the same topology of interlocking. The deformation peaks in the direction of 30° are much wider than the peaks in the 90° and the 150° directions.

Chapter 7

Conclusion

This thesis represents the first attempt to formally characterize the mechanical behaviour of DIMs, a class of materials that distinguishes itself from others by its biphasic response to deformation. Our work focused on rigid DIMs and the deformations that they can be subjected to. We achieve this through an energy-based method that leverages the geometrical properties of the torus to minimize overlap between elements of the DIM and maintain the neighbour relationship of pairs of elements at every time step. We show that the distance between two circles in three dimensional space can be used to determine whether or not two tori overlap. Through a similar set of operations, we also show how to recognize whether or not two tori are interconnected. The method thus works by using an energy to induce deformation while penalizing for intersection between elements and adjusting step sizes to maintain the state of interconnectedness at every time step. The use of a unit cell implementation allows us to study the behaviour of DIMs in a manner that is independent from the simulated sample's size.

The lack of a manner in which to objectively describe the rest state of a DIM is a major limitation: it makes it very hard to compare the behaviour between different DIMs while also making it challenging to do physical experiments to validate our simulations' results.

Otherwise, the use of a score function specifically built for tori is certainly a limitation, however, it should be possible to extend it to other shapes as well. For instance, our score function could easily be adapted for a rod-like shape composed of a cylinder with a half-sphere at both bases: this shape could then be used to create a much greater variety of elements than is possible with just tori.

Lastly, our method is also limited by the fact that, like most minimization routines, it can find a set of parameter values that are locally minimal, but not globally minimal. From our experiments, it seems that such minima are obtained very rarely, but they do occur as can be seen by the jittery lines of some graphs among our results.

A natural extension of this work would be to study out-of-plane deformations on planar arrangements of DIMs and deformations in 3D arrangements of DIMs rather than planar ones. Conversely, looking into elastic DIMs with the goal of better understanding the elastic regime that DIMs can experience (which this work ignores) is also an interesting avenue. An important problem to solve would be the inverse task of deriving a DIM from a specific deformation behaviour. Because of their discrete nature, this would be a challenging endeavour, but it would likewise make finding new and interesting applications for DIMs much easier, which is ultimately our goal.

References

[1] Erik Andreassen, Boyan S Lazarov et Ole Sigmund : Design of manufacturable 3d extremal elastic microstructure. *Mechanics of Materials*, 69(1):1–10, 2014.

[2] David Baraff et Andrew Witkin : Large steps in cloth simulation. *In Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998.

[3] Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross et Wojciech Matusik : Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010.

[4] Michele Caraglio, Cristian Micheletti et Enzo Orlandini : Mechanical pulling of linked ring polymers: Elastic response and link localisation. *Polymers*, 9(8):327, 2017.

[5] Desai Chen, David IW Levin, Shinjiro Sueda et Wojciech Matusik : Data-driven finite elements for geometry and material design. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

[6] Gabriel Cirio, Jorge Lopez-Moreno et Miguel A Otaduy : Yarn-level cloth simulation with sliding persistent contacts. *IEEE transactions on visualization and computer graphics*, 23(2):1152–1162, 2016.

[7] Eulalie Coevoet, Otman Benchekroun et Paul G Kry : Adaptive merging for rigid body simulation. *ACM Transactions on Graphics (TOG)*, 39(4):35–1, 2020.

[8] Bailin Deng, Sofien Bouaziz, Mario Deuss, Alexandre Kaspar, Yuliy Schwartzburg et Mark Pauly : Interactive design exploration for constrained meshes. *Computer-Aided Design*, 61:13–23, 2015.

[9] Evan Drumwright : A fast and stable penalty method for rigid body simulation. *IEEE transactions on visualization and computer graphics*, 14(1):231–240, 2007.

[10] Simon Duenser, Roi Poranne, Bernhard Thomaszewski et Stelian Coros : Robocut: hot-wire cutting with robot-controlled flexible rods. *ACM Transactions on Graphics (TOG)*, 39(4):98–1, 2020.

[11] D. Eberly : Distance to circles in 3d. `https://www.geometrictools.com/Documentation/DistanceToCircle3.pdf`. Accessed: 2020-03-10.

[12] Jonathan Engel et Chang Liu : Creation of a metallic micromachined chain mail fabric. *Journal of Micromechanics and Microengineering*, 17(3):551, 2007.

[13] Agustin Flowalistik : Chainmail - 3d printable fabric. `https://www.thingiverse.com/thing:3096598`. Accessed: 2021-07-17.

[14] Marc GD Geers, Erica WC Coenen et Varvara G Kouznetsova : Multi-scale computational homogenization of structured thin sheets. *Modelling and Simulation in Materials Science and Engineering*, 15(4):S393, 2007.

[15] Shu-Wei Hsu et John Keyser : Piles of objects. *ACM Transactions on Graphics (TOG)*, 29(6):1–6, 2010.

[16] Jonathan M Kaldor, Doug L James et Steve Marschner : Simulating knitted cloth at the yarn level. *In ACM SIGGRAPH 2008 papers*, pages 1–9. 2008.

[17] Samuel Lensgraf, Karim Itani, Yinan Zhang, Zezhou Sun, Yijia Wu, Alberto Quattrini Li, Bo Zhu, Emily Whiting, Weifu Wang et Devin Balkcom : Puzzleflex: kinematic motion of chains with loose joints. *In 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6730–6737. IEEE, 2020.

[18] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang et Danny M Kaufman : Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics. *ACM transactions on graphics*, 2020.

[19] Qiang Lu et Rui Huang : Nonlinear mechanics of single-atomic-layer graphene sheets. *International Journal of Applied Mechanics*, 1(03):443–467, 2009.

[20] Luigi Malomo, Nico Pietroni, Bernd Bickel et Paolo Cignoni : Flexmolds: Automatic design of flexible shells for molding. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

[21] Mohammad Mirkhalaf, Tao Zhou et Francois Barthelat : Simultaneous improvements of strength and toughness in topologically interlocked ceramics. *Proceedings of the National Academy of Sciences*, 115(37):9128–9133, 2018.

[22] Julian Panetta, Abtin Rahimian et Denis Zorin : Worst-case stress relief for microstructures. *ACM Transactions on Graphics (TOG)*, 36(4):1–16, 2017.

[23] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni et Denis Zorin : Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[24] Danit Peleg : 3d printed fashion. `https://danitpeleg.com/`. Accessed: 2021-04-03.

[25] Jesús Pérez, Miguel A Otaduy et Bernhard Thomaszewski : Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[26] Ante Qu et Doug L. James : Fast linking numbers for topology verification of loopy structures. 2021.

[27] Mark Ransley, Peter Smitham et Mark Miodownik : Active chainmail fabrics for soft robotic applications. *Smart Materials and Structures*, 26(8):08LT02, 2017.

[28] Jessica Rosenkrantz et Jesse Louis-Rosenberg : Nervous system: about us. `https://n-e-r-v-o-u-s.com/about_us.php`. Accessed: 2021-04-03.

[29] Harald Jurgen Schmidl : *Optimization-based animation*. University of Miami, 2002.

[30] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio et Markus Gross : Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

[31] Christian Schumacher, Steve Marschner, Markus Gross et Bernhard Thomaszewski : Mechanical characterization of structured sheet materials. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[32] Mélina Skouras, Stelian Coros, Eitan Grinspun et Bernhard Thomaszewski : Interactive surface design with interlocking elements. *ACM Transactions on Graphics (TOG)*, 34(6):1–7, 2015.

[33] Nobuyuki Umetani, Takeo Igarashi et Niloy J Mitra : Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.*, 31(4):86–1, 2012.

[34] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi et Eitan Grinspun : Sensitive couture for interactive garment modeling and editing. *ACM Transactions on Graphics*, 30(4):90, 2011.

[35] Paras Wadekar, Eric Markowicz, Dani Liu, Genevieve Dion, Antonios Kontsos et David Breen : An optimized yarn-level geometric model for fea simulation of weft-knitted fabrics.

[36] Ziqi Wang, Peng Song, Florin Isvoranu et Mark Pauly : Design and structural optimization of topological interlocking assemblies. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[37] Ziqi Wang, Peng Song et Mark Pauly : Desia: A general framework for designing interlocking assemblies. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[38] Yong-Liang Yang, Yi-Jun Yang, Helmut Pottmann et Niloy J Mitra : Shape space exploration of constrained meshes. *ACM Trans. Graph.*, 30(6):124, 2011.

[39] Xiaoting Zhang, Guoxin Fang, Mélina Skouras, Gwenda Gieseler, Charlie Wang et Emily Whiting : Computational design of fabric formwork. *ACM Transactions on Graphics*, 38(4):1–13, 2019.

Appendix A

Circle-to-circle distance

We solve for θ in the following way. As a reminder:

$$g = \frac{d}{d\theta}|P-K|^2 = 2(N_1 \cdot \Delta(\theta))(N_1 \cdot \Delta'(\theta)) + 2(|N_1 \times \Delta(\theta)| - R_1)\frac{N_1 \times \Delta(\theta) \cdot N_1 \times \Delta'(\theta)}{|N_1 \times \Delta(\theta)|}$$

$$= \Delta(\theta) \cdot \Delta'(\theta) - R_1 \frac{N_1 \times \Delta(\theta) \cdot N_1 \times \Delta'(\theta)}{|N_1 \times \Delta(\theta)|} = 0.$$

From g, we can define the following

$$H(\theta) = |N_1 \times \Delta(\theta)|^2 \Big(\Delta(\theta) \cdot \Delta'(\theta)\Big)^2 - R_1^2 \Big(N_1 \times \Delta(\theta) \cdot N_1 \times \Delta'(\theta)\Big)^2 = 0.$$

Letting $\gamma = \cos\theta$ and $\sigma = \sin\theta$, we obtain the following:

$$H(\gamma, \sigma) = \Big(p_0(\gamma) + \sigma p_1(\gamma)\Big)\Big(p_2(\gamma) + \sigma p_3(\gamma)\Big)^2 - R_1^2\Big(p_4(\gamma) + \sigma p_5(\gamma)\Big)^2 = p_6(\gamma) + \sigma p_7(\gamma)$$

where:

$$\begin{aligned}
a_0 &= R_2(c_2 - c_1) \cdot U \\
a_1 &= R_2(c_2 - c_1) \cdot V \\
a_2 &= |N_1 \times (c_2 - c_1)|^2 \\
a_3 &= R_2 N_1 \times (c_2 - c_1) \cdot N_1 \times U \\
a_4 &= R_2 N_1 \times (c_2 - c_1) \cdot N_1 \times V \\
a_5 &= R_2^2 |N_1 \times U|^2 \\
a_6 &= R_2^2 N_1 \times U \cdot N_1 \times V \\
a_7 &= R_2^2 |N_1 \times V|^2
\end{aligned}$$

$$
\begin{aligned}
p_0 &= (a_2 + a_7) + 2a_3\gamma + (a_5 - a_7)\gamma^2 \\
p_1 &= 2(a_4 + a_6\gamma) \\
p_2 &= a_1\gamma \\
p_3 &= -a_0 \\
p_4 &= -a_6 + a_4\gamma + 2a_6\gamma^2 \\
p_5 &= -a_3 + (a_7 - a_5)\gamma \\
p_6 &= p_0\Big(p_2^2 + (1-\gamma^2)p_3^2\Big) + (1-\gamma^2)(2p_1p_2p_3) - R_1^2\Big(p_4^2 + (1-\gamma^2)p_5^2\Big) \\
p_7 &= 2p_0p_2p_3 + p_1\Big(p_2^2 + (1-\gamma^2)p_3^2\Big) - R_1^2(2p_4p_5)
\end{aligned}
$$

Lastly, we solve for $p_6 = -\sigma p_7$ and rearrange the terms to obtain

$$\phi(\gamma) = p_6^2 - (1-\gamma^2)p_7^2 = 0.$$

The roots of $\phi(\gamma)$ are then used to find a series of points on C_2. For each point, we compute its distance to C_1 using equation 3.1.2. These distances can then be used to compute energy contributions to the system of elements. See [**11**] for a more in depth explanation of the derivations.

Appendix B

Gradient of the torus-to-torus distance

B.1. Distance gradient

Here we cover the derivation of the different components needed to compute the term $\frac{\partial D}{\partial q}$ of equation 4.3.3.

$$D = d_c^2 = |P_1 - P_2|^2 = (P_1 - P_2)_x^2 + (P_1 - P_2)_y^2 + (P_1 - P_2)_z^2 = D_x^2 + D_y^2 + D_z^2$$

$$\frac{\partial D}{\partial c_1} = 2D_x \frac{\partial D_x}{\partial c_1} + 2D_y \frac{\partial D_y}{\partial c_1} + 2D_z \frac{\partial D_z}{\partial c_1}$$

$$\frac{\partial D_x}{\partial c_1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \qquad \frac{\partial D_y}{\partial c_1} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \qquad \frac{\partial D_z}{\partial c_1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$\frac{\partial D}{\partial \theta_1} = \begin{bmatrix} \frac{\partial D}{\partial \theta_{1x}} & \frac{\partial D}{\partial \theta_{1y}} & \frac{\partial D}{\partial \theta_{1z}} \end{bmatrix}^T$$

$$\frac{\partial D}{\partial \theta_{1x}} = (P_1 - P_2) \cdot (R_{1y} \frac{\partial}{\partial \theta_{1x}} R_{1x} R_{1z} p_{1c})$$

$$\frac{\partial D}{\partial \theta_{1y}} = (P_1 - P_2) \cdot (\frac{\partial}{\partial \theta_{1y}} R_{1y} R_{1x} R_{1z} p_{1c})$$

$$\frac{\partial D}{\partial \theta_{1z}} = (P_1 - P_2) \cdot (R_{1y} R_{1x} \frac{\partial}{\partial \theta_{1z}} R_{1z} p_{1c})$$

$$\frac{\partial D}{\partial c_2} = 2D_x \frac{\partial D_x}{\partial c_2} + 2D_y \frac{\partial D_y}{\partial c_2} + 2D_z \frac{\partial D_z}{\partial c_2}$$

$$\frac{\partial D_x}{\partial c_2} = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T \qquad \frac{\partial D_y}{\partial c_2} = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T \qquad \frac{\partial D_z}{\partial c_2} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$$

$$\frac{\partial D}{\partial \theta_2} = \begin{bmatrix} \frac{\partial D}{\partial \theta_{1x}} & \frac{\partial D}{\partial \theta_{1y}} & \frac{\partial D}{\partial \theta_{1z}} \end{bmatrix}^T$$

$$\frac{\partial D}{\partial \theta_{2x}} = -(P_1 - P_2) \cdot (R_{2y} \frac{\partial}{\partial \theta_{2x}} R_{2x} R_{2z} p_{2c})$$

$$\frac{\partial D}{\partial \theta_{2y}} = -(P_1 - P_2) \cdot (\frac{\partial}{\partial \theta_{2y}} R_{2y} R_{2x} R_{2z} p_{2c})$$
$$\frac{\partial D}{\partial \theta_{2z}} = -(P_1 - P_2) \cdot (R_{2y} R_{2x} \frac{\partial}{\partial \theta_{2z}} R_{2z} p_{2c})$$

B.2. Sensitivity analysis

Here we cover the derivation of the values necessary for sensitivity analysis to obtain the term $\frac{\partial \theta}{\partial q}$ in equation 4.3.5. Note that R_a is a scalar (major radius of T_a) whereas R_{ab} is a rotation matrix applied to T_a about axis $b \in \{x,y,z\}$. Likewise, θ is the angle at which a point is evaluated on C_2 whereas $\theta_a = \begin{bmatrix}\theta_{ax} & \theta_{ay} & \theta_{az}\end{bmatrix}^T$ is a vector of rotation angles applied to T_a.

$$\Delta(\theta) = c_2 - c_1 + R_2\Big(\cos\theta U + \sin\theta V\Big)$$
$$\Delta'(\theta) = R_2\Big(-\sin\theta U + \cos\theta V\Big)$$
$$g = \frac{\partial D}{\partial \theta} = 2(N_1 \cdot \Delta(\theta))(N_1 \cdot \Delta'(\theta)) + 2(|N_1 \times \Delta(\theta)| - R_1)\frac{N_1 \times \Delta(\theta) \cdot N_1 \times \Delta'(\theta)}{|N_1 \times \Delta(\theta)|}$$

$$\begin{aligned}\frac{\partial g}{\partial \theta} =& 2(N_1 \cdot \Delta''(\theta))(N_1 \cdot \Delta(\theta)) + 2(N_1 \cdot \Delta'(\theta))(N_1 \cdot \Delta'(\theta)) \\ &+ 2((N_1 \times \Delta(\theta)) \cdot N_1 \times \Delta'(\theta))\frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|^2} \\ &+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(N_1 \times \Delta''(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\ &+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta'(\theta)}{|N_1 \times \Delta(\theta)|} \\ &- 2(|N_1 \times \Delta(\theta)| - R_1)\frac{[(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)][(N_1 \times \Delta(\theta)) \cdot (N_1 \times \Delta'(\theta))]}{|N_1 \times \Delta(\theta)|^3}\end{aligned}$$

$$\frac{\partial \Delta(\theta)}{\partial c_{1x}} = \begin{bmatrix}-1 & 0 & 0\end{bmatrix}^T \qquad \frac{\partial \Delta(\theta)}{\partial c_{1y}} = \begin{bmatrix}0 & -1 & 0\end{bmatrix}^T \qquad \frac{\partial \Delta(\theta)}{\partial c_{1z}} = \begin{bmatrix}0 & 0 & -1\end{bmatrix}^T$$
$$\frac{\partial \Delta(\theta)}{\partial c_{2x}} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T \qquad \frac{\partial \Delta(\theta)}{\partial c_{2y}} = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T \qquad \frac{\partial \Delta(\theta)}{\partial c_{2z}} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$$

Considering that $U = R_{2y}R_{2x}R_{2z}U_c$, $V = R_{2y}R_{2x}R_{2z}V_c$, $N_1 = R_{1y}R_{1x}R_{1z}N_{1c}$:

$$\frac{\partial \Delta(\theta)}{\partial \theta_2} = \begin{bmatrix}\frac{\partial \Delta(\theta)}{\partial \theta_{2x}} & \frac{\partial \Delta(\theta)}{\partial \theta_{2y}} & \frac{\partial \Delta(\theta)}{\partial \theta_{2z}}\end{bmatrix}^T$$
$$\frac{\partial \Delta(\theta)}{\partial \theta_{2x}} = R_2 \cdot (R_{2y}\frac{\partial}{\partial \theta_{2x}} R_{2x}R_{2z})\Big(\cos\theta U_c + \sin\theta V_c\Big)$$
$$\frac{\partial \Delta(\theta)}{\partial \theta_{2y}} = R_2 \cdot (\frac{\partial}{\partial \theta_{2y}} R_{2y}R_{2x}R_{2z})\Big(\cos\theta U_c + \sin\theta V_c\Big)$$

$$\frac{\partial \Delta(\theta)}{\partial \theta_{2z}} = R_2 \cdot (R_{2y} R_{2x} \frac{\partial}{\partial \theta_{2z}} R_{2z}) \Big(\cos\theta U_c + \sin\theta V_c \Big)$$

$$\frac{\partial \Delta'(\theta)}{\partial \theta_2} = \begin{bmatrix} \frac{\partial \Delta'(\theta)}{\partial \theta_{2x}} & \frac{\partial \Delta'(\theta)}{\partial \theta_{2y}} & \frac{\partial \Delta'(\theta)}{\partial \theta_{2z}} \end{bmatrix}^T$$

$$\frac{\partial \Delta'(\theta)}{\partial \theta_{2x}} = R_2 \cdot (R_{2y} \frac{\partial}{\partial \theta_{2x}} R_{2x} R_{2z}) \Big(-\sin\theta U_c + \cos\theta V_c \Big)$$

$$\frac{\partial \Delta'(\theta)}{\partial \theta_{2y}} = R_2 \cdot (\frac{\partial}{\partial \theta_{2y}} R_{2y} R_{2x} R_{2z}) \Big(-\sin\theta U_c + \cos\theta V_c \Big)$$

$$\frac{\partial \Delta'(\theta)}{\partial \theta_{2z}} = R_2 \cdot (R_{2y} R_{2x} \frac{\partial}{\partial \theta_{2z}} R_{2z}) \Big(-\sin\theta U_c + \cos\theta V_c \Big)$$

$$\frac{\partial N_1}{\partial \theta_1} = \begin{bmatrix} \frac{\partial N_1}{\partial \theta_{1x}} & \frac{\partial N_1}{\partial \theta_{1y}} & \frac{\partial N_1}{\partial \theta_{1z}} \end{bmatrix}^T$$

$$\frac{\partial N_1}{\partial \theta_{1x}} = (R_{1y} \frac{\partial}{\partial \theta_{1x}} R_{1x} R_{1z}) N_{1c}$$

$$\frac{\partial N_1}{\partial \theta_{1y}} = (\frac{\partial}{\partial \theta_{1y}} R_{1y} R_{1x} R_{1z}) N_{1c}$$

$$\frac{\partial N_1}{\partial \theta_{1z}} = (R_{1y} R_{1x} \frac{\partial}{\partial \theta_{1z}} R_{1z}) N_{1c}$$

$$\begin{aligned}
\frac{\partial g}{\partial c_1} =& 2(N_1 \cdot \Delta'(\theta))(N_1 \cdot \frac{\partial}{\partial c_1} \Delta(\theta)) + 2((N_1 \times \Delta(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_1} \Delta(\theta)) \frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|^2} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1) \frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_1} \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&- 2(|N_1 \times \Delta(\theta)| - R_1) \frac{[(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)][(N_1 \times \Delta(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_1} \Delta(\theta)]}{|N_1 \times \Delta(\theta)|^3}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial g}{\partial c_2} =& 2(N_1 \cdot \Delta'(\theta))(N_1 \cdot \frac{\partial}{\partial c_2} \Delta(\theta)) + 2((N_1 \times \Delta(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_2} \Delta(\theta)) \frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|^2} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1) \frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_2} \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&- 2(|N_1 \times \Delta(\theta)| - R_1) \frac{[(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)][(N_1 \times \Delta(\theta)) \cdot N_1 \times \frac{\partial}{\partial c_2} \Delta(\theta)]}{|N_1 \times \Delta(\theta)|^3}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial g}{\partial \theta_2} =& 2(N_1 \cdot \frac{\partial}{\partial \theta_2}\Delta'(\theta))(N_1 \cdot \Delta(\theta)) + 2(N_1 \cdot \Delta'(\theta))(N_1 \cdot \frac{\partial}{\partial \theta_2}\Delta(\theta)) \\
&+ 2((N_1 \times \Delta(\theta)) \cdot (N_1 \times \frac{\partial}{\partial \theta_2}\Delta(\theta)))\frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|^2} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(N_1 \times \frac{\partial}{\partial \theta_2}\Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \frac{\partial}{\partial \theta_2}\Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&- 2(|N_1 \times \Delta(\theta)| - R_1)\frac{[(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)][(N_1 \times \frac{\partial}{\partial \theta_2}\Delta(\theta)) \cdot N_1 \times \Delta(\theta)]}{|N_1 \times \Delta(\theta)|^3}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial g}{\partial \theta_1} =& 2(\frac{\partial}{\partial \theta_1}N_1 \cdot \Delta'(\theta))(N_1 \cdot \Delta(\theta)) + 2(N_1 \cdot \Delta'(\theta))(\frac{\partial}{\partial \theta_1}N_1 \cdot \Delta(\theta)) \\
&+ 2((N_1 \times \Delta(\theta)) \cdot (\frac{\partial}{\partial \theta_1}N_1 \times \Delta(\theta)))\frac{(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|^2} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(\frac{\partial}{\partial \theta_1}N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&+ 2(|N_1 \times \Delta(\theta)| - R_1)\frac{(N_1 \times \Delta'(\theta)) \cdot \frac{\partial}{\partial \theta_1}N_1 \times \Delta(\theta)}{|N_1 \times \Delta(\theta)|} \\
&- 2(|N_1 \times \Delta(\theta)| - R_1)\frac{[(N_1 \times \Delta'(\theta)) \cdot N_1 \times \Delta(\theta)][(\frac{\partial}{\partial \theta_1}N_1 \times \Delta(\theta)) \cdot N_1 \times \Delta(\theta)]}{|N_1 \times \Delta(\theta)|^3}
\end{aligned}
$$

We can now evaluate $\frac{\partial \theta}{\partial q} = -\frac{\partial g}{\partial q}\left(\frac{\partial g}{\partial \theta}\right)^{-1}$.