

EXPANDING STRUCTURAL DESIGN THROUGH SHAPE OPTIMIZATION AND MICROSTRUCTURES

by

Davi Colli Tozoni

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NEW YORK UNIVERSITY

JANUARY, 2023

Professor Denis Zorin

© DAVI COLLI TOZONI

ALL RIGHTS RESERVED, 2023

DEDICATION

To my parents, with love.

ACKNOWLEDGEMENTS

First, I would like to thank everyone in the Geometry Computing Lab (GCL) at NYU, especially my advisor, Prof. Denis Zorin. Thanks for this amazing learning opportunity and for your guidance and patience during this time. I also want to thank Prof. Daniele Panozzo for all his help. I really appreciate that.

In addition, I want to thank nTopology and their simulation team for my internship opportunities. I was able to work on very interesting and difficult problems that helped me understand more how research can have a good and important impact on the industry.

A very special thanks goes to my family, for always being there and trying to help me in any way they can. Thanks, Dona Pascoa and Seu Sergio! My father is a Mathematics Professor and he has always been super excited and proud to work with research, and I'm happy that some of that joy was passed to me.

Finally, I would like to thank all my friends that helped me through this process. Some of you are in Brazil, others in the US, and some are elsewhere, but I hope that all of you know how very important you are to me.

It was an amazing experience to be able to study at NYU with such an amazing group of people. I'm very proud and thankful!

ABSTRACT

3D printing and other modern manufacturing technologies enabled accessible customized fabrication of small-batch and unique objects, with unparalleled flexibility in shape design. However, creating structures that are able to meet the design requirements is not an easy task, and doing it manually can be a very slow and tedious process. In this context, structural optimization techniques can be very useful and help automate the design and analysis process.

This thesis describes techniques that expand the usage of structural optimization for digital fabrication by formulating optimization to be used with more realistic simulation models. We develop an efficient method (differentiable simulation) to compute gradients of optimization objectives depending on solutions of the physical deformation equations, which can be used to optimize the shape, material, and physical properties of our domain. We provide ways of expanding the use of two-scale topology optimization by presenting microstructures that have a smooth map from material to geometry and which can be used on curved shapes defined by irregular lattices with close to rhombic cells. Finally, we introduce two low-parametric microstructures that together are able to cover almost the whole possible range of elastic properties for isotropic metamaterials.

Our results in simulation and physical experiments, both for static and time-dependent scenarios, show the advantages of our techniques and how they can be applied in practice.

CONTENTS

Dedication	iii
Acknowledgments	iv
Abstract	v
List of Figures	x
List of Tables	xxii
1 Introduction	1
1.1 Realistic Differentiable Simulation	3
1.2 Two-scale Optimization with Microstructures	4
2 Optimizing Contact-based Assemblies	7
2.1 Introduction	8
2.2 Related Work	10
2.3 Problem Formulation	13
2.3.1 Overview	13
2.3.2 Notation	16
2.3.3 Elastic deformations with contact	17

2.3.4	Variational form and constraint approximation	19
2.3.5	Optimization objectives	22
2.4	Shape Derivatives and Discretization	27
2.4.1	Shape derivatives	27
2.4.2	Discretization	29
2.5	Method	30
2.6	Evaluation	33
2.7	Examples	40
2.7.1	2D examples	40
2.7.2	3D examples	46
2.8	Conclusions and limitations	56
2.9	Chapter Notes	57
2.9.1	Formulas for the adjoint equation terms.	57
2.9.2	Formulas for the discrete shape derivative coefficients $S_m[u, \rho]$	57
2.9.3	Fabrication processes	59
2.9.4	Adaptive Quadrature	60
2.9.5	Meshing	62
3	Differentiable solver for time-dependent deformation problems with contact	64
3.1	Introduction	65
3.2	Related Work	67
3.3	General static formulation	72
3.4	Adjoint-based objective derivatives in general form	75
3.5	Time-dependent problems	78
3.5.1	Implicit Euler	80
3.5.2	General BDF time integration	82

3.5.3	Initial condition derivatives	83
3.6	Physical model and discretization	84
3.7	Parametric derivatives of forces	85
3.7.1	Obtaining shape derivatives on perturbed domain	85
3.7.2	Volume forces	86
3.8	Objective derivatives	94
3.8.1	General forms of objectives	94
3.8.2	Specific objectives	95
3.9	Implementation	100
3.10	Results	101
3.10.1	Shape Optimization Without Contact	102
3.10.2	Shape Optimization Contact	104
3.10.3	Initial Conditions	107
3.10.4	Material Optimization	108
3.11	Concluding Remarks	115
4	A Low-Parametric Rhombic Microstructure Family for Irregular Lattices	117
4.1	Introduction	118
4.2	Related Work	122
4.3	Problem formulation and solution overview	126
4.4	Choice of cell shape and structure	129
4.5	Homogenization and shape optimization	131
4.5.1	Homogenization and shape optimization on square cells	131
4.5.2	Deformed structure homogenization	134
4.6	Material-to-geometry map construction	136
4.6.1	Orthotropy	136

4.6.2	Map construction for $\alpha = \pi/2$ (squares)	137
4.6.3	Arbitrary angles and base materials	145
4.7	Quad mesh optimization	149
4.8	Evaluation and examples	152
4.9	Conclusions and Limitations	162
4.10	Chapter Notes	165
4.10.1	Microstructure geometry representation	165
4.10.2	Sweep parameters	167
5	A parametric class of composites with a large achievable range of effective elastic properties	168
5.1	Introduction	169
5.2	Background and related work	172
5.3	Periodic structure families	178
5.4	Methods	180
5.5	Numerical studies	184
5.6	Discussion, conclusions and future work	194
5.7	Chapter Notes - Approximation accuracy	196
	Bibliography	199

LIST OF FIGURES

2.1	Coat rack stress: stress concentrations on coat rack example. Left figure presents how load is applied. Middle figures show stresses on each PLA piece. Right figures show the optimized coat rack and 3d printed connectors assembled with wooden dowels.	7
2.2	Notation for boundary parts for a simple assembly: Γ_D is attached to the vertical wall (red), Γ_C can slide along the ground (yellow), Γ_N is where the load is applied, and Γ_S is the contact area of two parts. We view both parts as a single domain Ω consisting of detached parts.	17
2.3	Smoothing function N_η , h_η , and g_ϵ	20
2.4	Assembly, disassembly and parallel alignment objectives. Top: left shape is easy to assemble in $-y$ direction, while shape on the right cannot be assembled without effort (high objective value, due to normals pointing in the direction opposite to y). Middle: For the same disassembly direction y , and user-defined load l , the assembly on the right falls apart under the load; for the shape on the left, the parts cannot move apart in direction y under load l . Bottom: left shape has a large region parallel to disassembly direction (marked in blue), making it hard to disassemble in any direction other than y , while shape on the right allows for multiple easy disassembly directions (shown in light red).	24

2.5	Convergence plots for connector (Figure 2.9) on the left and stool (Figure 2.24) on the right side. The charts show stress objective vs iterations.	33
2.6	Examples used for validation. The left: box in contact with a rigid surface. The right: joint in contact with a holder. We use the same notations for boundary parts as in Fig. 2.2. Starts from this Figure, all the orange arrows in the following figures illustrate the direction of load.	34
2.7	Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing α . The right is relative displacement error with changing η	34
2.8	Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing η_n . The right is relative displacement with changing μ	35
2.9	Optimization results when using different combinations of objectives. For both baseline ($p = 20$) and compliance ($p = 2$) scenarios, we use a target volume equal to the original one.	36
2.10	Optimization considering multiple loads.	36
2.11	Multimaterial optimization. The stiff material has Young's modulus equal to 100 and Poisson's ratio of 0.3, while the flexible one has Young's modulus equal to 1.0 and Poisson's ratio of 0.0.	37
2.12	Middle shapes: with and without friction; rightmost shape: the optimization of the top assembly is disabled.	37
2.13	Optimization results when using assembly and parallel alignment objectives, with two different strengths. The white arrow demonstrates the dissemble direction.	38
2.14	Optimization results when using a disassembly energy term with low tolerance. As before, the white arrow demonstrates the dissemble direction.	39

2.15	Displacement results when optimizing our connector using varying assembly weights and a fixed disassembly weight.	40
2.16	Lever optimization	42
2.17	Lever optimization fabrication	42
2.18	Initial and optimized <i>bridge</i> model.	42
2.19	Fabricated initial and optimized <i>bridge</i> shapes.	43
2.20	Top: undeformed shapes of different versions of the <i>hook</i> model. bottom: deformations under load (displacements exaggerated).	44
2.21	3D printed <i>Hook</i> loading (Please see supplementary video.)	44
2.22	<i>Tower crane</i> optimization	45
2.23	<i>Tower crane</i> fabrication	45
2.24	Left: the initial <i>stool</i> model; Right: the result after optimizing the white (PLA) pieces.	46
2.25	Stool stress: stress concentrations on stool example. Top figures present how load is applied and general stresses (restricted to PLA). Middle figures show stresses on each PLA piece. Bottom figure shows cross section of region where stress is more concentrated.	47
2.26	<i>Bench</i> model with two different types of legs and the initial and optimized connectors.	47
2.27	Bench stress for twisted and bended set of legs	48
2.28	<i>Coat rack</i> base stress distribution.	49
2.29	<i>Coat rack</i> hook connector stress distribution.	50
2.30	Mounted coatrack	50
2.31	Initial and optimized <i>tent</i> assemblies.	51
2.32	Tent stress	52
2.33	Tent connections	52
2.34	Dodecahedron	53

2.35	Dodecahedron stress: simulation and optimization are executed on an octant of the original shape, using reflection symmetry on all three axes. A diagonal force to the center of the shape is used.	53
2.36	Connectors optimization: result after optimizing only sphere connectors. We show our two different types of connector (showing two sides of each) and two different types of pipes, which were not optimized with exception of their tips.	54
2.37	Truss	55
2.38	Example of different regions in piecewise quadrature computation with $\eta = 10^{-2}$. In this case, we have two hyperbolas. So, in total, 5 different regions. On the right side, an example of how the triangles can be split.	61
3.1	Notation for domains and maps we use, see Table 3.1.	75
3.2	Domain perturbation θ , see Table 3.1.	75
3.3	The result of the shape optimization (blue surface) matches the target shape (wire-frame).	102
3.4	Result of shape optimization to minimize the average stress.	103
3.5	Result of stress minimization on a beam standing on a platform, with two loads on its sides.	104
3.6	Result of shape optimization to minimize the maximum stress.	104
3.7	Shape optimization of a hook to reduce stress concentration (left). Fabricated results with maximum load before failure (right).	105
3.8	Result of shape optimization of a hanger to reduce stress concentration.	106
3.9	The result of the shape optimization (blue surface) matches the desired trajectory (wire-frame).	107
3.10	Result of shape optimization to reduce stress.	107

3.11	Optimizing the initial velocity of a bouncing puzzle. Target is shown as a black outline while the trajectory being optimized is blue.	108
3.12	The initial conditions of the green bunny (left) are optimized to push the blue bunny in the corner of the pool table (right).	109
3.13	Optimized material parameters to obtain a displacement (blue surface) in y -direction similar to a sine function for a linear material model (wire-frame). . . .	110
3.14	Optimization of the materials of a bridge (blue surface) to match a forward simulation (wire-frame).	111
3.15	Material optimization (blue) to match real data (orange).	111
3.16	We acquire the deformation of a silicon micro-structure using a 3D scanner, and use the observed displacement, created by a controlled load, to reconstruct its material parameters. The simulated mesh (transparent blue) matches well the 3d scan data (orange). The optimization is initialized with $E = 1e6\text{Pa}$, $\nu = 0.3$ and converges to $E = 2.27 \times 10^5\text{Pa}$, $\nu = 0.348$	112
3.17	Non-linear transient simulation of a kangaroo (blue surface) bouncing on a plane to match a target shape (wire-frame).	112
3.18	Optimization of the friction (blue surface) to match the target shape (wire-frame).	113
3.19	Material optimization to increase the bouncing height.	113
3.20	Material optimization (blue) to match experimental data (orange).	114
4.1	The cells of a quadrilateral mesh are optimized to become quasi-rhombic; then material properties (variable Young's moduli and Poisson's ratio) are assigned to the cells. The assigned material properties are used to evaluate the geometric parameters of a tileable microstructure, encoded with a smooth spline map.	119

4.2	An array of microstructures with Young's modulus E varying from 0.005 to 0.1 and ν varying from -0.15 to 0.5. Top: obtained by pointwise inverse homogenization of [Panetta et al. 2017a]; Bottom: our family.	121
4.3	Parameters defining our elementary cell geometry: 4 displacement parameters (two nodes 3,4 have only one free displacement and the node 1 is fixed, due to symmetry), and 4 radii, one per node. The displacements are specified in barycentric coordinates with respect to the triangle OAB.	127
4.4	We consider structures consisting of edges of the graph depicted on the left, with degrees of freedom consisting of radii at nodes and node displacements. Right: several examples of structures with few edges/simple topology.	129
4.5	Coverage in (E, ν) space for three cell structures, with smaller and larger coverage. 130	
4.6	A tiling of a domain Ω with base cell Y having geometry ω [Panetta et al. 2015a]. 132	
4.7	Left: An example of the cell geometry discretization. Right: fluctuation strain norm obtained from (4.3) for the constant strain e^{11}	134
4.8	Transversal subspace V illustrated for the simplified case of a 3-parametric structure and 2-dimensional property space (E, ν) (In our construction, we consider spaces that are 8 and 4-dimensional respectively). The curved surfaces are isosurfaces of $E(\mathbf{p})$. The map H , restricted to V , is bijective.	138
4.9	Material properties coverage for the subset of samples of B_p with constant values of p_1 and p_2 fixed (6D space, samples shown in green), p_1, p_2, p_5 fixed (5D space, samples shown in blue) and V (4D space, shown in red), compared to the full set of samples \mathbf{p}_j , shown in purple.	140
4.10	Plots of parameters p_1 and p_4 for pointwise inverse homogenization. Note the lack of smoothness; other parameters also have a varying degree of noise. The color in the charts represents the geometric parameter value, varying from blue to red. . .	143

4.11	Changes of geometry as a result of small changes in Young's modulus Top: obtained by pointwise inverse homogenization [Panetta et al. 2017a]; Bottom: our family. .	143
4.12	Expanding the initial material-to-geometry map coverage. Left: original coverage; Right: expanded coverage.	143
4.13	Lines defining the shape of our convex domain.	144
4.14	Domain coverage for $\alpha = \pi/2$, with approximating linear bounds. Left: resulting polygon when restricting minimum and maximum Young's modulus to 0.01 and 0.1; Right: convex polygon (in yellow) when Young's modulus minimum and maximum values are unlimited. We show a comparison with the bounds in the constrained case (in red). Depending on the Young's modulus range the application needs, a different polygon may be obtained in the optimization. If the application requires larger stiffness values, the polygon on the right should be preferred. On the other hand, if a larger range on Poisson's ratio values is needed, the polygon in red is a better option.	145
4.15	Dependence of the geometry parameters \mathbf{p} on the angle, for $E = 0.076$ and $\nu = 0.263$ (using base material $E = 1, \nu = 0$).	147
4.16	Left: coverage for $\nu = 0, E = 1$. Right: coverage for $\nu = 0.3, E = 2$	147
4.17	Plots of parameters $p_1 \dots p_8$ as functions of (E, ν) for $\alpha = \pi/2$. The color of the surface corresponds to parameter values.	149
4.18	Plots of parameters for $\alpha = \pi/4$. The color of the surface corresponds to the parameter values.	150
4.19	A quadrilateral mesh (left) is optimized with (right) and without (middle) the boundary term. The shape of the eyes and the mouth is lost in the middle image, while it is preserved on the right. The color for each quad maps to α (the minimum angle of the cell).	152

4.20	Top left: example of a cropped regular grid of perfect square cells. Top right: deformation using the model produced with square grid as shown on the left. Bottom: result using our new method.	154
4.21	Left: deformation result after one level of quad mesh refinement when using our new method. Right: result using regular grid with cut cells.	155
4.22	Middle: non-deformed solid shape. Left: deformation of the structure using geometry optimized for $\alpha = \pi/2$ everywhere; the solid shape in the background shows the reference deformation of a single-material solid. Right: using correct cell geometry with parameters $\mathcal{P}(\alpha)$ everywhere. The microstructure has material parameters $E = 0.02$ and $\nu = -0.4$. Bottom part is fixed (0 Dirichlet), and on the top part we apply a force parallel to the slanted side (at an angle of $4\pi/3$).	155
4.23	Left: deformation of an object constructed using a table of structures obtained using pointwise inverse homogenization as in [Panetta et al. 2017a]. Right: deformation of the object constructed using our family, with the same boundary conditions. The gray shape in the background shows the ground truth obtained by simulation on a solid quad mesh with materials assigned per quad.	156
4.24	Coverage in (E, ν) plane for $\alpha = \pi/2$ obtained by pointwise shape optimization (blue) in the range $E = 0.01 \dots 0.1$ and $\nu = -0.4 \dots 0.9$. The coverage of our family is shown in orange.	156
4.25	Top: deformation of a shape with constant assigned materials and regular connectivity. Bottom: deformation of the same shape, with irregular mesh connectivity. The deformed pattern (simulated using FEM on a fine mesh) is compared to the simulated homogeneous mesh with equivalent material properties ($E = 0.05$, $\nu = 0.4$). In both cases, a uniform vertical load is applied on the top, while we keep a 0 Dirichlet condition at the bottom.	157

4.26	An example of the twisted structure. Left: initial mesh with perfect squares. Middle: twisted mesh by a rotation r of 20 degrees. Right: final structure after rotation.	158
4.27	Plot of twisting accuracy when the rotation angle r is increased. The plot shows the Frobenius norm of the distance between the target elasticity tensor and the actual elasticity tensor obtained by running homogenization on the compound pattern.	158
4.28	Refinement of the twisted structure. Left: one level of quad refinement. Right: two levels of quad refinement.	159
4.29	Non-rhombic cell deformation: the center vertex is slowly translated from $(0, 0)$ to $(1, 1)$, increasing non-rhombicity of cells and distance from the homogenized properties of the target cell to target material properties.	159
4.30	Plots of accuracy in non-rhombicity experiment. Left: plot shows how rhombic error varies when d value is increased. Notice that the change is almost linear and, as expected, error for Q1 and Q3 are the same. Right: smooth variation of target error when rhombic error increases. Dots indicate corresponding error for our example models in Figure 4.30.	160
4.31	Scalability experiment. Left: example of optimized structure with 4 levels of refinement on the initial mesh, which means a final number of 1024 faces. Right: Plot (in log-log scale) showing how our running time grows with the number of quad faces. For this experiment, we used a machine with 20 cores and 32GB of RAM.	160
4.32	Error due to merging cell structures. Left: radii of contact nodes of two structures with same material properties (in red) and resulting radius after merge (in blue). Right: base homogenization cell to verify the properties of merged structure (which are significantly different than original).	161

4.33	Accuracy of \mathcal{P} away from sample points, measured as the difference between directly computed and interpolated E and ν	162
4.34	Summary table of the results fabricated with our method. The plier model was laser-cut in acrylic, while the rest were laser-cut from foam. From left to right: angle per quad on the optimized mesh, optimized material distribution (Young's modulus and Poisson's ratio), final geometry at rest, deformed pattern (simulated) and photograph of deformed pattern (fabricated). Video clips of the deformation of the fabricated results are provided in the additional material. The minimum angle after optimization is 57.4 deg for the pliers, 64.5 deg for the ghost, 66.0 deg for the disk, and 67.2 deg for the bar. On these examples, our Young's moduli ranges from 0.005 to 0.25, and our Poisson's ratio ranges from -0.27 to 0.8.	163
4.35	Left: example of geometry in our 3D family. Right: comparison of the material properties obtained with our initial 9 parameters and the reduced set of only 3. . .	165
5.1	(A,B) parametric structure families, (C) Domains of achievable effective elastic properties and Hashin-Shtrikman theoretical bounds ($E_b = 1, \nu_b = 0$) in (E, ν) domain.	171
5.2	(A, B) Hashin-Shtrikman bounds for the elastic moduli of isotropic void-material composites: (A) in bulk and shear modulus coordinates and (B) the same bounds transformed to Poisson's ratio and Young's modulus coordinates. (C,D) Hashin-Shtrikman (grey) and Cherkaev-Gibiansky (blue) bounds for elastic moduli of isotropic bimaterial composite ($\nu_s = \nu_w = 0, E_w = 0.001E_s$) (C) in terms of bulk and shear moduli and (D) in terms of Poisson's ratio and Young's modulus. For brevity, here and below Hashin-Shtrikman bounds are labeled on images as HS bounds and Cherkaev-Gibiansky bounds – as CG bounds.	175

5.3	Known extremal composites (A) Coated spheres assemblage, exhibiting extremal bulk modulus, (B) Vigdergauz structure, achieving maximum bulk modulus (C) rank 3 sequential laminate, exhibiting maximum bulk and shear modulus (D) “Herringbone” laminate structure providing Poisson’s ratio of -1 and (E) and its unimode analog. (F) Auxetic material made of rotating squares [Grima et al. 2005] (G) Sigmund extremal structure, providing maximum bulk and minimum shear modulus (H) Chiral isotropic auxetic material [Prall and Lakes 1996] and (I) similar structure, suggested in [H et al. 2011]	176
5.4	(A,B) Structure families: (A) Triangle, (B) Hexagon. (C,D) Limit structures: (C) Triangle, (D) Hexagon	179
5.5	Diagrams of (A) a cell computational domain (B) rank-1 laminate (C) prescribed rotational symmetries of a computational domain.	183
5.6	(A, B) Coverage of Hashin-Shtrikman bounds with the new structures of all types. Letters denote tentative positions of the extremal composites presented in Figure 5.3. (A) - covered sets, (B) - raw data samples and the conjectured analytical bounds. (C) Expansion of the coverage as the number of beams $n = 1/p_3$ increases. (D) Comparison of the coverage obtained in this work with Triangle and Hexagon structures with ($n = 30$, $\nu_b = 0$, only extreme structures are shown, red points), in comparison with the widest coverages obtained based on previous numerical studies [Panetta et al. 2015b, 2017b] (blue points), and analytically computed coverage achieved with isotropic laminates proposed in [Milton 1992] (yellow region, parameters of lamination are swepted maintaining isotropy, the ratio between stiffnesses of weak and strong phases is 10^{-6}	188
5.7	(A) Coverage of elastic properties for positive Poisson’s ratios as a function of the number of beams in the joint (B) Coverage achieved in the limit case of two-scale structure (C) coverage measure Δ_R as a function of volume fraction ϕ	189

5.8	Displacement fields in the horizontal tension ($pq = 11$, Triangle limit structure) cell problem. Rotation of solid parts of the structure, leads to auxetic behavior.	189
5.9	Limit structures approaching left HS bound. (A,B) Left-bound extreme elastic moduli, achieved with the Hexagon limit structure; chart colored by values of p_1 (A) and p_2 (B). (C) Dependence of the achievable negative Poisson's ratio as the function of chirality angle. Chirality angle p_3 sweeps are performed for several values of p_2	190
5.10	Coverage of Hashin-Shtrikman bounds with triangular and hexagonal structures for a set of intermediate volume fractions. (A) Structures of the first type, (B) structures of the second type. (C) Coverage of Cherkaev-Gibiansky bounds with bimaterial composite for two volume fractions.	190
5.11	(B) Change in coverage with addition of indentations. (C) Von Mises stress concentrations caused by the introduction of beam indentations.	193
5.12	Possible generalizations of the extremal structures. (A) Structure with an improved strength due to elimination of stress concentrations (B) Structure with wide coverage of elastic moduli at intermediate volume fractions.	196

LIST OF TABLES

2.1	Final state of objective term values when running optimization with different assembly weights	39
2.2	Summary of experiments. In constraints column, “/” means that the weight is modified after each remeshing and varies between these values linearly. Column Relative gradient norm presents the ratio of the final gradient norm and the initial one from the first iteration. The last column (Final L_p-norm reduction) shows the reduction in the stress objective due to the last outer iteration, expressed as a percentage of the initial objective value. *For both bridge and coat rack (hooks), running 5 more outer iterations reduces stress to 1.80 and 346.51, relative gradient norm to 2.6e-2 and 2.81e-2, and final L_p -norm reduction to 0.95% and 0.00%, respectively. For fabrication, we used optimization results at outer iteration 5 for lever, hook, crane and at iteration 10 for bridge and coat rack.	41
3.1	Notation.	71
5.1	Norm of the shape derivatives vector as a function of iteration number	195
5.2	Matching regular grid computations and unstructured mesh computations. Beam thicknesses parameter for the regular grid is chosen to match the values of the Young’s modulus found on an unstructured grid	197

1 | INTRODUCTION

In recent years, advances in manufacturing technologies made highly customized objects, from toys to aerospace components, easier and cheaper to fabricate. These technologies include 3D printing, laser cutting, and modern CNC milling, with each technology having its own limitations and advantages.

A common characteristic of some of these technologies, 3D printing in particular, is that production cost is usually not directly linked with geometry complexity. Moving from a simple shape to a more complex and lighter one that can serve the same purpose may improve cost by reducing the amount of material and time used for production. As a consequence, there is a need to develop automated technologies that allow to create highly complex geometric structures optimized for a specific goal, a task for which conventional CAD software is ill-suited.

In this context, structural optimization plays a crucial role and includes a large set of techniques that can help designers to build new and improved objects and tools, according to users' needs. The overall goal is to optimize a mechanical structure under possibly multiple and time-varying conditions and external forces defining loads and displacements. Structural optimization involves simulating the physical behavior of the object, extracting quantities of interest from the solution (for example, displacements and stress distributions), and modifying the shape or material properties to improve an objective function depending on these quantities, which measures the quality and

performance of the object.

Two main techniques are widely used in structural optimization: topology optimization (TO) and shape optimization (SO). While they share similarities, there are also important differences between these methods. TO uses implicit surface representations and optimizes quantities on a fixed spatial mesh or grid, while SO more commonly uses a Lagrangian surface description and evolves the mesh, which may require frequent remeshing. One of the main advantages of TO is that it supports topology changes, which is not necessarily true for shape optimization, even though some methods may enable limited variations. On the other hand, having a fixed mesh or grid makes it more difficult to compute physical properties which may be critical for specific objectives, especially close to the surface (e.g., rapid stress variations near corners), given that the boundary is represented only implicitly and approximately. In contrast, it is much easier to resolve such boundary effects in SO context. The choice between using one or the other technique (or maybe even both) depends on the objective and constraints of the user and the application.

Notice that, while Structural Optimization algorithms are widely used in academic research, they also have been increasingly applied in the industry in the latest decades, and can be found in software platforms of companies like nTopology, Altair, AutoDesk, 3D Systems, Dassault Systemes, and PTC. Most of the methods developed in this thesis are released as open-source software.

In this thesis, our objective is to develop structural optimization in a number of contexts making it more practical (1) by including more realistic physical models, friction and contact in particular; (2) by developing more efficient optimization pipelines; (3) by increasing the number of scenarios where optimization can be used; and (4) by expanding the use of microstructures. In the next chapters, we detailed how SO and TO techniques can be applied in different scenarios to achieve our goal. In projects presented in Chapters 2 and 3, our main focus is on stress optimization and handling contact simulations (where boundary shape is extremely important), so we choose SO.

On the other hand, in Chapters 4 and 5, we focus on a particular type of two-scale TO techniques, mixing material optimization with the use of microstructures (which are, in turn, obtained with the help of SO).

1.1 REALISTIC DIFFERENTIABLE SIMULATION

An important focus of this thesis is to use more realistic simulations together with optimization, reducing the reality/simulation gap in order to better predict the behavior of objects produced through digital fabrication. In this context, finding ways of modeling and solving common physical behavior problems becomes an essential question. One of these problems in elasticity and mechanical simulation is dealing with contact accurately and robustly.

Moreover, computing gradients of the objective with respect to optimization parameters (shape, initial conditions, material properties, etc) is essential for efficient optimization, as gradient-based methods for sufficiently well-behaved objectives significantly outperform gradient-free methods. Since our objectives may depend on the simulated behavior of objects, such as displacements, our objective will depend on solutions of physical equations, making it important to have a *differentiable* formulation and solver for elasticity and contact/friction, i.e., a solver that outputs not just the solution but its gradients (sensitivities) with respect to optimization parameters. In our work, we present consistent and efficient ways of computing gradients for different contact models based on adjoint formulations and show how we can use optimization tied to these models to improve the performance of objects and structures in practice, and also reduce the reality/simulation gap by computing unknown parameters for our physical models.

In chapter 2, we focus on optimizing shapes of assemblies, i.e., objects constructed from separate parts that are held together by contact and friction. These assemblies are useful in a variety of scenarios: for example, to fabricate larger objects piece by piece, or in cases when different

materials need to be used for different parts (e.g., a chair made of wood with plastic joints), or e.g., to make it possible to disassemble and store an object in a more compact way. Since we expect objects to be mostly stationary and with the same areas of contact between different parts during operation, we use a simplified contact and friction model with static elasticity simulation where we know beforehand the areas of contact between different bodies.

Next, in Chapter 3, we focus on extending our adjoint formulation to a more robust contact model based on incremental contact potentials (IPC), [Li et al. 2020b] which is also differentiable in time-dependent cases. We present a consistent formulation that allows optimizing shape, material/physical parameters, and initial conditions. Our method uses higher-order FEM for space and time and can be used with a general class of elasticity models. In addition, we handle complex geometry for contact and friction, as proven in our many examples. In the end, we also present physical experiments showing how our results can match well with reality.

1.2 TWO-SCALE OPTIMIZATION WITH MICROSTRUCTURES

Using formulations like the ones presented in Chapters 2 and 3 allows for optimizing material distribution in a given domain, in a similar way to TO. The standard approach in TO is to use densities (one per cell) as optimization parameters and then map these to elasticity properties in our shape, but this may present critical drawbacks. A very important issue is related to resolution, since for a 3D shape the number of density parameters increases as the third power of size, making large-scale problems expensive to solve.

In addition, the map from density to elasticity may not be realistic. For example, 50% density doesn't necessarily translate to half Young's modulus. Finally, due to a large number of cells and optimization parameters, it ends up being necessary to apply smoothness and other filters to avoid falling into low-quality local optima.

Instead, we attempt a so-called two-scale scheme, where we directly optimize the material properties (like Poisson's ratio and Young's modulus, for isotropic materials) in each cell of our domain in order to achieve our desired behavior, while assuming constraints on material properties. Then, we place microstructures corresponding to materials in each cell and stitch them together to obtain our final shape. The direct use of elastic properties makes it possible to obtain a more realistic behavior and the use of microstructures reduce the number of variables in our optimization, since each microstructure occupies a larger portion of the object compared to a single density cell. Another important reason for using microstructures is that most 3D printers are only able to print with a single or a small number of different materials, which limits the design options or quality of the objects produced. The idea of varying macroscopic material properties through small-scale geometric changes can increase the range of effective material properties one can obtain.

However, there are obvious difficulties with the two-scale approach: first, we need a direct map from material properties to geometry (which can limit our achievable material properties); second, we need to be able to tile consecutive microstructures (which may be very different); and, third, microstructures are usually defined on square (in 2D) or cubic (in 3D) cells, which makes it difficult to adapt to curved shapes. We propose solutions to these three problems in Chapters [4](#) and [5](#).

In chapter [4](#), we demonstrate how to produce a map from a range of material properties to a family of microstructures, with a smooth change of geometry for small perturbations in material. Our family of microstructures has guaranteed tiling and smooth transitions between consecutive cells. Smooth dependence on parameters makes it possible to represent the family as 8 splines, which means having a constant time computation for our geometries. In addition, our family supports non-square cells, which enables much better adaptation to complex domains without losing material property accuracy.

Then, in chapter [5](#), we explain how to expand the range of material properties that can be achieved

with microstructures by using two families of structures that fit in hexagonal shape cells. Our objective is to have a large range of effective material properties offering the most flexibility in design.

We show how to achieve that with microstructures that can be parameterized with only four parameters and are sufficiently simple, and, at the same time, they can attain a large range of elasticity properties, close to the theoretical limits. We were able to approach the theoretical bound for positive Poisson's ratio arbitrarily closely, within limits of numerical accuracy. For negative Poisson's ratio, we cover a larger fraction of the region defined by the known theoretical bounds than all previously known work.

2 | OPTIMIZING CONTACT-BASED ASSEMBLIES

This chapter presents our work on shape optimization for static simulations with contact. It is largely based on my publication at SIGGRAPH Asia 2021[Tozoni et al. 2021] in collaboration with my colleague YunFan Zhou from NYU and my advisor, Prof. Denis Zorin. In this work, I was responsible for formulating the physical model, deriving the shape derivative formulas, and implementing the nonlinear simulation and the shape optimization technique. In addition, I performed computational experiments, designing models/examples and running optimization/simulation at NYU’s HPC, besides analyzing the corresponding data. Finally, I also helped in physical experimentation in our laboratory.

Modern fabrication methods have greatly simplified manufacturing of complex free-form shapes

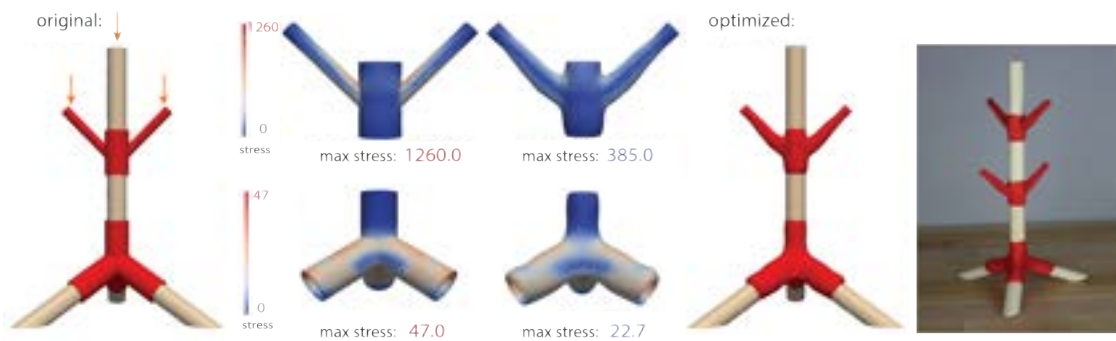


Figure 2.1: Coat rack stress: stress concentrations on coat rack example. Left figure presents how load is applied. Middle figures show stresses on each PLA piece. Right figures show the optimized coat rack and 3d printed connectors assembled with wooden dowels.

at an affordable cost, and opened up new possibilities for improving functionality and customization through automatic optimization, shape optimization in particular. However, most existing shape optimization methods focus on single parts. In this work, we focus on supporting shape optimization for assemblies, more specifically, assemblies that are held together by contact and friction. Examples of which include furniture joints, construction set assemblies, certain types of prosthetic devices and many other. To enable this optimization, we present a framework supporting robust and accurate optimization of a number of important functionals, while enforcing constraints essential for assembly functionality: weight, stress, difficulty of putting the assembly together, and how reliably it stays together. Our framework is based on smoothed formulation of elasticity equations with contact, analytically derived shape derivatives, and robust remeshing to enable large changes of shape, and at the same time, maintain accuracy. We demonstrate the improvements it can achieve for a number of computational and experimental examples.

2.1 INTRODUCTION

Creating shapes optimized for a particular function is one of the main tasks in computer-aided design. Modern fabrication methods have greatly simplified the creation of complex free-form shapes at an affordable cost and opened up new possibilities for improving functionality and customization through automatic optimization.

Methods for constructing optimal shapes have enjoyed considerable attention in a variety of settings, such as large-scale architectural forms, engine parts, footwear, medical prosthetic devices and metamaterial structures. Notably, most of the work focuses on designing continuous structures, fabricated from one material, or a set of materials fused or glued together.

In this work, our focus is on supporting shape optimization for *assemblies*, and more specifically for assemblies that are held together by contact and friction, a setting which has received relatively

little attention. At the same time, assemblies are ubiquitous, as most manufactured objects around us are assembled from separate parts, often made from different materials. For example, the steel legs of a table may be inserted into openings of a wooden or MDF top; a phone may have a snap-on plastic protector or cover; a prosthetic device is attached to the body with friction. While the specific mechanisms for holding objects together may vary broadly, they are all based on combining deformation with contact and, in many cases, friction.

In all these cases, contact plays a major role in the function and mechanical behavior of the assembled object. Shape optimization helps to achieve better performance or save on the costs of material for fabrication (for additive fabrication these two are closely related). Some of the important measures of performance of assemblies include total volume or weight, total deformation energy, maximal stress and more complex measures such as permeability.

In this paper, we present a formulation and a robust numerical method for computing optimized shapes in the presence of contact and friction. Compared to shape optimization tasks not involving contact, the problem is significantly more difficult to solve, as it involves complex inequality constraints required for handling collisions and friction. The resulting problem is non-smooth, and often hard to solve sufficiently accurately. Our overall approach is to use a smoothed version of the problem [Eck et al. 2005] amenable to standard optimization techniques on the one hand, and allowing us to approximate the desired solution as close as possible on the other hand.

Contributions. In summary, the contributions include

- A shape derivative-based formulation for optimization problems with contact and friction, building on [Maury et al. 2017];
- A novel shape-optimization framework based on FEM discretization of this formulation, capable of handling contact regions between two deformable objects as well as a deformable

and a rigid object. It provides sufficiently accurate elastic deformation computations to support, e.g., max stress reduction.

- The framework supports conventional functionals (stress-based and volume) and new, contact-specific functionals (assembly and disassembly, parallel alignment) that ensure that connection strength is maintained and that at the same time the parts held together by contact can be assembled. The framework also supports optimization involving multiple load scenarios.
- We demonstrate a range of 2D and 3D examples of shape optimization, and qualitatively evaluate these examples using laser cutting and 3D printing to fabricate them and demonstrate the expected behavior.

2.2 RELATED WORK

There is a broad range of work on shape/topology optimization and related methods, but relatively few works were trying to solve problems with contact. We focus on related work on shape optimization with contact, and briefly mention other shape/topology optimization research that we rely on.

Shape optimization with contact.. Some previous works have considered contact of a soft body with rigid surfaces, for example, [Beremlijski et al. 2014], [Haslinger et al. 1986] and [Herskovits et al. 2000]. While other some other works have studied the interaction of two or more bodies in contact, like recent works from [Maury et al. 2017], [Desmorat 2007] and [Stupkiewicz et al. 2010]. Most papers do not consider friction, and those which do often consider simplified (compared to the standard Coulomb) friction models as discussed in [Maury et al. 2017].

For contact models, there are two families of algorithms, Lagrangian and the penalization methods.

The first type of methods adds Lagrange terms for the model constraints to the objective function and uses sub-gradient-based optimization to deal with the fact that the problem is non-smooth. Examples can be found in [Herskovits et al. 2000] and [Stupkiewicz et al. 2010], where a bilevel approach and an augmented Lagrangian method are used. The second type of methods, penalization methods, use smooth approximations to the problem, which add terms to the variational formulation. Our method belongs to this category. For this type of method, the objective function is smooth, which considerably simplifies optimization. In our work, we choose to use penalization method following the mathematical model presented by Eck et al. [Eck et al. 2005]. In [Maury et al. 2017], a similar overall approach with level-set discretization. While level-set modeling has many advantages (e.g., allowing for easy topology changes) in our experience, it is not well-suited for handling important types of problems, in particular, those involving stress reduction, which is our focus in this work, as stress is harder to resolve precisely.

Stress minimization. Maximal stress minimization was considered in a number of papers; the formulation closest to ours is [Panetta et al. 2017c], where *worst-case* optimization for periodic metamaterial structures is considered; it uses parametric periodic shapes, which are meshed for shape derivative computation, and does not consider contact. Earlier work on minimization of maximal stress is [Allaire et al. 2004], which applies topology optimization to design lightweight minimal-stress objects built from sequentially laminated composites. Another similar work is [Allaire and Jouve 2008], which applies the level-set topology optimization method to minimize the p -norm of stress. None of these works consider contact, and level-set methods (with Eulerian discretizations on a fixed grid) require impractically fine meshes for accurate optimization of high p norms of stress. Other works considering max stress include [Xia et al. 2012; Polajnar et al. 2017; Lian et al. 2017; Sonmez 2009; Van Miegroet and Duysinx 2007]. In the computer graphics community, [Stava et al. 2012] was one of the first works to introduce heuristic shape correction techniques that effectively result in stress reduction. [Zhao et al. 2016; Zhou et al. 2016] consider

problems involving bounding von Mises stress.

Contact and friction modeling. The literature on contact is extensive, and we mention only few most closely related works here; for general theory see e.g., [Stewart 2001]. Typically, contact problems are viewed as constrained optimization problems, with per-element-pair constraints. In particular, contacts between deformable objects that we consider in this paper, are defined as constraints between surface primitives (triangles, edges and vertices); some examples include [Belytschko et al. 2000; Bridson et al. 2002; Otaduy et al. 2009; Harmon et al. 2009; Verschoor and Jalba 2019]. Penalty-based methods for handling these constraints are among the oldest, but were largely supplanted by constraint formulations and LCP (linear complementarity)-based or SQP solution methods. [Harmon et al. 2009] developed a method for which progressively high penalties are applied as the distance decreases, growing arbitrarily large as the distances to the object decreases. A recent work [Geilinger et al. 2020a] provides a differentiable method for solving dynamic problems with contact and Coulomb friction using a penalty based model combined with equality constraints for static friction, which can be used with gradient-based optimization. Also dealing with Coulomb Friction, [Ding and Schroeder 2020] proposes a penalty based solution that can be used coupling rigid to deformable bodies and material point method (MPM). Another recent method [Li et al. 2020b] shares some aspects with the approach we use, specifically, smoothed version of constraints is used, in particular for friction. However, we use finite penalties, rather than infinite barriers for constraints, and add smoothing both to contact and friction formulations.

It is well-known (see, e.g., [Moreau 1973]) that friction introduces significant non-smoothness to solutions, and due to their dissipative nature, solutions of problems with friction cannot be obtained using energy minimization. A variety of iterative methods for solving resulting problem were developed [Jean and Moreau 1992; Alart and Curnier 1991; Daviet et al. 2011]. Nonsmooth solution methods were applied in recent work with some success [Macklin et al. 2019], but are

very difficult to use in the optimization context. Smoothed version of friction were proposed both in simulation and optimization context in the works we have mentioned.

Contact-based Assemblies. It is also important to mention works designing contact-based assemblies using geometric techniques and not necessarily relying on elasticity simulations to achieve their goal. Works like [Vouga et al. 2012; Panozzo et al. 2013; Wang et al. 2019] consider stability for self-supporting surfaces. Other works [Sun and Zheng 2015; Ureta et al. 2016; Yao et al. 2017] consider the design of joints for more general objects, including furniture.

2.3 PROBLEM FORMULATION

2.3.1 OVERVIEW

We start with a high-level overview of the general problem we are solving. The input to our algorithm is a collection of 3D meshed objects, some of which may be in contact (the meshes will be updated in the process of optimization).

We solve the problem of the general form

$$\min_{\Omega(p)} J(\bar{u}), \quad \text{s.t.}, F(u) = 0, B(u) \leq B^0 \quad (2.1)$$

The main components of this formulation include the following.

- The *unknowns* in the optimization are shape parameters p , in our case displacements of mesh vertices on the boundary, defining the domain $\Omega(p)$, and displacements u of the points of $\Omega(p)$ resulting from elastic deformation with contact. (We use u for the continuous solution of the elasticity problem, and \bar{u} for the vector of displacements of vertices of a meshed $\Omega(p)$). Note that while u corresponds to the deformations of the shape $\Omega(p)$ in our

simulation, displacements p (on $\partial\Omega(0)$) define how the *rest-state* object shape is changed by optimization.

- The *PDE constraint* $F(\bar{u}) = 0$ is a FEM-discretized elasticity equation for \bar{u} on $\Omega(p)$ with contact and friction, leading to a non-linear system of equations in \bar{u} .
- The *objectives* in our optimization are of the one of two forms below, as we have objectives with computation on the whole object domain (e.g., stress) or only on its boundary (e.g., disassembly objective)

$$J(u) = \int_{\Omega_u} e(u, x) dx, \text{ or } E(u) = \int_{\Gamma_u^o} e(u, x) ds \quad (2.2)$$

where both the domain of integration and the integrands depend on the unknowns, and e represents a pointwise measure (e.g., stress), computed as a function of FEM solution defined by the discrete vector \bar{u} . Each objective can be either an *optimization target* to minimize as in (2.1) or a part of an inequality constraint (upper bound on an objective) described below.

As optimization targets, we consider L^p -norms of stress, yielding compliance for $p = 2$ and a close approximation of the maximum stress for large p and volume, of a part or the whole assembly. Other objectives are used primarily as constraints.

- The *inequality constraints*, critical for our formulation, are of the form $B(\bar{u}) \leq B^0$, where $B(\bar{u})$ is an integral of the type (2.2). L^p norm of stress (effectively allowing to bound maximal stress) and volume objectives can also be used as constraints, e.g., we can impose a bound on the maximum volume allowed.

Other objectives used primarily in constraints are the *assembly constraint*, ensuring that two parts can be assembled together; *disassembly constraint*, ensuring that once assembled, the object does not fall apart; and parallel alignment constraint, which makes it more difficult

to disassemble the object using any direction other than the disassembly one.

Shape derivatives. The key element of the optimization process is computing objective and constraint gradients with respect to \bar{u} , required for any first- or second-order optimization algorithm.

There are two main approaches to this problem: one can discretize the problem first, fixing the mesh for $\Omega(p)$ for all p , and a FEM basis. This converts the problem into a finite-dimensional nonlinear algebraic problem, and then compute gradients of the objectives and constraints with respect to p , the positions of boundary mesh vertices. The alternative is the "differentiate-first" approach. Specifically, for each objective and constraint $J(u)$, *before it is discretized*, we derive its *shape derivative*, a continuous analog of the gradient with respect to \bar{u} . The shape derivative is a functional $dJ[v]$, where v is a velocity field of the deformations of the domain $\Omega(p)$ (i.e., the displaced position of a point q is $q + tv$). $dJ[v]$ yields the rate of change of J as $t \rightarrow 0$. dJ is defined by a function ρ on Ω , which can be obtained by solving a *PDE*, similar in structure to the elasticity equation (*the adjoint equation*). The advantage of the latter approach is that it naturally allows for remeshing and refinement (the adjoint equation solver is just a standard elasticity solver), which is essential for large changes in the object shape: any method used to solve large-deformation elasticity PDEs can be used to compute ρ , without fixing the discretization in advance.

The cost of each optimization step is approximately equal to solving two elasticity problems, the nonlinear elasticity problem, and the linear adjoint problem, to obtain the solution and shape derivative respectively.

In the rest of the section, we provide more specifics on the components of the formulation described above and their discretizations. The complete derivations are included in the supplementary material.

2.3.2 NOTATION

We use the following notation, also illustrated in Figure 2.2

- Ω : optimization domain (may consist of multiple objects)
- $\Gamma_D, \Gamma_N, \Gamma_C, \Gamma_S$: parts of the domain boundary where Dirichlet, Neumann, rigid-deformable (C) and deformable-deformable (S) contact boundary conditions are applied.
- superscripts $+, -$ denote surface quantities on two sides of a contact surface between two deformable objects; $[\cdot]$ denotes the difference of two one-sided quantities.
- $u(x), x \in \Omega$: displacements.
- $D[f]$ for a function f , its material derivative $\frac{\partial f}{\partial t} + u \cdot \nabla f$, where t is a deformation parameter.
- $\sigma, \varepsilon = \varepsilon(u) = \frac{1}{2}(\nabla u + \nabla^T u)$: stress and linearized elasticity strain.
- C : elasticity tensor.
- T : tractions on Γ_N , i.e., surface force density on the boundary.
- $1/\alpha$: penalty parameter for constraints;
- $\rho(x)$: solution of the adjoint PDE used to compute shape derivatives.
- ϕ_i, ψ_i : volumetric finite element basis functions at node i .
- λ_m : basis functions on the boundary.
- p : optimization degrees of freedom (boundary vertices of Ω)

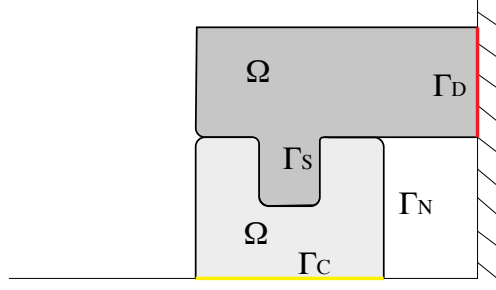


Figure 2.2: Notation for boundary parts for a simple assembly: Γ_D is attached to the vertical wall (red), Γ_C can slide along the ground (yellow), Γ_N is where the load is applied, and Γ_S is the contact area of two parts. We view both parts as a single domain Ω consisting of detached parts.

2.3.3 ELASTIC DEFORMATIONS WITH CONTACT

In this paper, we only consider static problems, so there is no time dependence in the equations we use.

The basic form of static equations of linearized elasticity (without contact or friction yet) is

$$\begin{aligned}
 \nabla \cdot \sigma &= 0 & \text{on } \Omega \\
 u &= \hat{u} & \text{on } \Gamma_D \\
 \sigma n &= g & \text{on } \Gamma_N
 \end{aligned} \tag{2.3}$$

The variational form of this problem, that we rely on to formulate the contact constraints in a computationally practical way, as well as for the finite element discretization, is given by the equation

$$\int_{\Omega} \epsilon(u) : C : \epsilon(w) dV - \int_{\Gamma_N} w \cdot g dS = 0 \tag{2.4}$$

satisfied for any w in an appropriate function space with $w = 0$ on Γ_D , and u satisfying the Dirichlet boundary conditions on Γ_D .

Contact constraints. We consider two forms of contact constraints, *rigid-deformable* (RD) and *deformable-deformable* (DD), defined on the parts of the boundary Γ_C and Γ_S respectively.

The former type of constraints involves a fixed boundary to which we refer as *obstacle*, on one side, which can be considered rigid, e.g., legs in contact with the floor.

$$\begin{aligned} u \cdot n &\leq 0 && \text{on } \Gamma_C \\ \sigma n \cdot n &\leq 0 && \text{on } \Gamma_C \\ (\sigma n \cdot n)(u \cdot n) &= 0 && \text{on } \Gamma_C \end{aligned} \tag{2.5}$$

The first equation on Γ_C says that the displacement should move the object points away from the obstacle; the second equation says that the normal force on this boundary should be 0 or point towards the other object; and finally, the last equation, the complementarity condition, ensures that if the force is nonzero, the displacement in the normal direction is zero, i.e., there is contact. The DD contact constraints are similar, however, because we have deformable material on both sides of the boundary, and the differences in stress need to be considered. Normals n^- and n^+ and displacements u^- and u^+ correspond to the parts on the opposite sides of contact, and $[u]$ is the jump $u^- - u^+$.

$$\begin{aligned} [u] \cdot n^- &\leq 0 && \text{on } \Gamma_S \\ [\sigma] n^- \cdot n^- &\leq 0 && \text{on } \Gamma_S \\ ([\sigma] n^- \cdot n^-)([u] \cdot n^-) &= 0 && \text{on } \Gamma_S \end{aligned} \tag{2.6}$$

Friction constraints. Similarly, there is a set of equations for friction:

$$\begin{aligned}
\|(\sigma n)_t\| &\leq \mu |(\sigma n)_n| \text{ on } \Gamma_C \\
\|(\sigma n)_t\| &< \mu |(\sigma n)_n| \implies u_t = 0 \text{ on } \Gamma_C \\
\|(\sigma n)_t\| &= \mu |(\sigma n)_n| \implies u_t = -\lambda(\sigma n)_t
\end{aligned} \tag{2.7}$$

where $\lambda > 0$, and μ is the friction coefficient; t and n refer to tangential and normal components of the force. The first inequality captures the main aspect of Coulomb static friction model (the force is bounded by μ times the normal force). The second equation states that if the force is below maximal, no displacement happens, and the third one that the displacement at maximal friction force is parallel to it and opposite in direction.

A modification (2.6) (replacing one-sided constraints with differences), applies to friction (please see the supplementary document), yielding *deformable-deformable* contact.

2.3.4 VARIATIONAL FORM AND CONSTRAINT APPROXIMATION

While the basic elasticity problem (2.4) is quite straightforward to solve and well-understood, contact constraints, especially friction constraints, result in numerous difficulties: (1) the system becomes highly nonlinear and non-convex; (2) the solutions may be non-smooth; (3) due to dissipative nature of friction, the problem cannot be cast as a (constrained) energy minimization problem. This makes even the direct solution of the problem difficult to make robust, and presents a particular challenge for shape optimization.

The key mathematical ideas for resolving these difficulties can be found in [Eck et al. 2005], further developed in [Maury et al. 2017]. (Recent work [Li et al. 2020b] also follows a related approach for dynamic deformable contact).

The main elements of the approach include:

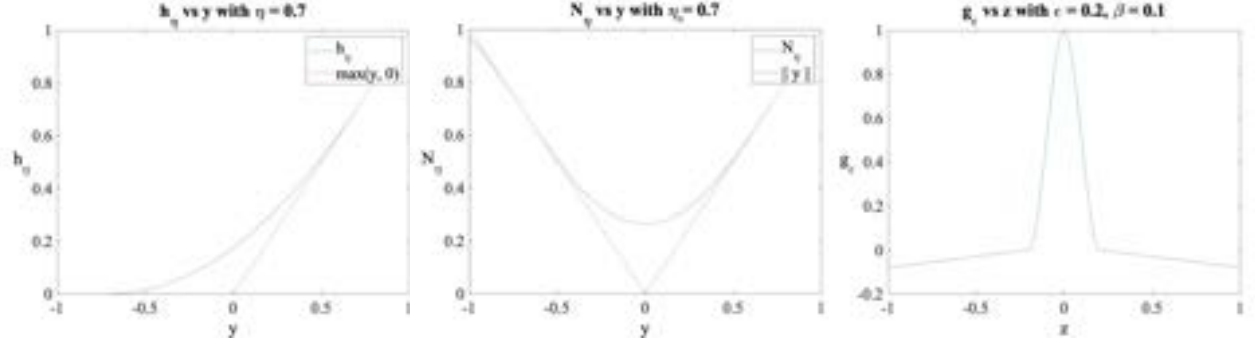


Figure 2.3: Smoothing function N_η , h_η , and g_ϵ

- Use *smooth* approximations of contact with friction (these are used to prove solution existence in [Eck et al. 2005], but are also valuable computationally). In particular, the norms in (2.7), that result in non-smoothness of the constraints are replaced by a smoothed norm that will be described in detail in this section.
- Replace the inequality constraints by suitable (also smooth) penalty functions, added to the variational formulation of the problem (2.4); a number of works (see [Maury et al. 2017] for discussion) show that the results converge to the true solution, as the penalty weight $1/\alpha \rightarrow \infty$. Like penalty approaches commonly used in graphics, using these methods has the benefit of replacing constrained optimization with unconstrained, and the downside of potential constraint violations which we discuss below.

We largely follow the smooth formulation of [Eck et al. 2005], which demonstrates that the solution to the smooth problem exists under typical assumptions, and with smoothing parameters approaching zero, converges to a solution of the original problem.

Smoothed contact functional. We introduce a smoothing function [Eck et al. 2005]

$$h_\eta(y) = \begin{cases} 0 & y \leq -\eta \\ \frac{1}{4\eta}y^2 + \frac{1}{2}y + \frac{\eta}{4} & -\eta \leq y \leq \eta \\ y & y \geq \eta \end{cases} \quad (2.8)$$

approximating $\max(0, y)$ as $\eta \rightarrow 0$. This is the same function used in [Maury et al. 2017].

The contact equations (2.5) lead to the following objective

$$j'_{N,\alpha}(u, w) = \frac{1}{\alpha} \int_{\Gamma_C} h_\eta(u \cdot n) w \cdot n \, dS \quad (2.9)$$

As $\alpha \rightarrow 0$, the solutions obtained with this term added to (2.4) converge to solutions satisfying the constraint $u \cdot n \leq 0$. Similarly, we have

$$j'_{S,\alpha}(u, w) = \frac{1}{\alpha} \int_{\Gamma_S} h_\eta([u] \cdot n^-) [w] \cdot n^- \, dS \quad (2.10)$$

for DD contact.

Friction functionals. The Coulomb friction constraints (2.7) and the analogous constraint for DD friction, are expressed in terms of norms. For the smoothed functional, we use the following function to approximate $|y|$, as $\eta_n \rightarrow 0$:

$$N_\eta(y) = \begin{cases} \|y\| & \|y\| \geq \eta_n \\ -\frac{1}{8\eta^3}\|y\|^4 + \frac{3}{4\eta}\|y\|^2 + \frac{3}{8}\eta & \|y\| \leq \eta_n \end{cases} \quad (2.11)$$

Then friction constraints are captured by the following objectives [Eck et al. 2005]:

$$\begin{aligned} j'_{CF,\alpha}(u, w) &= \frac{\mu}{\alpha} \int_{\Gamma_C} h_\eta(u \cdot n) N'_\eta(u_t) \cdot w_t \, dS \\ j'_{SF,\alpha}(u, w) &= \frac{\mu}{\alpha} \int_{\Gamma_S} h_\eta([u] \cdot n^-) N'_\eta([u]_t) \cdot [w]_t \, dS \end{aligned} \quad (2.12)$$

Smoothed elasticity with contact and friction. Four objectives (2.9)- (2.12) are added to (2.3) to model contact with friction both for deformable-deformable and deformable-rigid contacts on parts of boundaries.

$$\begin{aligned} &\int_{\Omega} \epsilon(u) : C : \epsilon(w) \, dV - \int_{\Gamma_N} w \cdot g \, dS \\ &+ j'_{N,\alpha}(u, w) + j'_{S,\alpha}(u, w) + j'_{CF,\alpha}(u, w) + j'_{SF,\alpha}(u, w) = 0 \end{aligned} \quad (2.13)$$

for any w . The finite element discretization of these equations is standard and leads to a non-linear system of equations, because of the nonlinear functions. This system requires computing a Jacobian of the left-hand-side to solve efficiently. We refer to the supplementary document for the derivation of the Jacobian.

2.3.5 OPTIMIZATION OBJECTIVES

Next, we describe the optimization problem we solve, specifically, the set of objectives we use in the functional J as optimization targets or inequality constraints, and how their shape derivatives are computed.

In our model, we consider only the simplest type of assembly, namely, moving parts together in a specific direction y . This can be generalized to nonlinear trajectory settings (e.g., a screw motion trajectory) but this will make these constraints substantially more complex. L^p **stress and**

volume. These objectives are standard, and expressed as

$$J_\sigma^p(\Omega, u) = \left(\int_{\Omega} (\|\sigma(u)\|_F)^p dV \right)^{1/p}; \quad J_V(\Omega) = \left(\int_{\Omega} 1 dV \right)^2 \quad (2.14)$$

The stress objective for large p approximates the non-smooth maximum-stress functional well. Following [Panetta et al. 2017c], we consider the average stress at each element. Also, we use value $p = 20$ in most cases. We note that von Mises stress can be used just as easily. In addition, we have options for setting target stress and target volume, which are expressed as follows:

$$J_\sigma^t(\Omega, u) = \int_{\Omega} \varphi(\|\sigma[u]\|_F - S_t) dV; \quad J_V^t(\Omega) = \varphi\left(\int_{\Omega} dV - V_t\right) \quad (2.15)$$

where S_t and V_t represent the stress and volume targets and φ is the following function

$$\varphi(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z^2 & \text{otherwise} \end{cases} \quad (2.16)$$

We normalize our objective approximating maximal stress by the ratio between the load (in our boundary conditions) and the area of our shape's surface, which has units of stress. For the volume objective, we use $1/V_t^2$ as a normalization constant.

We assume an disassembly direction y per connection, with assembly happening in opposite direction $-y$. We define two objectives, an assembly objective, which ensures that parts can be put together without much deformation in direction $-y$, and dissassembly objective, which penalizes parts moving apart along y , under the loads specified by the user.

Assembly objective. A fully physical treatment of assembly would require simulating the assembly process, leading to a large number of nonlinear solves for different time steps for a

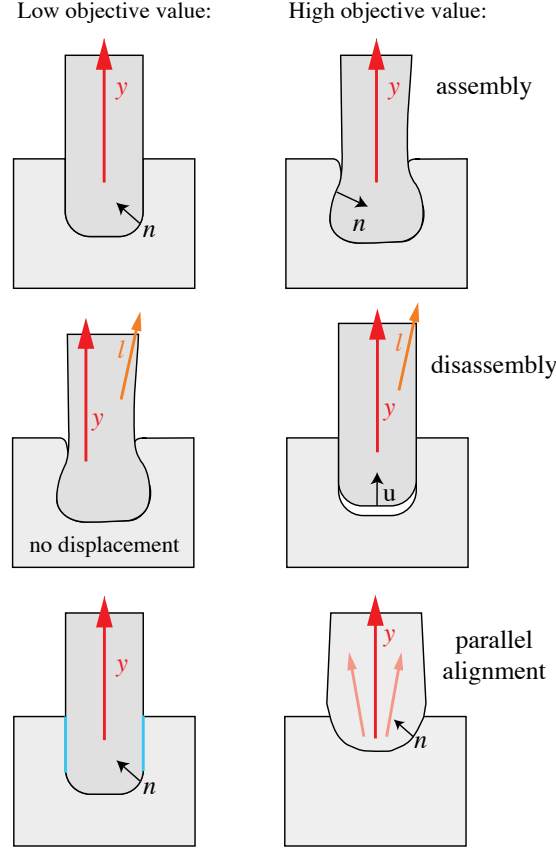


Figure 2.4: Assembly, disassembly and parallel alignment objectives. Top: left shape is easy to assemble in $-y$ direction, while shape on the right cannot be assembled without effort (high objective value, due to normals pointing in the direction opposite to y). Middle: For the same disassembly direction y , and user-defined load l , the assembly on the right falls apart under the load; for the shape on the left, the parts cannot move apart in direction y under load l . Bottom: left shape has a large region parallel to disassembly direction (marked in blue), making it hard to disassemble in any direction other than y , while shape on the right allows for multiple easy disassembly directions (shown in light red).

single gradient evaluation. Instead, we use a geometric heuristic with negligible cost to allow parts in contact to be assembled without a large deformation while moving a part along an a priori fixed direction in space. Assembly is ensured if the contact surface is a height field when we view y as a vertical direction. The angle between the direction of connection/disconnection and surface normal needs to be less than $\pi/2$. In our experiments, we always start with assemblable non-optimized parts, in order to facilitate the optimization.

This is not a hard constraint for deformable objects, as it can be violated by a small amount, to squeeze in a connector. We do not attempt to bound maximal possible deformation, but empirically it can be easily controlled by increasing the weight of this objective.

$$J^a(\Omega) = \frac{1}{|\Gamma_S|} \int_{\Gamma_S} \varphi(-n(x) \cdot y) ds \quad (2.17)$$

where y is the disassembly direction vector (and $-y$ the assembly direction), $n(x)$ is the normal at the contact position x , and $|\Gamma_S|$ is the area $A_S = \int_{\Gamma_S} 1 ds$.

Disassembly objective. This objective ensures that under user-defined loads, or for a collection of different user-defined loads, the parts should not move away from the assembled positions. We formalize this by requiring that the optimized shape in equilibrium does not move in the disassembly direction, i.e., the forces holding contact points together are sufficient.

$$J^d(\Omega, u) = \frac{1}{|\Gamma_S|} \int_{\Gamma_S} \varphi(u(x) \cdot y - u_{tol}) ds \quad (2.18)$$

A tolerance u_{tol} is added, allowing a small displacement in the assembly direction. While the assembly term is purely geometric, the disassembly objective relies on actual simulation responses and it bounds how much optimized parts can move. As long as it is finite, the assembly does not fall apart.

Although the last two objective terms may work against each other in the cases when the loads have a significant component along the disassembly direction, in other situations they work together to provide structures that are, at the same time, both assemblable and are not prone to accidental disassembly under user-defined load. See Figure 2.33 for an example.

Parallel alignment objective. Finally, this objective is a secondary heuristic that is not essential

for solving the problems but empirically makes the structure more robust with respect to being disassembled by forces close to assembly direction, since it favors solutions with larger contact zones tangential to the assembly direction. When such zones are present, forces deviating from the disassembly direction have normal components resulting in deformations and friction keeping the parts from accidental disassembly. In the bottom of Figure 2.4, we can see how the shape without parallel alignment allows for easy disassembly in different directions (shown in red), without any reaction force (normal or friction), while the shape on the left allows for a single disassembly direction y .

This term is defined per continuous contact area Γ_S with respect to a disassembly direction y :

$$J^p(\Omega) = \varphi \left(W_{target} - \frac{1}{|\Gamma_S|} \int_{\Gamma_S} g_\epsilon(n(x) \cdot y) ds \right) \quad (2.19)$$

where W_{target} is the target percentage area in the connection which should be parallel to the disassembly direction, which can help increasing/decreasing reaction forces avoiding disassembly. The function g is parameterized by ϵ , defining the interval for which g is positive and β setting the linear decay of the function when $abs(z)$ is larger than ϵ . It has the following form:

$$g(z) = \begin{cases} \beta(z + \epsilon) & \text{if } z \leq -\epsilon \\ -az^3 + bz^2 + 1 & \text{if } z \leq 0 \\ az^3 + bz^2 + 1 & \text{if } z \leq \epsilon \\ -\beta(z - \epsilon) & \text{otherwise} \end{cases} \quad (2.20)$$

where ϵ is a smoothing constant and where a and b are defined as:

$$a = -\beta/\epsilon^2 + 2/\epsilon^3 \text{ and } b = -a\epsilon - 1/\epsilon^2$$

Summary. To summarize, our solver solves problems of the form (2.1), for which

$$J(u) = \sum_i k_i J_i, \quad (2.21)$$

where each of J_i is one of the objectives, enumerated above, combined with weights. The equality constraint $F(u) = 0$ is given by (2.13), the static elasticity equation with smoothed contact and friction.

The inequality constraints are of the form $J_m < J_m^{bnd}$ where J_m are some of the remaining objectives for which upper bounds are imposed (e.g., a bound on L^p norm of stress). These are imposed as soft constraints by adding $\sum_i \frac{1}{\epsilon_i} \phi(J_m^{bnd} - J_i)$ to $J(u)$.

2.4 SHAPE DERIVATIVES AND DISCRETIZATION

In this section, we describe our approach to computing the discretization of the problem and the gradient of the functional projected to the constraint space $F(u) = 0$ (i.e., space of displacements satisfying the elasticity equation with contact and friction).

2.4.1 SHAPE DERIVATIVES

For each objective, regardless of it is used as an optimization target, we need to compute the gradient with respect to p . As explained in Section 2.3.1, we use shape derivatives, which are computed using the solution of an adjoint equation. More specifically, the solution is used to construct a form dJ_d on the surface, which, when applied to changes δp of vertex positions on the boundary, produces the (linearized) change of the functional J :

$$dJ[\delta p] = \lim_{t \rightarrow 0} \frac{J(\Omega(p + t\delta p), u_t) - J(\Omega(p), u_0)}{t} \quad (2.22)$$

where u_t is the solution of the elasticity equations obtained on the domain with deformation $t\delta p$. The theory of shape derivatives is well-established, and we refer to [Bonnetier and Dapogny 2020].

Here, we only present a brief summary of the steps for computing these derivatives. A detailed derivation can be found in the supplementary material. Our formulation follows [Panetta et al. 2017c], in that we consider volume, rather than surface integrals, which proved to be critical for derivative accuracy, at a moderate additional cost, compared to the elasticity solves involved.

Adjoint equation. The first step in computing dJ is to solve the *adjoint equation*, which in our case has the following form:

$$\int_{\Omega} \epsilon(\rho) : C : \epsilon(\psi) dV = \int_{\Omega} \tau : \epsilon(\psi) dS - (T'_C + T'_S + T'_{CF} + T'_{SF}), \quad (2.23)$$

for any ψ , where the terms T'_X , corresponding to the contact and friction smoothed penalties are of the form $\int_{\Gamma_X} F[u] \cdot \psi dS$, and ρ is the unknown. We observe that this is a *linear* elasticity equation which is relatively inexpensive to solve compared to the primal nonlinear elasticity equation. The quantity τ is defined as the sum of $2e'\sigma : C$ for all objectives of the form (2.2) where $e(s(\bar{u}), x)$ is a function on a stress measure, as our case. Then, e' means the partial derivative of e with respect to our stress measure.

Shape derivative. If the boundary deformation is defined by a set of basis functions λ_m , with the deformation expressed as $v = \sum_m \delta p_m \lambda_m$, where δp is the vector of changes of our variables, it is possible to express the shape derivative as a dot product

$$dJ \left[\sum_m \delta p_m \lambda_m \right] = S[u, \rho] \cdot \delta p \quad (2.24)$$

where $S[u, \rho]$ is a vector of the same length as p of vector valued functions depending on u and ρ . The expressions for these functions are included in the appendix, and derived in the supplementary

document. We emphasize that no discretization, other than discretization of the deformation of the boundary was performed so far.

The vector $S[u, \rho]$ computed by numerical integration of FEM solutions of two elasticity problems for ρ and u is the gradient of the functional J with respect to p ; this is what we use in the optimization process as discussed in Section 2.5.

2.4.2 DISCRETIZATION

To make our discussion of the formulation complete, we summarize the discrete form of the problems we solve. The exact expressions are straightforward to derive but tedious (please see the supplementary material). We use quadratic Lagrangian elements on tetrahedras for discretization of (2.13) to obtain a system of the form

$$F(\bar{u}) = K\bar{u} - F_N + N_C(\bar{u}) + N_S(\bar{u}) + F_C(\bar{u}) + F_S\bar{u} = 0 \quad (2.25)$$

The first two terms come from the standard elasticity equation, and the rest correspond to the remaining terms in (2.13). The remaining terms, while simple, are not linear, so the system after discretization is a general algebraic system, and requires a non-linear solver. We also derive expressions for the Jacobians DN and DF , which are needed for efficient optimization.

In contrast, the adjoint PDE (2.23) is linear and has the form

$$(K + DN_C + DN_S + DF_C + DF_S) \bar{\rho} = D \quad (2.26)$$

i.e. involves exactly the Jacobians of the constraint functions. The right-hand side D is expressed in terms of τ . Formulas for the entries of these matrices are provided in the supplementary document.

Finally, once \bar{u} and $\bar{\rho}$ are available, the coefficients of the shape derivative are computed from these

values by integration, following formulas in the appendix, using numerical quadrature described in Section 2.5.

2.5 METHOD

Input. The input to our algorithm is a collection of tetrahedral meshes, with some parts of their surfaces in contact, and with boundary conditions applied to other parts of surfaces. Each object can be designated as deformable or rigid.

Our overall method is conceptually straightforward (Algorithm 1). The functions used in the pseudocode are:

- ELASTICITYSOLVE solves the nonlinear elasticity system (2.3), to obtain \bar{u} ;
- ADJOINTSOLVE solves the adjoint system (2.26) to obtain ρ ;
- DISCRETESHAPEDERIVATIVE, given \bar{u} and ρ , computes the gradient $S[u, \rho]$ with respect to vertex positions on the boundary using (2.24).
- CONVERGED is the outer iteration stopping criterion discussed below.

The inner loop works on a fixed connectivity for $\Omega(p)$, and is close to the standard BFGS algorithm: at each step, a descent direction is computed, and a line search is performed to determine our step size. There are three important differences: (1) we check for any inversions of tetrahedra and choose a step that maintains a bound on mesh element shape quality; (2) after each update of the boundary vertices, we call the SLIM smoothing algorithm [Rabinovich et al. 2017a] on interior vertex positions p^{int} with boundary vertices p fixed, to move the interior vertices so that the quality of the mesh is improved; (3) with a valid mesh, we run simulation and if it doesn't converge in a maximum number of iterations, we reduce step by half in the line search.

If the step becomes too small, the inner loop is terminated, and the domain is remeshed in the outer loop.

A natural stopping criterion for the algorithm consists of three parts: (1) the objective reduction obtained in a step of (outer) iteration is below a threshold ϵ_r computed relative to the initial objective value; (2) the step size of the line search falls below a threshold (3) the maximal number of iterations max_oi is exceeded. Due to remeshing, the energy however may oscillate slightly, and for robust behavior we require that sufficient energy decrease does not happen over m steps. While many other options are possible (e.g., relative or absolute gradient norm threshold), we consider the rate of change in the objective to be most appropriate: our goal is to obtain a reduction in the objective, and a slow rate of reduction indicates that optimization will not improve the target by much more in a reasonable number of iterations; this happens either due to being close to a local minimum value or step size going to zero for geometric reasons, typically thin regions, resulting in distorted elements.

We discuss the choices for ϵ_r and m in Section 2.6, as well as examine convergence for specific test cases.

For remeshing, we use Triangle in 2D, and CGAL and fTetWild [Hu et al. 2020a] for 3D tetrahedral meshing. More information on meshing is presented in the Appendix 2.9.5.

Smoothness penalty. In addition to all optimization targets described in Section 2.3, we use a discrete regularization term $S_p(\Omega) = \sum_{v \in V} \|s(v)\|^p$ where

$$s(v) = \frac{\sum_{u \in N_v} (u - v)}{\sum_{u \in N_v} \|u - v\|}$$

and N_v is the neighborhood of v . The value of power p can be adjusted to obtain smoother surfaces at the cost of less optimal shapes; we use value 2 for most cases, increasing it to 4 for some objects.

Algorithm 1 contact optimization

```
function GRADIENT( $p$ )
   $u \leftarrow \text{ELASTICITYSOLVE}(\Omega(p))$ 
   $\rho \leftarrow \text{ADJOINTSOLVE}(\Omega(p), u)$ 
   $g \leftarrow \text{DISCRETESHAPEDERIVATIVE}(u, \rho)$ 
end function

function SHAPEOPTIMIZATION
   $p \leftarrow$  non-fixed boundary vertex positions
  repeat
    repeat
       $d \leftarrow \text{BFGSDIRECTION}(\text{GRADIENT}, p)$ 
       $\sigma, s, g \leftarrow \text{LINESEARCH}(\text{GRADIENT}, d)$ 
       $p \leftarrow p + sd$ 
       $\text{SLIM}(p^{int})$ 
       $ii \leftarrow ii + 1$ 
    until  $s < \delta$  or  $ii = oi\_max\_ii$ 
     $\text{REMESH}(\Omega(p))$ 
     $oi = oi + 1$ 
  until  $oi = max\_oi$  or not CONVERGED
end function
```

This term is scale-invariant and pushes the triangles/tetrahedra of the mesh toward equilateral.

Elasticity solver. In ELASTICITYSOLVE we use the standard Newton’s method with line search to solve the nonlinear elasticity system with contact. We consider the simulation solved when residual is lower than a given tolerance. For 2D, we use 10^{-10} for the tolerance, while the value of 10^{-8} is used for our 3D examples. Moreover, for solving linear systems at each iteration of the Newton’s method, we use CHOLMOD (for frictionless scenarios) and UMFPACK (when friction is present) in 2D. In 3D, we use MKL Pardiso library.

Adaptive Quadrature. An extremely important aspect of our implementation is the quadrature used to compute the integrals in the FEM system discretization, as well as in the shape derivative coefficient formulas. For problems with friction, precise computation of these integrals proved to be very important. At the same time, due to functions like h_η , the functions we integrate, while

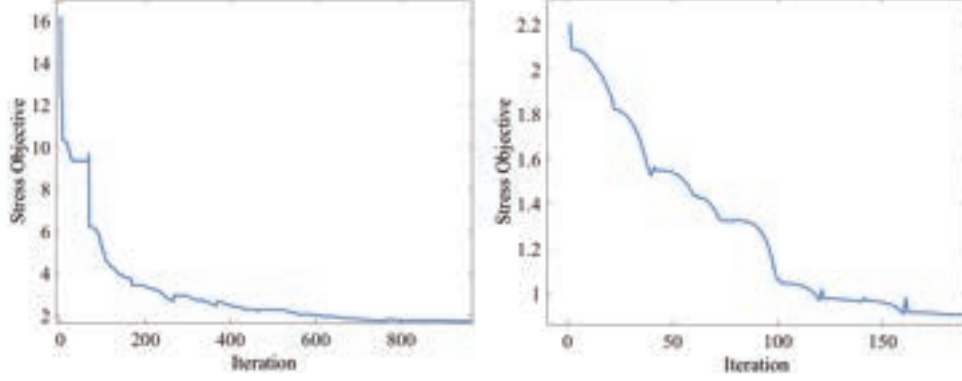


Figure 2.5: Convergence plots for connector (Figure 2.9) on the left and stool (Figure 2.24) on the right side. The charts show stress objective vs iterations.

smooth, have higher derivative discontinuities. We use a combination of adaptive refinement on triangles and high-order (order equals to 10) Gaussian quadrature to integrate all functions to high precision. For more information, see Section 2.9.4 in the Appendix.

Convergence behavior. Figure 2.5 shows the objective as a function of iteration number for a 2D connector (Figure 2.9) and a 3D stool (Figure 2.24). For both cases, we optimized L_p norm of stress, while keeping volume below or equal the initial value. For the connector, during the last 200 iterations, the energy decreased only by 0.87%, while for the stool, the minimum decreased by 0.27% in the last 30 iterations. We provide additional data in Section 2.6.

Observe that there are increases in energy at some of our iterations, corresponding to the remeshing of the current solution. However, after that the objective function quickly decreases, which motivates our formulation of the convergence criterion.

2.6 EVALUATION

Simulation Validation. As we use approximations to the standard physical models of contact and friction we evaluate the accuracy of these models. We use two examples. The first one is box in contact with a rigid surface. The second one is joint in contact with a holder as shown in Figure

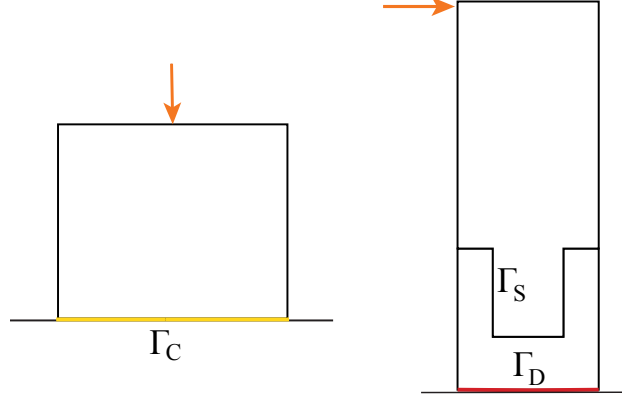


Figure 2.6: Examples used for validation. The left: box in contact with a rigid surface. The right: joint in contact with a holder. We use the same notations for boundary parts as in Fig. 2.2. Starts from this Figure, all the orange arrows in the following figures illustrate the direction of load.

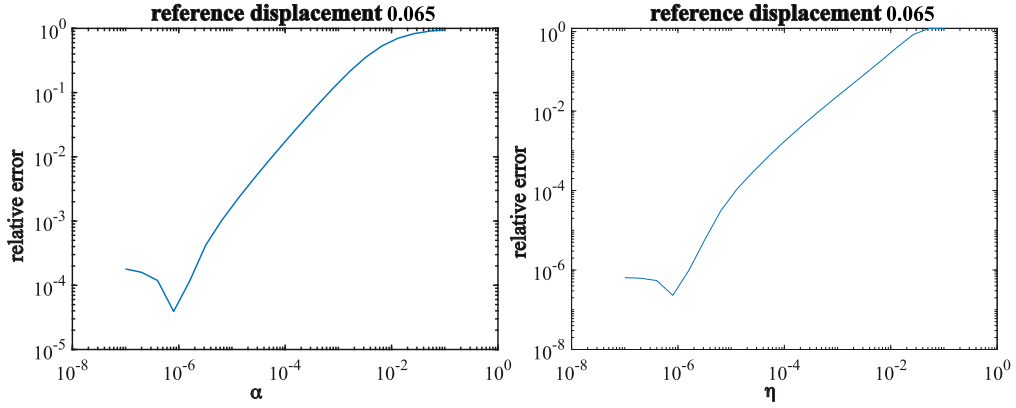


Figure 2.7: Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing α . The right is relative displacement error with changing η

2.6.

The results for both cases are largely the same. Thus, we will only show the results for box in contact with a rigid surface in this section and leave out the results for joint in contact with holder. We first change the value of α and η . The reference value we use for comparison is obtained by simulation with $\alpha = \eta = 10^{-6}$. We set the smallest value of α to be 10^{-7} and scale it by the power of 2. The relative value of the difference in relative displacement $|u - u_{ref}|/|u_{ref}|$ is shown in Figure 2.7.

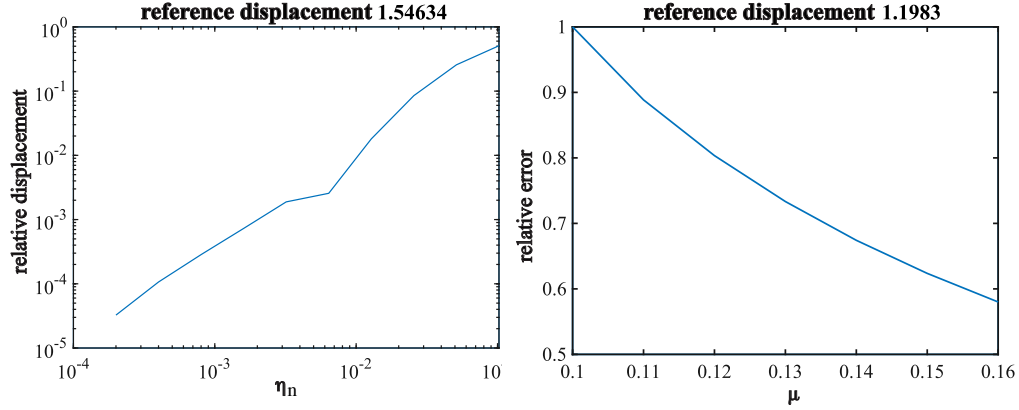


Figure 2.8: Simulation validation for box in contact with rigid surface. The left is relative displacement error with changing η_n . The right is relative displacement with changing μ

Then, we test the same scenarios but with friction at contact regions. First, we consider the dependence of accuracy on η_n , using $\eta_n = 10^{-4}$ as the reference value, as for friction we typically need more smoothing in the constraints. To test how friction coefficient affects simulation, we add in a small horizontal load that is 10% of the vertical load for the box example. Test with μ ranging from 0.1 to 0.16 and increase μ every time by 0.01 for the two examples. We compare the tangential displacement of the examples with respect to different μ to the tangential displacement of the examples with respect to $\mu = 0.1$ and take the ratio. We get the plots shown in Figure 2.8.

Effects of optimization targets and constraints. We use a simple two-dimensional connector example to demonstrate the effects of various optimization objectives and constraints (Figure 2.9) We compare stress distribution and maximal stress in different cases.

For most of the examples related to this scenario, we used a similar optimization configuration, running a maximum of 1000 iterations and remeshing at least every 100 iterations ($max_ii = 100$ and $max_oi = 10$).

The most radical difference is between minimizing the volume while bounding stress and minimizing maximal stress.

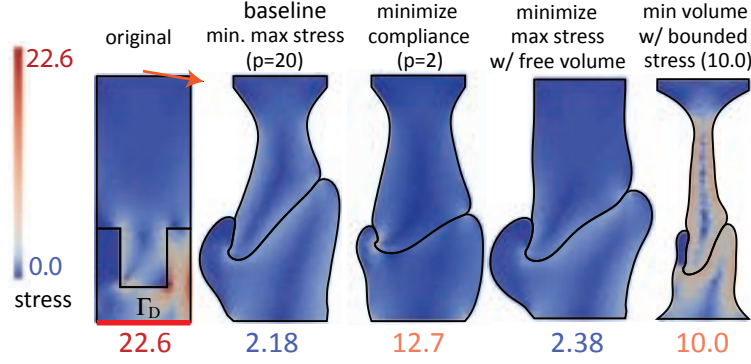


Figure 2.9: Optimization results when using different combinations of objectives. For both baseline ($p = 20$) and compliance ($p = 2$) scenarios, we use a target volume equal to the original one.

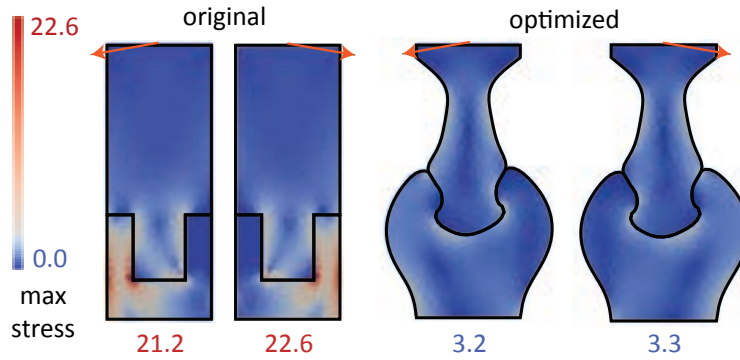


Figure 2.10: Optimization considering multiple loads.

We also compared the scenario of minimizing maximal stress with and without a bound on volume (equal to the initial volume). In this comparison, stress results were actually very close to each other, with a slight advantage to the version with the constraint.

Multiple loads. Our framework allows optimizing for multiple separate acting forces, meaning that the energy related to stress will be a combination of the values from each separate scenario. An example is shown in Figure 2.10, where forces of the same intensity are applied to the right and to the left of the top piece. The stress results of the optimized shape are very close to each other and the final shapes present considerable symmetry, even though we are not enforcing any geometry symmetry through constraints.

Differences in material properties. Figure 2.11 shows how the results of optimization are

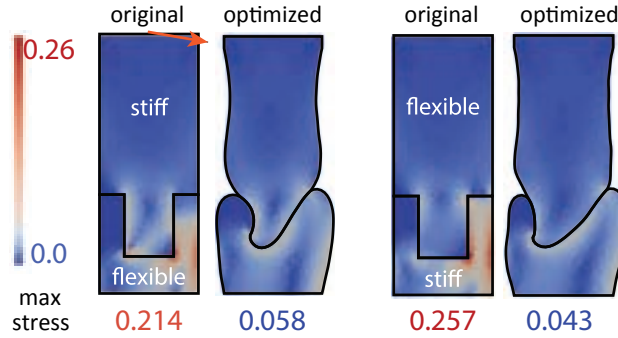


Figure 2.11: Multimaterial optimization. The stiff material has Young’s modulus equal to 100 and Poisson’s ratio of 0.3, while the flexible one has Young’s modulus equal to 1.0 and Poisson’s ratio of 0.0.

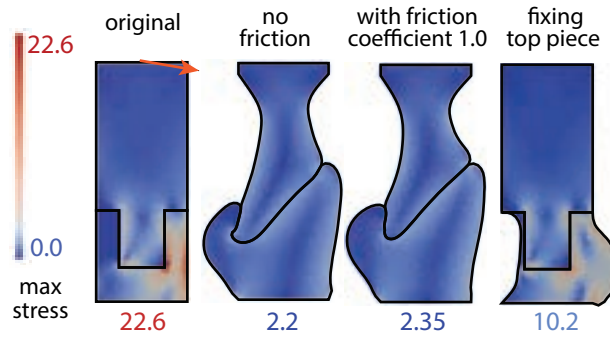


Figure 2.12: Middle shapes: with and without friction; rightmost shape: the optimization of the top assembly is disabled.

affected by combining a highly flexible and a stiff material on our two different pieces. (In both cases the target is the baseline case of minimizing stress with no constraints). We observe that the stress reduction is similar, although shapes required for this are substantially different.

Role of friction. In the next comparison (see Figure 2.12), we observe that in the presence of friction ($\mu = 1.0$), we achieve a similar maximum stress value with the assembly not needing a more extreme protrusion to stay together, due to additional forces resulting from friction. This is demonstrated in Figure 2.12. The rightmost example shows the drastic effect of fixing only one part.

Assembly/disassembly and parallel alignment constraints. We show the effects of our connection-related constraints presented in Section 2.3.5.

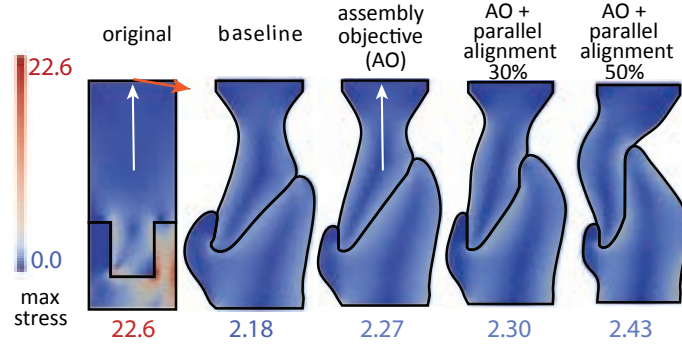


Figure 2.13: Optimization results when using assembly and parallel alignment objectives, with two different strengths. The white arrow demonstrates the disassemble direction.

Figure 2.13 shows the baseline optimization (minimizing maximal stress with a bounded volume) compared to different cases when our assembly constraint in all cases, and different target parallel alignments. Observe that, as expected, the contact area aligned with the disassembly direction increases.

All three examples are possible to assemble in vertical direction without deformation. Also, the parallel alignment objective guarantees that the desired proportion (30% and 50%) of the contact walls are parallel to the input direction.

Figure 2.14 shows the baseline case compared to one using a disassembly constraint with a very small tolerance. Compared to the the baseline optimization, two pieces do not detach when the disassembly constraint is applied. Another example of the importance of this constraint is shown in Figure 2.20.

Assembly/disassembly tradeoff. Finally, we also studied the effect of choosing different balance of assembly and disassembly objectives, in a setting when these counteract each other. Consider the case of Figure 2.15, where a force is pulling the top part of the connector up and we initially optimize the shapes using the same weight (100.0) for both terms and weight 1.0 for stress. Then, we reran our optimization using three lower weights for the assembly term. Figure 2.15 shows that, by reducing assembly’s importance, almost no movement is observed in the last shape where

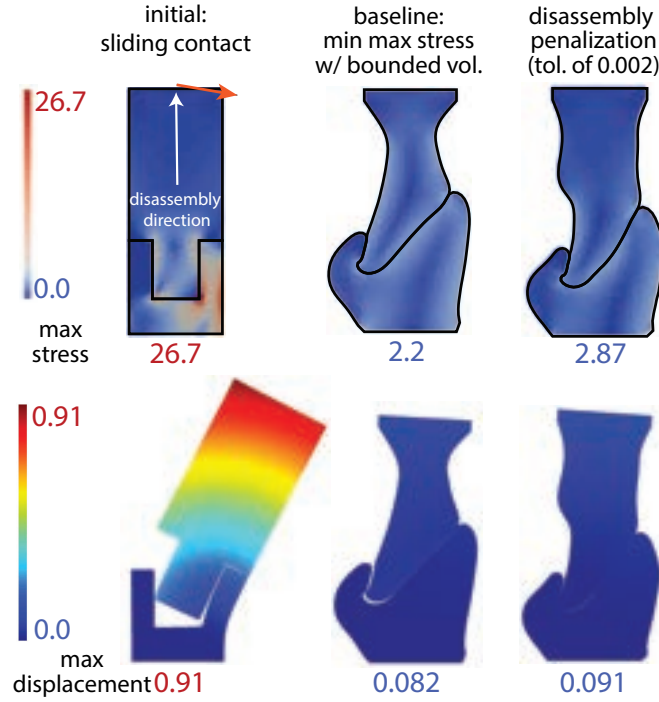


Figure 2.14: Optimization results when using a disassembly energy term with low tolerance. As before, the white arrow demonstrates the dissemble direction.

Table 2.1: Final state of objective term values when running optimization with different assembly weights

Objective	Assembly weight			
	100	10	1	0.1
Assembly	0.02598	0.06968	0.35474	1.34750
Disassembly	0.01302	0.00275	0.00033	0.00003
Stress	2.53228	1.95349	1.35349	0.83647

assembly weight equals 0.1. When the disassembly term is dominant, the optimized shape has small protrusions on both sides that keep it firmly in place, at the expense of greater effort required for (dis)assembling it along the (dis)assembly direction.

Table 2.1 presents a comparison of our main energies in this study. Note how assembly objective values increase when it's weight is reduced, while, at the same time, disassembly and stress values decrease considerably.

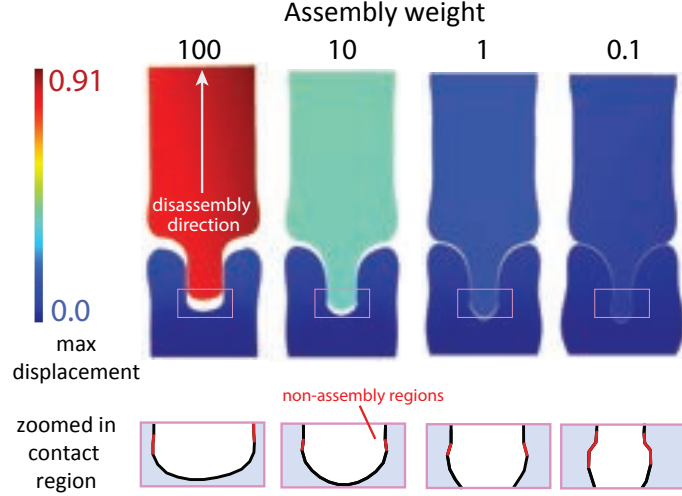


Figure 2.15: Displacement results when optimizing our connector using varying assembly weights and a fixed disassembly weight.

2.7 EXAMPLES

To validate our framework, we computed optimized shapes for a number of 2D and 3D realistic scenarios. In all illustrations in this section, parts of the object boundary with Dirichlet conditions are shown in red, and contact regions with external supports in yellow. We use $p = 20$ for stress optimization in all examples. Except for specific scenarios, in most of our examples we used the same objective function weights. A summary of our experiments is presented in Table 2.2, where we add information about the simulation, as well as objective functions used in each example, instance size and initial and final maximum values of stress.

2.7.1 2D EXAMPLES

Lever. In this example shown in Figure 2.16, one part of the boundary of the black piece of the assembly is clamped to the table and in simulation assumed to satisfy the Dirichlet boundary condition; the part touching the vertical wall has an RD contact condition. Maximal stress is optimized, with no constraints imposed. As seen in Figure 2.16, we are able to reduce maximum

Table 2.2: Summary of experiments. In constraints column, “/” means that the weight is modified after each remeshing and varies between these values linearly. Column **Relative gradient norm** presents the ratio of the final gradient norm and the initial one from the first iteration. The last column (**Final L_p -norm reduction**) shows the reduction in the stress objective due to the last outer iteration, expressed as a percentage of the initial objective value. *For both bridge and coat rack (hooks), running 5 more outer iterations reduces stress to 1.80 and 346.51, relative gradient norm to 2.6e-2 and 2.81e-2, and final L_p -norm reduction to 0.95% and 0.00%, respectively. For fabrication, we used optimization results at outer iteration 5 for lever, hook, crane and at iteration 10 for bridge and coat rack.

Instance	Contact parameters $\alpha/\eta/\mu/\eta_n$	Iterations oi / max_oi	Constraints (= weights)	Post vol. min.	BCs	Tets	Initial stress	Final stress	Relative gradient norm	Final L_p -norm reduction
2D examples										
Connector	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3$	no	$\Gamma_D + DD + \Gamma_N$	1250	22.6	2.18	2.38e-3	0.24%
Lever	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3, J^d = 10^2/10^3$	no	$\Gamma_D + RD + DD + \Gamma_N$	1447	8.46	0.49	2.88e-5	0.31%
Bridge*	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 200/2000, J^d = 10^2/10^3$	no	$\Gamma_D + RD + DD + \Gamma_N$	2073	7.33	2.21	1.55e-1	4.92%
Hook	$10^{-4}/10^{-4}/0.0/-$	9/10	$J_V^t = 10^2/10^3$	no	$RD + \Gamma_N$	1279	16.5	2.97	1.11e-2	0.00%
Hook (disassembly)	$10^{-5}/10^{-5}/0.0/-$	6/10	$J_V^t = 10^2/10^3, J^d = 10^2/10^3$	no	$RD + \Gamma_N$	1279	16.5	3.82	2.11e-2	0.74%
Crane	$10^{-5}/10^{-5}/0.1/10^{-3}$	10/10	$J_V^t = 10^2/10^3, J^d = 10^2/10^3$	no	$\Gamma_D + DD + \Gamma_N$	2087	8.47	1.09	7.81e-4	0.40%
3D multimaterial										
Stool	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	$RD + DD + \Gamma_N$	58264	2.68	1.12	3.23e-2	0.00%
Bench Twisted	$10^{-4}/10^{-4}/0.2/10^{-2}$	8/10	$J_V^t = 10^4/10^5$	no	$RD + DD + \Gamma_N$	12731	4.38	1.71	5.31e-2	0.00%
Bench Bended	$10^{-4}/10^{-4}/0.2/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	$RD + DD + \Gamma_N$	12362	2.82	1.99	1.23e-1	0.00%
Coat rack (base)	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	$RD + DD + \Gamma_N$	15464	47	22.7	2.28e-2	0.17%
Coat rack* (hooks)	$10^{-4}/10^{-4}/0.4/10^{-2}$	10/10	$J_V^t = 10^4/10^5$	no	$\Gamma_D + DD + \Gamma_N$	15726	1260	385	7.53e-2	2.16%
3D pipe-like										
Tent (baseline)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5, J^a = 10$	yes	$RD + DD + \Gamma_N$	15325	119	59.3	2.73e-2	0.24%
Tent (disassembly)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5, J^a = 10, J^d = 1.0$	no	$RD + DD + \Gamma_N$	15325	119	59.1	5.06e-2	0.84%
Tent (parallel)	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5, J^a = 10, J^d = 1.0, J^p = 0.1$	no	$RD + DD + \Gamma_N$	15325	119	70.8	2.33e-2	0.08%
Truss	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5, J^a = 10, J^d = 1.0, J^p = 0.1$	yes	$\Gamma_D + DD + \Gamma_N$	29882	136	48.7	2.63e-2	0.15%
Dodecahedron	$10^{-5}/10^{-5}/0.4/10^{-3}$	50/50	$J_V^t = 10^4/10^5, J^a = 10, J^d = 1.0, J^p = 1.0$	yes	$DD + \Gamma_N$	23121	239	26.1	2.12e-2	0.00%

stress more than 14 times compared to the initial shape. The optimized shape naturally evolves into an interlocking assembly that can support a far higher load without large deformation.(see the video in supplementary material). We fabricated resulting shape (the process is described in Appendix 2.9.3, and performed a stress test on it, by loading it with increasing weight. Our setup is shown on Figure 2.17. The unoptimized lever breaks with a weight of 400g, whereas the optimized lever can endure more than 6400g, consistently with simulated results.

Bridge. Our second 2D example is a simple bridge model (Figure 2.18), with Dirichlet conditions on fixed rectangular parts, 3 optimizable parts, with boundaries of the supports partially fixed,

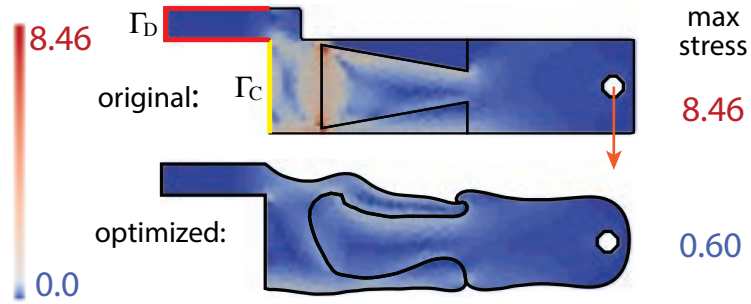


Figure 2.16: Lever optimization



Figure 2.17: Lever optimization fabrication

and DD contact conditions between parts. Due to symmetry, we run the optimization on one half of the shape, resulting in 3 times stress reduction.

The left and right support parts of the bridge were clamped to the table and weights were attached to the middle of the bridge. The experiments show that unoptimized bridge breaks at the left or right assembly with a load of 1500g, whereas the optimized bridge can hold more than 5800g of weights. See photo of our bridge in Figure 2.19.

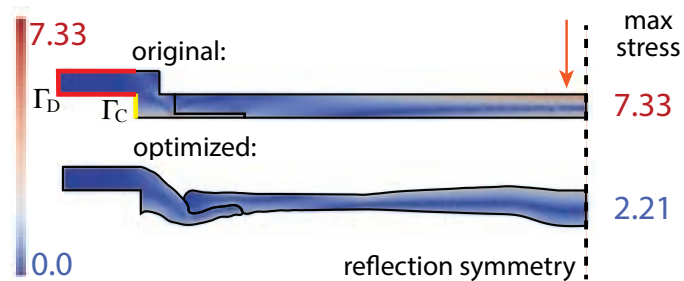


Figure 2.18: Initial and optimized *bridge* model.



Figure 2.19: Fabricated initial and optimized *bridge* shapes.

Hook. Figure 2.20, where we built an optimized *hook* that fits on the top of a door, with a single large contact area Γ_C , and *RD* boundary conditions on that part. In this scenario, we use multiple separate loads, one emulating, e.g., a coat hanging from the hook, and the other a force exerted when it is taken off the hook. In addition to multiple loads, this example also demonstrates the importance of disassembly energy: we make sure that accidental push from below does not result in the hook getting detached. The disassembly direction is shown in Figure 2.20. We perform stress minimization with a volume bound equal to the original volume, and compare the results with and without the disassembly constraint. Note that when the hook optimized without disassembly constraint is loaded, the left part lifts up, resulting in detachment from the support (Figure 2.20, middle) It remains stable once the disassembly constraint is enabled (Figure 2.20, right).

Figure 2.21 shows the experiments with 3 versions of 3d printed hook (unoptimized and optimized with and without disassembly constraint). The unoptimized hook breaks with 4500g load. The two optimized hooks can hold weights more than 19000g. Qualitatively, the behavior of loaded hooks matches the simulation.

Tower Crane. Our last 2D example is the tower shown in Figure 2.22, demonstrating an assembly consisting of four pieces with a Dirichlet condition on the bottom one, and DD contact conditions on the remaining parts.

In this example we demonstrate how our method can be combined with a simple ESO-like topology optimization technique (e.g., [Huang and Xie 2010]). to decrease the weight beyond what is possible with shape optimization only. We used a filtering technique to remove triangles with average stress

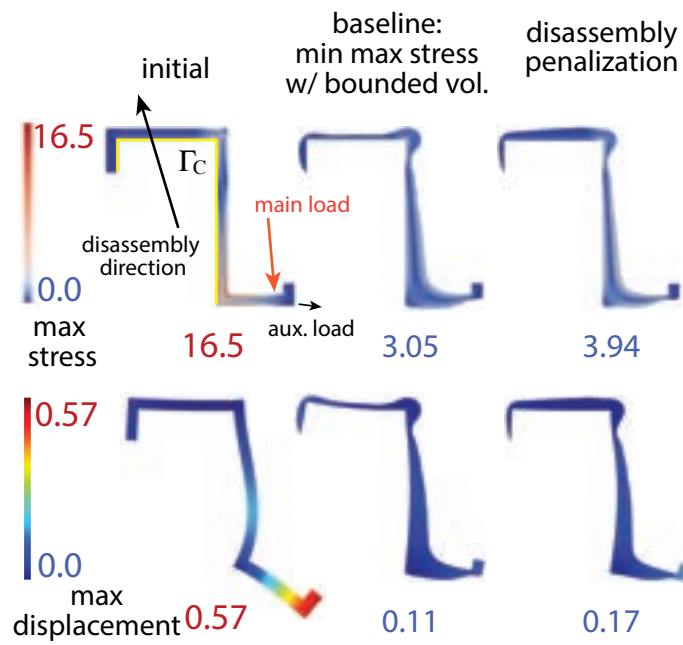


Figure 2.20: Top: undeformed shapes of different versions of the *hook* model. bottom: deformations under load (displacements exaggerated).



Figure 2.21: 3D printed *Hook* loading (Please see supplementary video.)

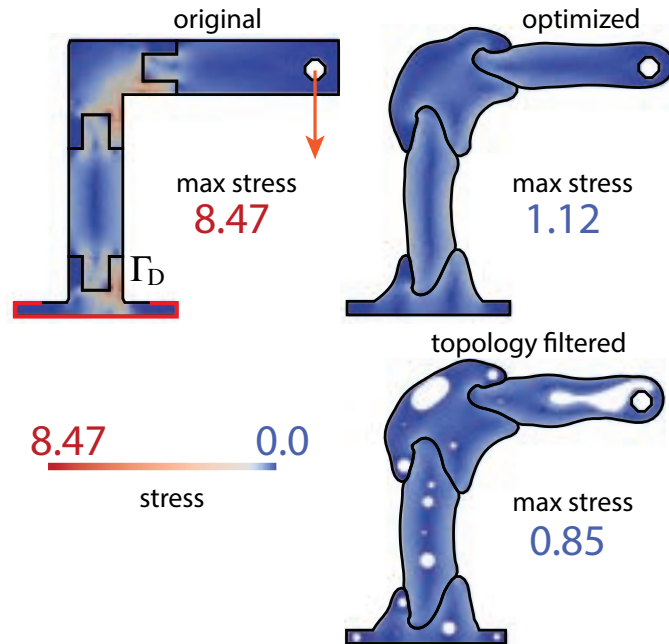


Figure 2.22: *Tower crane optimization*



Figure 2.23: *Tower crane fabrication*

lower than 10% of the maximum value. The filtered mesh was then again optimized, obtaining the result shown on the bottom-right. The laser-cut tower model is shown in Figure 2.23. In our experiment, the base of the tower was clamped to the table. Experiments show that the unoptimized tower breaks with 700g load, whereas the optimized tower can sustain more than 4600g load.



Figure 2.24: Left: the initial *stool* model; Right: the result after optimizing the white (PLA) pieces.

2.7.2 3D EXAMPLES

In the next 3 examples, we apply our algorithm to 3D assemblies of parts made of different materials: wood ($E_{wood} = 10000, \nu_{wood} = 0.3$), MDF ($E_{MDF} = 4000, \nu_{MDF} = 0.25$) and PLA plastic ($E_{PLA} = 3500, \nu_{PLA} = 0.36$). Moreover, in these examples, only some pieces of the shape (those made of PLA, which can be 3D printed) were optimized.

Stool. Our first example was a *stool* assembly with a fixed MDF top, fixed wood legs and connectors made of PLA (Figure 2.24). In this example, we use volume constraints. The stress concentrations for optimized and noncan be seen on Figure 2.25.

The optimization reduces stress 2.5 times without changing the total volume. If 10% increase in the volume is allowed, the stress decreases to 0.88, and to 0.85 at 20%.

Bench. Our second example is a *bench* with fixed parts (seat, legs) made of wood and optimizable PLA connectors (Figure 2.26). We consider two different leg shapes and show how connectors are optimized for each. We observe that, depending on the type of legs, we obtain a very different level of stress reduction: 30% in one case, and 2.5x times in the other (see Figure 2.27).

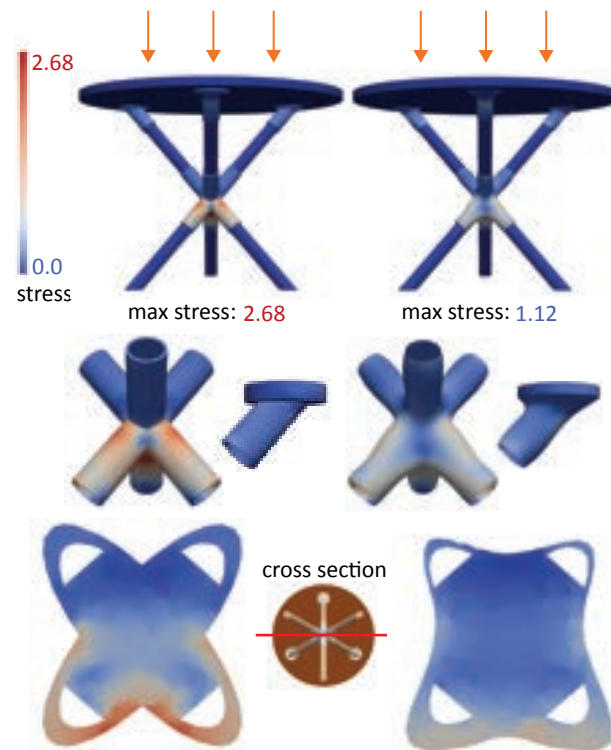


Figure 2.25: Stool stress: stress concentrations on stool example. Top figures present how load is applied and general stresses (restricted to PLA). Middle figures show stresses on each PLA piece. Bottom figure shows cross section of region where stress is more concentrated.



Figure 2.26: Bench model with two different types of legs and the initial and optimized connectors.



Figure 2.27: Bench stress for twisted and bended set of legs

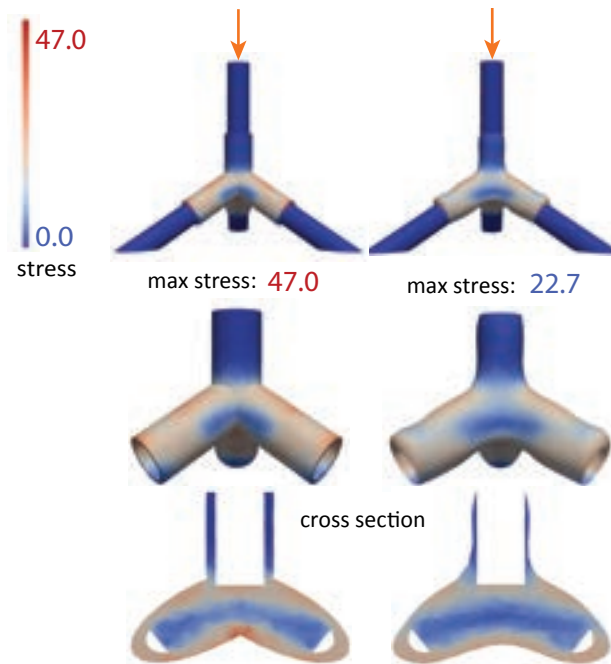


Figure 2.28: *Coat rack* base stress distribution.

Coat rack. Our final multimaterial example is a *coat rack* made of wood and plastic (Figure 2.1). Differently from both previous examples, this object is modular, in the sense that you can always use a longer cylinder and add identical plastic parts to increase the height and the amount of hangers in your object. Here, we optimized our two different plastic parts for separate sets of loads, the leg connector and the hook.

The leg connector is optimized with (fixed) legs in contact with the ground and the load applied at the top (Figure 2.28).

The hook attachment was optimized with loads applied to hooks. As shown in Figure 2.29, our framework was able to lower maximum stress more than 3 times.

We 3D printed the result using PLA (Figure 2.30).

For our last set of experiments, we implemented a framework for generating pipe assemblies from graphs, where each vertex of our graph becomes a sphere and each edge becomes a pipe. See an

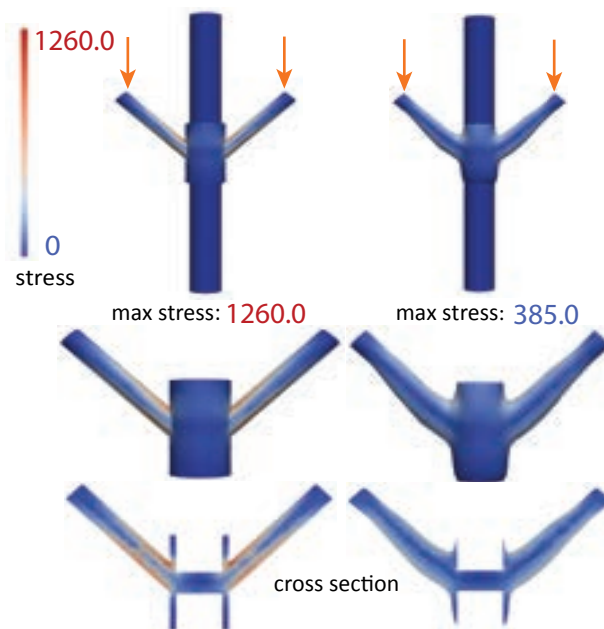


Figure 2.29: *Coat rack* hook connector stress distribution.

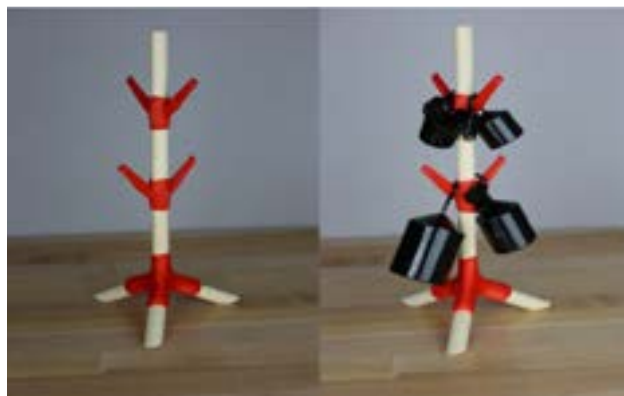


Figure 2.30: Mounted coatrack

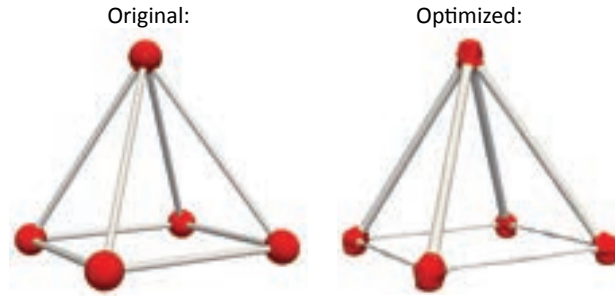


Figure 2.31: Initial and optimized *tent* assemblies.

example on Figure 2.31.

We used this tool to generate three different simple instances of our problem: a tent, a truss-like bridge and a dodecahedron. Below we detail more about each of these assemblies.

Tent. This object (Figure 2.31) is composed of 5 ball connectors and 8 pipes. A load is applied on the top of the structure pointing down and the bottom part of the object is in contact with the ground. We minimize stress, while keeping the same volume impose the assembly constraint. We were able to reduce stress by a factor of 2. We again ran our framework for reducing the volume of the shape, while keeping the stress bounded by the result of the previous optimization for stress (Figures 2.31 and 2.32). Then we add disassembly and parallel alignment objectives and the results are shown in Figure 2.33. Note that we optimize both the bars and connectors.

While the resulting structure can support this particular load well, it is likely to fall apart if any lateral load is applied, as resulting connectors are very shallow. Adding disassembly and parallel alignment constraints produces more resilient connectors (Figure 2.33). When using only disassembly constraint, the final stress (59.1) obtained was similar to the baseline case (59.3), while adding parallel alignment constraint to make at least 20% of the connection surface (in each connector) parallel to the disassembly direction increased stress to 70.8. **Dodecahedron.** This example is shown in Figure 2.34. For this example we use symmetric compression loads at the connectors at dodecahedron vertices. Using reflection symmetry on the three axes (x,y and z), we

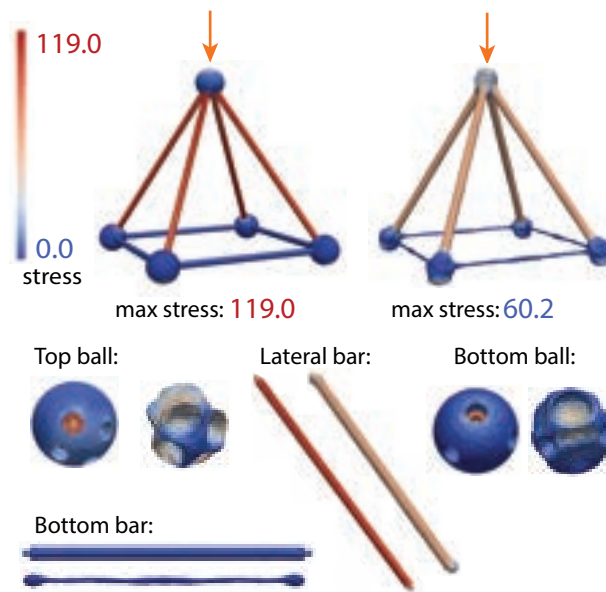


Figure 2.32: Tent stress

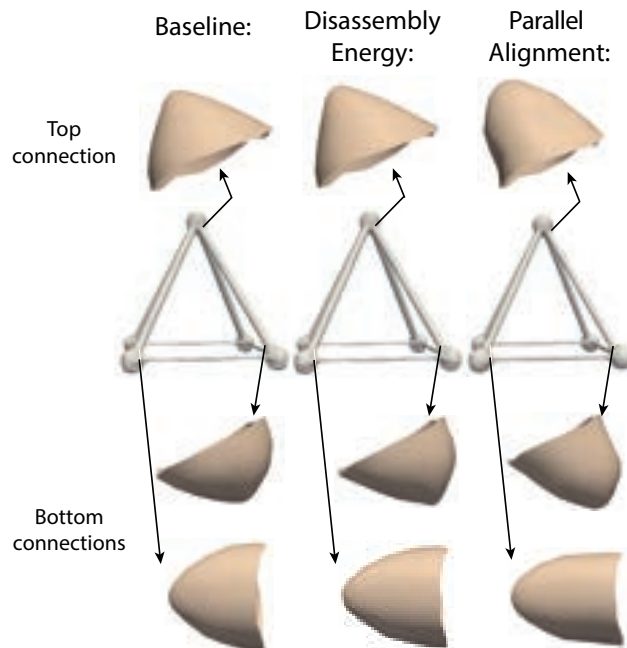


Figure 2.33: Tent connections

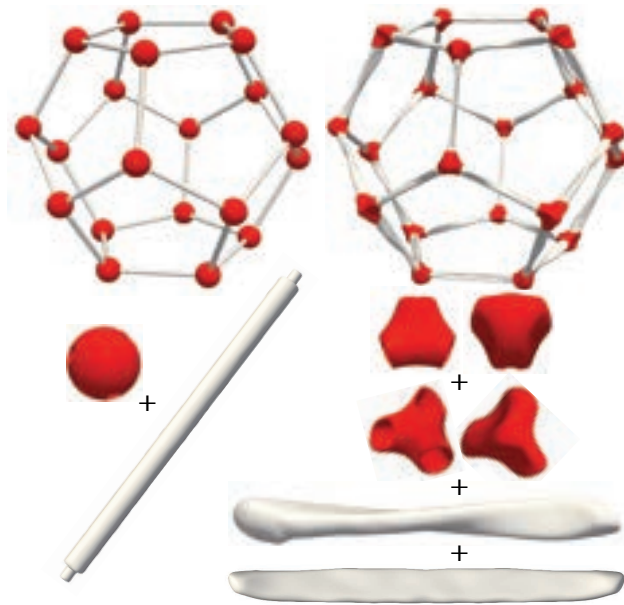


Figure 2.34: Dodecahedron

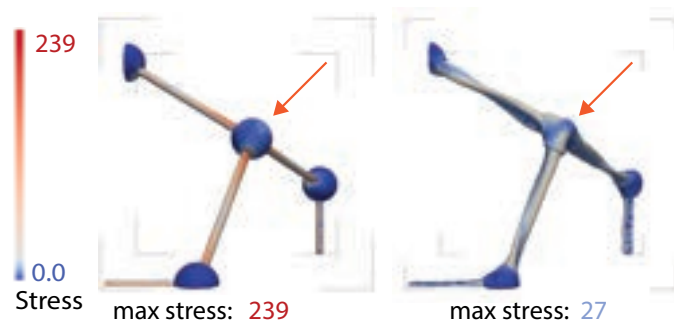


Figure 2.35: Dodecahedron stress: simulation and optimization are executed on an octant of the original shape, using reflection symmetry on all three axes. A diagonal force to the center of the shape is used.

were able to run our optimization on an eighth of the shape, as presented in Figure 2.35, where our boundary conditions are also shown.

We use a similar optimization setup as for *tent*: first, run stress minimization with a volume constraint, then minimize volume with a stress constraint, and including fixed assembly, disassembly and parallel alignment constraints. Notice in Figure 2.35 we were able to reduce stress by almost 10 times, while reducing volume in 12.6% compared to the initial dodecahedron. The emerging twisted shape for the bars are optimal for load support.

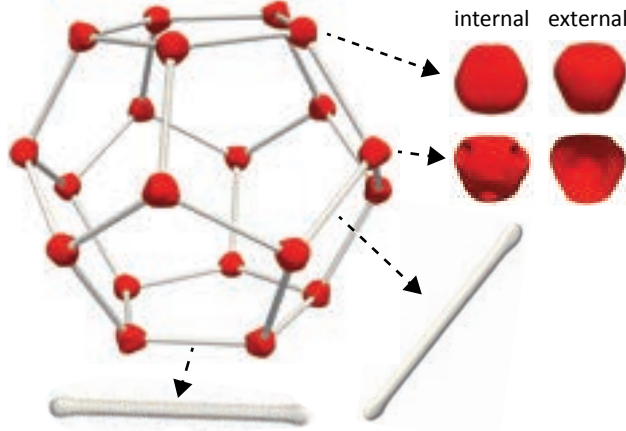


Figure 2.36: Connectors optimization: result after optimizing only sphere connectors. We show our two different types of connector (showing two sides of each) and two different types of pipes, which were not optimized with exception of their tips.

We also investigated the effect of optimizing only the connectors (balls) of the dodecahedron example. Again, we first minimize stress and then, in the second step, minimize the volume. We were able to reduce stress by around 28% to 172, and reduce the volume by 22%. The result can be seen in Figure 2.36.

Truss. The initial shape and the optimized result are shown in Figure 2.37. Similar to the dodecahedron, we optimized for strength (keeping initial volume) and then for volume (keeping stress close to its minimum level). We were able to reduce stress by more than half and volume by 20%. Notice that the optimized connectors are smaller than original ones while keeping deep connections. This result can be obtained thanks to the disassembly and parallel alignment energies.

Termination and convergence. To understand the convergence behavior of the method, we run our optimization for a fixed number of iterations which we aimed to set high enough for obtaining maximal possible improvement. Table 2.2 lists several indicators of convergence termination: number of actual iterations performed before stopping, relative gradient norm, and per-step objective reduction, expressed as a percentage of the initial objective value.

In the cases when the optimization reached the maximal number of steps (10 outer iterations for

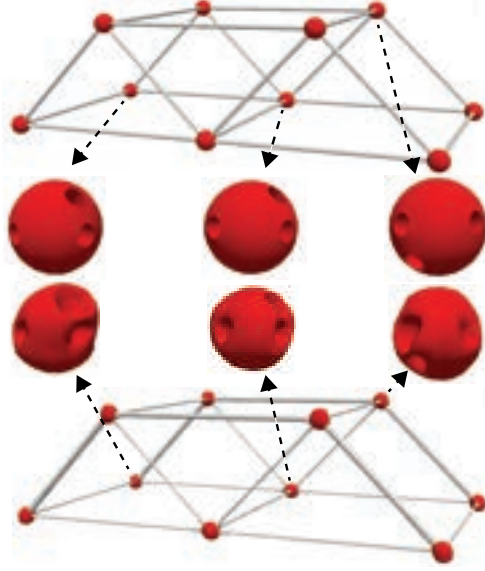


Figure 2.37: Truss

2D and multi-material examples and 50 for 3D examples), the result is close to a local minimum, or the maximal number of iterations is insufficient. In most cases, relative gradient norm is 5% or significantly less of the original, indicating that the result is close to local minimum. In one case, (*Bridge*) additional iterations also decrease the relative norm below 5%.

In several cases (*hook* and two *bench* variations) the optimization stops for geometric reasons (in some cases, topology changes are needed to achieve an optimum); in this case, the optimization cannot reach a local minimum by additional iterations, although the gradient reduction is also high in most cases.

Examining how the objective changes with iteration suggests that values $m = 2$ and $\epsilon_r = 1\%$ are adequate for all examples (considering the criterion discussed in Section 2.5 for our stress objective), except truss-like structures, where values $m = 5$ and $\epsilon_r = 0.5\%$ are more suitable due to higher oscillations in the objective early in the optimization. We emphasize that many other convergence criteria can be used; determining an optimal one is not our focus.

2.8 CONCLUSIONS AND LIMITATIONS

We have described a framework for shape optimization with contact and friction, using a smoothed, penalty-based model, and applied it to optimization of assemblies in which parts are held together by contact and friction forces. We demonstrated that in our framework a number of functionals can be optimized reliably, producing significant improvements e.g., in stress concentrations or volume, while maintaining various types of constraints. We validated the results with computational experiments in two and three dimensions and with qualitative experiments using fabricated objects. The experiments confirmed significant improvements in strength. In addition, since we support nonlinear solves, our model can be easily extended to a variety of functionals and constraints. The code for this project will be open-sourced.

Limitations. Our algorithm may (1) not reach a minimum or (2) find a local minimum that does not improve the objective much. In the first case, the optimization stalls: no progress is possible for one of the following reasons: (a) nonlinear solve does not converge (e.g. if the system has no static equilibrium or bad-quality elements not allowing to achieve residual tolerance in max number of iterations); (b) inversions of elements lead to a severe line search step restriction, as mesh quality improvement is not guaranteed to succeed; (c) meshing with tolerance using fTetwild may produce volume meshes for pieces with non-manifold surfaces, while surface-preserving remeshing does not improve quality enough. In case (2), the algorithm may reach a local minimum without significant decrease of the objective value, while other minimal with lower values exist. Case (1) was the most observed in our experiments, occurring in our hook and bench examples due to narrowing regions during optimization; note that we still obtain a significant stress reduction.

In addition, we use linearized elasticity model that does not account well for large deformations. As the contact and friction terms in our equations are already non-linear and we use a nonlinear solver, extension to nonlinear elasticity is likely to be straightforward. While the functionals we

have introduced in this work perform quite well for a number of tasks, there are restriction on the type of connections they can capture. E.g., snap connections are not handled easily with these constraints only. Expanding the range of objectives is an important direction for future work. The method can be considerably sped up but using parameterized geometric models, or model reduction, as well as more efficient solvers.

2.9 CHAPTER NOTES

2.9.1 FORMULAS FOR THE ADJOINT EQUATION TERMS.

$$\begin{aligned}
T'_C &= \frac{1}{\alpha} \int_{\Gamma_C} \left(h'_\eta(u \cdot n) (D[u] \cdot n) \right) w \cdot n \, dS \\
T'_S &= \frac{1}{\alpha} \int_{\Gamma_S} \left(h'_\eta([u] \cdot n^-) (D[[u]] \cdot n^-) \right) w \cdot n \, dS \\
T'_{CF} &= \frac{\mu}{\alpha} \int_{\Gamma_C} h'_\eta(u \cdot n) (n \cdot D[u]) N'_\eta(u_t) \cdot w + \\
&\quad + h_\eta(u \cdot n) N''_\eta(u_t) (I - nn^T) D[u] \cdot w \, dS \\
T'_{SF} &= \frac{\mu}{\alpha} \int_{\Gamma_S} \left(h'_\eta([u] \cdot n^-) n \cdot D[u - u^o] \right) \left(N'_\eta((u - u^o)_t) \cdot w \right) \\
&\quad + h_\eta([u] \cdot n^-) N''_\eta((u - u^o)_t) (I - nn^T) D[u - u^o] \cdot w \, dS
\end{aligned}$$

2.9.2 FORMULAS FOR THE DISCRETE SHAPE DERIVATIVE COEFFICIENTS $S_m[u, \rho]$

$S_m[u, \rho] = S_m^e + S_m^c$, where the first term is related to the elasticity equation and the second to contact and friction terms. In turn, $S_m^c = S_C + S_S + S_{CF} + S_{SF}$, the sum of terms for different types of contact forces. For these terms we have the following expressions, where we consider uniform surface load $g = \frac{G}{|\Gamma_N|}$:

$$\begin{aligned}
S_m^e = & \left(\int_{\Omega} [e - \varepsilon(\rho) : \sigma] \nabla \lambda_m + [\nabla \lambda_m \cdot (\sigma \rho_n + (\varepsilon(\rho) : C - \tau) u_n)] \nabla \varphi_n dV \right) + \\
& \left(-\frac{1}{|\Gamma_N|} \left(\int_{\Gamma_N} \rho \cdot g d\Gamma_N \right) \left(\int_{\Gamma_N} \nabla \lambda_m d\Gamma_N \right) + \left(\int_{\Gamma_N} (\rho \cdot g) \nabla \lambda_m d\Gamma_N \right) \right)
\end{aligned} \tag{2.27}$$

$$\begin{aligned}
S_C = & -\frac{1}{\alpha} \int_{\Gamma_C} h'_\eta(u \cdot n) (\rho \cdot n) (-(u \cdot \nabla \lambda_m) + (u \cdot n)(\nabla \lambda_m \cdot n)) n \\
& + h_\eta(u \cdot n) (-(\rho \cdot \nabla \lambda_m) + (\rho \cdot n)(\nabla \lambda_m \cdot n)) n \\
& + h_\eta(u \cdot n) (\rho \cdot n) \nabla \lambda_m dS
\end{aligned}$$

$$\begin{aligned}
S_S = & -\frac{1}{\alpha} \int_{\Gamma_S} h'_\eta([u] \cdot n^-) (\rho \cdot n) \left[-([u] \cdot \nabla \lambda_m) + ([u] \cdot n^-)(\nabla \lambda_m \cdot n^-) \right] n^- \\
& + h_\eta([u] \cdot n^-) (-(\rho \cdot \nabla \lambda_m) + (\rho \cdot n)(\nabla \lambda_m \cdot n)) n \\
& + h_\eta([u] \cdot n^-) (\rho \cdot n) \nabla \lambda_m dS
\end{aligned}$$

$$\begin{aligned}
S_{CF} = & \frac{\mu}{\alpha} \int_{\Gamma_C} -h'_\eta(u \cdot n) \rho \cdot N'(u_t) ((u \cdot n)(\nabla \lambda_m \cdot n) - u \cdot \nabla \lambda_m) n \\
& + h(u \cdot n) (N''(u_t) \cdot \rho) \cdot (-u \cdot \nabla \lambda_m n + \\
& (u \cdot n)(\nabla \lambda_m \cdot n) n + (u \cdot n)(-\nabla \lambda_m + (\nabla \lambda_m \cdot n) n)) n \\
& - h(u \cdot n) \rho \cdot N'(u_t) \nabla \lambda_m dS
\end{aligned}$$

$$\begin{aligned}
S_{SF} = & \frac{\mu}{\alpha} \int_{\Gamma_S} -h'_\eta([u] \cdot n^-) \rho \cdot N'([u]_t) (([u] \cdot n^-)(\nabla \lambda_m \cdot n^-) - [u] \cdot \nabla \lambda_m) n^- \\
& + h([u] \cdot n)(N''([u]_t) \cdot \rho) \cdot (-[u] \cdot \nabla \lambda_m n^- + \\
& ([u] \cdot n^-)(\nabla \lambda_m \cdot n^-) n^- + ([u] \cdot n^-)(-\nabla \lambda_m + (\nabla \lambda_m \cdot n^-) n^-)) n^- \\
& - h([u] \cdot n^-) \rho \cdot N'([u]_t) \nabla \lambda_m dS
\end{aligned}$$

2.9.3 FABRICATION PROCESSES

We have validated the performance of our system on several simple assemblies fabricated with 3D printing and laser-cutting. These are shown in Figures 2.17-2.18-2.20-2.22. For the lever and hook example as shown in Figures 2.17, 2.21, the fabrications are done with 3D printing (Ultimaker 3), using PLA material. For these experiments, we used Ultimaker Cura for slicing. We used fine printing setting, with 0.1mm layer height and infill density equal to 100%. The general print speed was also altered to 40mm/s and reduced to 12mm/s (18mm/s) when printing the inner (outer) walls of the object, in order to have a more precise printing of connections. In addition, to guarantee that objects connect, we offset the polygon shapes by 0.05mm, reducing them to account for the printing thickness.

For the bridge and tower examples (see Figures 2.18, 2.23), the experimented objects were obtained through laser-cutting 1/8" (3.175mm) acrylic boards, using Epilog Mini cutter. Notice that laser-cutting removes material, so we perturbed the models to account for that, offsetting the polygonal shapes by 0.05mm.

For our 3D examples, we also 3D printed our coat rack optimized model, shown in Figure 2.30. We used the fast setting on Cura software, with 0.2mm layer height, and reduced the speed to 50mm/s. Even though we assembled with wood dowels, which can carry imprecision in the measures, the pieces still fit tightly.

2.9.4 ADAPTIVE QUADRATURE

In different steps of our algorithm, during the simulation and also shape derivative computation, we need to perform the computation of integrals. For simpler cases using linear elasticity, this is not a challenge and we can simply use Gaussian quadrature, for example. However, when adding penalization due to normal contact and friction, the choice of how to perform these computations becomes very important.

As an example, consider the discretized normal contact force. Here, we assume we are using quadratic FEM elements. Our discretized contact function per contact element Γ_S^e is:

$$j'_{N,\alpha}{}^D(u) = \frac{1}{\alpha} \int_{\Gamma_S^e} \varphi^A(p(x)) n_i h_\eta((u(p(x))) \cdot n) dS$$

where α is a constant, n is constant for each element, $p(x)$ maps each point x into barycentric coordinates of the element, φ corresponds to the quadratic baricentric basis and u , the displacement, is also quadratic. h_η is the same as previously defined:

$$h_\eta(y) = \begin{cases} 0 & y \leq -\eta \\ \frac{1}{4\eta}y^2 + \frac{1}{2}y + \frac{\eta}{4} & -\eta \leq y \leq \eta \\ y & y \geq \eta \end{cases}$$

Notice that $u(p) \cdot n$ defines the region of the piecewise function. To find the boundary between each of these regions, we would need to solve $u(p) \cdot n = \pm\eta$. Notice that this corresponds to a quadratic 2D equation and the solution corresponds to a conic section: parabola, hyperbole, ellipse or circle. We have 2 equations, so our original domain is partitioned by a maximum of 2 conic sections. Since the two equations differ only by η , we expect that the regions boundary are going to have very similar shapes. See an example in [Figure 2.38](#):

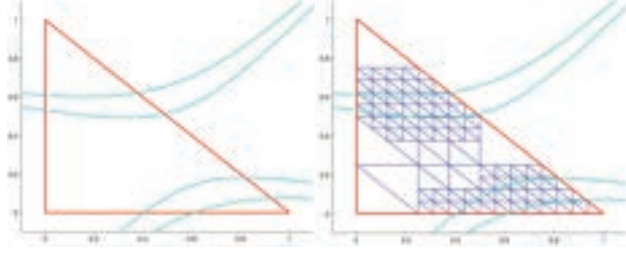


Figure 2.38: Example of different regions in piecewise quadrature computation with $\eta = 10^{-2}$. In this case, we have two hyperbolas. So, in total, 5 different regions. On the right side, an example of how the triangles can be split.

To show the accuracy of the quadrature computation, consider the following example in 3D: $u_0 = [0.0, 0.1, -0.05]$, $u_1 = [0.1, 0.2, -0.03]$, $u_2 = [-0.1, 0.0, -0.02]$, $u_3 = [-0.05, 0.01, 0.02]$, $u_4 = [-0.02, -0.02, 0.01]$, $u_5 = [0.02, -0.03, 0.01]$, $n = \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right]$, $\alpha = 10^{-4}$, $\eta = 10^{-4}$ and nodes and coordinates $A = 0$ and $i = 0$. Here, u_j corresponds to the discrete displacement at each of our 5 nodes of our triangle element. Using Matlab's `integral2` function (with absolute tolerance set at 10^{-10}), we can compute $j'_{N,\alpha}^D$ to obtain -0.329516 . Using (order 10) Gaussian quadrature, we obtain -0.438527 . These errors can accumulate and generate problems in convergence for our simulation.

We implemented then an adaptive approach for computing integrals in our framework. First, we define the scheme to be used by the regions of our triangle. We evenly sample the triangle using m points, identifying the regions where these points are contained. In our case, we used $m = 10$. If not all m points are from the same region (easily verified by computing $u(p) \cdot n$ value), we split the triangle into 4 pieces of same area and repeat this process recursively, until a maximum depth is reached. See an example on the right side of Figure 2.38. In our case, we identified that depth equal to 5 guarantees good results. Inside each of the final triangles, we use (order 10) Gaussian quadrature to compute the integral. Using this method, for the same example as above, we obtained result -0.329560 , which is accurate up to 4th digit after the decimal point when compared to the numerical result of the `integral2` function.

2.9.5 MESHING

We use triangle meshes in 2D and tetrahedral meshes in 3D for our shapes.

For 2D, we define shape outlines using Adobe Illustrator, encoding boundary conditions using color; shared boundaries between parts are represented as single polylines. We then transform the geometry in the resulting SVG file into the input format for Triangle [Shewchuk 1996a], removing the duplicate edges defining the contact area between different pieces. After obtaining the mesh for the interior, we split the triangle mesh into part meshes. At the end of each inner loop in Algorithm 1, we extract the polygonal boundary of the current solution and remesh it using Triangle again.

In 3D, the input to the meshing process is a collection of surface meshes of individual parts. We assume that in the contact zone, the distance between meshes is small, (in our specific examples we use boolean operations to produce surface for parts, with connectors obtained, e.g., by adding a cylinder to one part, and subtracting it for another, which produces close surfaces. After that, we generate the volumetric mesh using fTetWild [Hu et al. 2020a] with surface meshes from the previous step as input. fTetWild has a threshold ϵ_m within which close parallel triangles of a surface mesh are merged; we set it to a sufficiently large value to obtain a single contact surface. For most cases, we used the default value of $1e-3$ for ϵ_m , but increased the value to adjust to different gaps coming from the surface mesh generation.

As in 2D, we apply remeshing to keep a good mesh quality after a sequence of optimization steps. In 3D, remeshing is done in different ways: (a) preserving mesh surface, which is done with CGAL’s package for tetrahedral remeshing [Tournois et al. 2021]; or (b) remeshing the boundary as well, by reapplying fTetWild. Choosing between this two options is also a parameter in our framework. We used option (a) for most of our scenarios, but for truss and dodecahedron example, boundary remeshing was used at every 5 remeshing operations. If remeshing the boundary with

fTetWild fails, e.g., due to generation of pieces with non-manifold meshes, option (a) is used instead.

3 | DIFFERENTIABLE SOLVER FOR TIME-DEPENDENT DEFORMATION PROBLEMS WITH CONTACT

This chapter is based on a recently submitted work [Gjoka et al. 2022] in which I share the first authorship with Zizhou Huang from NYU. This work is also co-authored by Arvi Gjoka, Zachary Ferguson and Professors Daniele Panozzo and Denis Zorin from NYU, and Prof. Teseo Schneider from the University of Victoria.

My contribution to this project was working on the adjoint formulation and the algorithm for optimizing the shape, material, and other physical properties of our simulations, which can include contact and friction. I also participated in analyzing results and designing, fabricating, and running physical experiments in our hardware laboratory.

We introduce a general differentiable solver for time-dependent deformation problems with contact. Our approach uses a finite element discretization with a high-order time integrator coupled with the recently proposed incremental potential contact method for handling contact and friction forces to solve PDE- and ODE-constrained optimization problems on scenes with a complex geometry. It support static and dynamic problems, it support differentiation with respect to all

physical parameters involved in the physical problem description, which include shape, material parameters, friction parameters and initial conditions. Our analytically derived adjoint formulation is efficient, with an overhead of not more than 2 times the forward simulation, and shares many similarities with the forward problem, allowing reusing large parts of the code of an existing forward simulator code.

We implement our approach on top of the open-source PolyFEM FE library, and demonstrate the applicability of our solver to shape design, initial condition optimization, and material estimation on both simulated results and in physical validations.

3.1 INTRODUCTION

PDE-constrained and ODE-constrained optimization problems, i.e. the minimization of a functional depending on the state of a physical system modeled using a set of (partial) differential equations, appear in many application areas: optimized design in engineering and architecture, metamaterial design in material science, inverse problems in biomedical applications, controllable physically-based modeling in computer graphics and policy optimization and physical parameter estimation in robotics.

A common family of PDE-constrained optimization problems in graphics, robotics, and engineering involve static or time-dependent elastic deformation problem (which can be infinitely stiff, leading to rigid simulation problems), often interacting between each other via contact and friction forces.

Due to their wide and cross-disciplinary interest, a large number of approaches have been proposed to tackle PDE-constrained optimization. However, as discussed in Section 3.2, these approaches often make application-specific assumptions usually aimed at simplifying the differential simulator, on either robustness (e.g., using a non-robust contact model), accuracy (e.g., using inaccurate spatial discretizations, non-physical material models, or simplified contact/friction models), geometric

complexity (e.g., using coarse geometric proxies in lieu of a high resolution geometry), or scalability (e.g., restricting the number of dofs that can be optimized).

We describe a differentiable solver for this family of problems with the following characteristics:

1. We use a standard finite element discretization of arbitrary order, both in space and time for both linear and non-linear elasticity material models.
2. We support differentiation with respect to all physical parameters involved in physical problem description: shape, material parameters, friction parameters and initial conditions. An arbitrary subset of these parameters can be used in the objective function.
3. Our contact/friction formulation builds upon the recently proposed Incremental Potential Contact approach [Li et al. 2020a], which uses a linear approximation of the boundary to define contact and friction forces. Our differentiable simulator supports complex geometry, is automatic and robust (with only two major parameters to tweak to control the accuracy of the spatial and temporal discretizations), and guarantees physically valid configurations at all timesteps, without intersections and without inverted elements.
4. Our formulation supports both static and dynamic problems.
5. Our approach is efficient. The computation of the derivatives for one PDE-constrained optimization step is at most 2 times more expensive than a forward evaluation of the underlying forward simulation of the physical systems.

Overview. We summarize the key elements we use in our differentiable solver. Most importantly, we use the *adjoint equation* approach to computing functional gradients with respect to all parameter types, including, e.g., shape parameters affecting the PDE domain, or material properties. While this approach requires extensive PDE- and objective-specific analytic derivations, it has significant

advantages over techniques based on either automatic differentiation or finite differences:

1. It is efficient for large number of degrees of freedom, as the overhead of computing derivatives is less than a factor of 2 with respect to a forward solve;
2. It allows to reuse the majority of the forward PDE solver code, as the adjoint equations are very similar to forward equations. In our case, we use the open-source solver PolyFEM [Schneider et al. 2019], inheriting its ability to use high order basis, high order elements, and its framework for h - and p - refinement.

Other important technical features of our approach include:

- Our discretization of the adjoint equations is *consistent*, i.e., yields exactly the derivatives of the discrete objective with respect to discretized optimization variables; See Section 3.4.
- We use the *perturbed-domain* parameterization of PDE solutions for derivatives, which considerably simplifies the gradient formulations in the case when shape differentiation is needed (Section 3.7).
- We build on a *fully differentiable formulation of contact and friction*, avoiding the need for handling non-smooth constraints: all types of forces are handled in a uniform manner.

We demonstrate the effectiveness of our approach on a set of examples involving multiple objectives and optimizing for the shape, material parameters, friction parameters, and initial conditions.

3.2 RELATED WORK

We summarize the most relevant simulation frameworks, primarily focusing on those supporting differentiable simulation of elastic deformable objects and robotics systems.

Differentiable Deformable Object Simulators. Numerous bespoke differentiable elastic body simulators have been developed for applications in generative design of shapes [Panetta et al. 2017c; Tozoni et al. 2020], actuators [Skouras et al. 2013a; Chen et al. 2020; Maloysel et al. 2021], and sensors [Tapia et al. 2020], material characterization [Hahn et al. 2019; Schumacher et al. 2020], and robotic control [Bern et al. 2019; Hoshyari et al. 2019]. These simulators broadly fit into three categories: (i) those employing analytic derivatives computed using sensitivity analysis; (ii) those using generic automatic differentiation libraries [Hu et al. 2019a; Heiden et al. 2020]; and (iii) neural surrogate models replacing the entire simulation with a differentiable neural network [Chang et al. 2016; Zhang et al. 2016; Baque et al. 2018; Bern et al. 2020]. Analytic sensitivity analysis is the most common, but generally requires manual differentiation of the physics equations and construction of the adjoint equations for specific functionals [Liang et al. 2019; Qiao et al. 2020; Bern et al. 2019; Rojas et al. 2021; Du et al. 2021]. One exception is Dolphin-Adjoint [Mitusch et al. 2019], which automatically and robustly derives adjoint models for models written in the finite element software FEniCS. Automatic differentiation is more flexible but requires existing simulators to be essentially rewritten and incurs a performance penalty, especially for complex solvers. Surrogate models, though promising dramatic speedups, require huge training sets for even simple design spaces [Gavriil et al. 2020], are unsuitable for high-precision applications, and are opaque black boxes [Bächer et al. 2021]. To the best of our knowledge, none of these simulators support robust handling of contact and friction for complex geometries, and they only support a subset of the design parameters compared to the generic formulation of this paper.

Meshfree Methods. Especially for shape optimization, methods like XFEM [Schumacher et al. 2018b; Hafner et al. 2019] and MPM [Hu et al. 2019b] that do not maintain conforming meshes might appear attractive to circumvent remeshing-induced discontinuities [Bächer et al. 2021]. However, these methods sacrifice accuracy [de Vaucorbeil et al. 2019], particularly for stress minimization problems [Sharma and Maute 2018]. Furthermore, for remeshing to alter the

objective or constraint functionals to the degree that it impacts optimization, those functionals must not have been sufficiently resolved by the mesh to begin with. Our approach computes accurate displacement and stressed by using a finite element method framework using high-order elements, coupled with dynamic remeshing to compensate for the distortion introduced by large deformations.

Differentiable Simulations with Contact. Differentiable simulators incorporating various contact models have recently been developed for rigid [Heiden et al. 2020] and soft bodies [Liang et al. 2019; Qiao et al. 2020; Geilinger et al. 2020b]. These models unfortunately suffer the same robustness issues as the underlying physics engines. Differently, our approach uses the recently proposed Incremental Potential Contact (IPC) formulation [Li et al. 2020a], replacing the traditional zero-gap assumption [Wriggers 1995; Kikuchi and Oden 1988; Stewart 2001; Brogliato 1999; Belytschko et al. 2000; Bridson et al. 2002; Otaduy et al. 2009; Harmon et al. 2008, 2009; Daviet et al. 2011; Verschoor and Jalba 2019] with a smooth version ensuring minimal separation between objects at every frame of the simulation. This approach was designed with the explicit goal to guarantee *robustness* and its smooth formulations of contact and friction avoids the need for handling non-smooth constraints. In our formulation, all types of forces are handled in a uniform manner.

Optimization with Contact. In recent years, advances were obtained in using techniques like shape and topology optimization in the context of contact and friction. Some previous works have considered the specific case of contact between a soft body with rigid surfaces, like [Beremlijski et al. 2014], [Haslinger et al. 1986] and [Herskovits et al. 2000]. Other works, like ours, have studied the interaction of two or more bodies in contact, as recent works from [Desmorat 2007], [Stupkiewicz et al. 2010], [Maury et al. 2017] and [Tozoni et al. 2021]. It is worth mentioning that most papers do not consider friction or use a simplified model (compared to the standard Coulomb formulation) as discussed in [Maury et al. 2017].

Due to their results and similarity with our work, it's worth looking at two of these techniques with deeper attention. In 2017, Maury et al. [Maury et al. 2017] presented a level set discretization technique where contact and friction were modeled through penalization energies, using smooth approximations to the problem. Using a similar contact and friction model, Tozoni et al. [Tozoni et al. 2021] designed a shape optimization technique that focused on reducing stress of assemblies that are held together by contact and friction, while enforcing constraints essential for assembly functionality, like weight, difficulty of putting the assembly together, and how reliably it stays together. Both these works followed the mathematical model of contact presented by Eck et al. [Eck et al. 2005].

Physics Engines for Graphics and Robotics. Physics engines for games, visual effects, and robotics prioritize fast, often real-time simulation over accuracy. Three in particular have gained widespread popularity in robotics research: Bullet [Coumans et al. 2013], NVIDIA PhysX [NVIDIA 2022], and Mujoco [Todorov et al. 2012]. These libraries support dynamics simulations of rigid and soft bodies, joints and actuators, and contact. However, the deformable objects in each engine are of limited accuracy and generality, and all engines encounter convergence failures and other catastrophic robustness issues in challenging contact scenarios, particularly with complex geometry. Furthermore, none of these engines offers parameter estimation, derivatives, or shape optimization support. Those features must be implemented separately, which is complicated by the nonrobustness. More advanced support for deformable object simulation is provided by the SOFA framework, which includes a robotics simulation plug-in [Faure et al. 2012; Coevoet et al. 2017] used in a number of works [Navarro et al. 2020; Coevoet et al. 2019; Goury and Duriez 2018; Morzadec et al. 2019]. However, SOFA suffers from the same robustness issues as other simulation frameworks in its current form (see [Li et al. 2020a] for comparisons).

Engineering Solvers. Commercial multiphysics software packages used in engineering prioritize accuracy and sophisticated material, damage, and frictional contact models over efficiency. Despite

Domains and bases		Functions on physical domain $\Omega_{\bar{q}}$	
D_d	Domain dimension, 2 or 3	$u_{\bar{q}}(x), \mathbf{u}$	PDE solution defined on $\Omega_{\bar{q}}$ with values in \mathbb{R}^{D_s} . We denote the vector of coefficients of u in the FE basis ϕ by \mathbf{u} . $u(x) = \sum_{\ell} u_{\ell} \phi^{\ell} \circ \bar{q}^{-1}(x) = \sum_{\ell} u_{\ell} \phi^{\ell}(x)$
D_s	Solution dimension, 1, 2 or 3	$\mathbf{w}(x), \psi(x), \mathbf{w}, \psi$	Test functions (scalar) defined similarly to $u(x)$ in the same basis and vectors of their coefficients are \mathbf{w} and ψ .
Ω_{ref}	Reference domain $\Omega_{\text{ref}} \subset \mathbb{R}^{D_d}$ consists of copies of identical reference elements \hat{K}_j , $j = 1 \dots n_K$ identified along edges.	$p(x), \mathbf{p}$	Adjoint solution - the solution of the adjoint equation and the vector of its coefficients, with values in \mathbb{R}^{D_s} .
\hat{x}_{ℓ} and \hat{z}_{ℓ}	Nodes are points in Ω_{ref} used to define bases, $\ell = 1 \dots n_N^x$, and $1 \dots n_N^z$ respectively. The set of nodes \hat{z}_{ℓ} does not include nodes with Dirichlet boundary conditions; the set of nodes \hat{x}_i does include these nodes.	$q^m(x), \mathbf{q}^m$	m-th optimization parameter $q^m(x) = \sum_{\ell=1}^{n_q^m} q_{\ell}^m \zeta^{\ell}(x)$ with a basis ζ^{ℓ} with values in $\mathbb{R}^{D_q^m}$ parameters can be material properties, boundary conditions etc, defined on all or parts of $\Omega_{\bar{q}}$. For the geometry map \bar{q} , ξ on $\text{Dom}(\bar{q}) = \Omega_{\text{ref}}$, and $\zeta^{\ell} = \xi^{\ell}$.
ϕ^{ℓ} and ξ^{ℓ}	FE basis functions are scalar basis functions defined on Ω_{ref} ; ξ^{ℓ} correspond to nodes \hat{x}_i , and is used for geometric maps (we use p.w. linear basis); ϕ^{ℓ} correspond to \hat{z}_i and used for all other quantities (arbitrary order Lagrangian).	PDE and derivatives	
$\bar{q}, \bar{q}^j(y^j), \bar{\mathbf{q}}, x_{\ell}$	Geometric map \bar{q} embedding a reference element in space, is defined on each \hat{K}_j in Ω_{ref} with local coordinates y^j as $\bar{q}^j(y^j) = \sum_{\ell} x_{\ell} \xi^{\ell}(y^j)$, where $x_{\ell} \in \mathbb{R}^{D_d}$ are the positions of the nodes of the element j forming the vector $\bar{\mathbf{q}}$. Concatenation of these maps yields the global geometric map $\bar{q} : \Omega_{\text{ref}} \rightarrow \mathbb{R}^{D_d}$.	$\mathbf{h}(\mathbf{u}, \mathbf{q}) = 0$	Discretized form of the PDE , i.e., a system of n_u algebraic equations with components of \mathbf{u} as unknowns.
$\Omega_{\bar{q}}$	Physical domain is the domain on which the PDE is solved, parametrized by $\bar{\mathbf{q}}$. $\Omega_{\bar{q}} = \bar{q}(\Omega_{\text{ref}})$. The global coordinate on $\Omega_{\bar{q}}$ is $x = x^{\bar{q}} \in \mathbb{R}^{D_d}$.	$J(\mathbf{u}, \mathbf{q})$	Discretized form of the objective.
$\phi^{\ell}(x), \xi^{\ell}(x)$	FE bases on $\Omega_{\bar{q}}$. The bases ϕ^{ℓ} and ξ^{ℓ} can be pushed forward to the domain $\Omega_{\bar{q}}$ via $\phi(x) = \phi \circ \bar{q}^{-1}(x)$ and $\xi(x) = \xi \circ \bar{q}^{-1}(x)$.	$\partial_{\bar{\mathbf{q}}} a(\mathbf{u}, \mathbf{q})$	Derivative of a (possibly) vector quantity a with respect to a vector of optimization parameters , not including dependence through u . The vector is the vector of coefficients of one of q^m or \bar{q} . If we the dimension of a is n_a , then $\partial_{\bar{\mathbf{q}}} a$ is a matrix of size $n_a \times D_q^m n_q^m$.
$\Omega_{\bar{q}+\theta t}$	Perturbed domain obtained using a perturbation direction θ in $\bar{\mathbf{q}}$. Perturbation $\theta(x) \in \mathbb{R}^{D_d}$ is: $\theta = \theta(x) = \sum_{\ell} \theta_{\ell} \xi^{\ell} \circ (\bar{q}^j)^{-1}(x) = \sum_{\ell} \theta_{\ell} \xi^{\ell}(x)$	$\partial_{\mathbf{u}} a(\mathbf{u}, \mathbf{q})$	Derivative of a quantity a with respect to the the PDE solution u ; it is a vector of length $D_s n_u$.
		$d_{\bar{\mathbf{q}}} a(\mathbf{u}, \mathbf{q})$	Full derivative of a with respect to \mathbf{q}, including through the dependence on u.
		$\nabla a(v), \nabla_i a(v, w)$	Derivatives of a with respect to arguments $v, w \in \mathbb{R}^D$.

Table 3.1: Notation.

their improved accuracy, they suffer many of the same robustness issues as physics engines, often requiring parameters to be tuned for specific scenes to achieve reliable convergence for complex geometry; see [Li et al. 2020a]. Consequently, batch processing for different boundary conditions and especially for different geometries as needed for PDE-constrained optimization problems is problematic and can require significant manual effort [Whalen et al. 2021]. While some of these packages provide basic support for shape optimization, gradients cannot be obtained efficiently.

3.3 GENERAL STATIC FORMULATION

We start with formulating a general static problem with few assumptions on the specific physics involved; we generalize to time-dependent problems in Section 3.5; we describe the details of the physical formulation in Section 3.6.

Typographical conventions. We use lower-case italic for functions $a(z)$ and variables z , with both z and a in \mathbb{R}^D , where $D = 2, 3$.

Boldface lower-case letters for vectors of coefficients of a FEM (or any other) discretization of a function: \mathbf{a} .

For a vector or matrix quantity, superscripts refer to the quantity as a whole, e.g., \mathbf{a}^i is the value of \mathbf{a} at time step i . Subscripts are used for the indices of components of a vector. The components themselves are denoted with italic letters. E.g.,

$$a(z) = \sum_{\ell=1}^n a_{\ell} \phi^{\ell}(z) \quad (3.1)$$

means that the function $a(z) : \mathbb{R} \rightarrow \mathbb{R}^D$ is a linear combination of ϕ^{ℓ} , with coefficients a_{ℓ} ; the vector of a_{ℓ} is \mathbf{a} .

If $a(z)$ has values in \mathbb{R}^D , its coefficients in a scalar basis ϕ^{ℓ} are D -dimensional, so these can be represented by a $D \times n$ matrix. \mathbf{a} is the flattened form of this matrix i.e., a vector of length $D \cdot n$, with coordinates of each components a_{ℓ} in sequential entries.

Notation. We summarize the notation used in the general formulation and the variable-coefficient Poisson example above in a more systematic way in the Table 3.1 assuming a static problem. For time-dependent case, it is mostly the same, but all quantities depend on time t or time step i after discretization.

Static problem formulation. We consider an optimization problem with force equilibrium equations imposed as a constraint:

$$\min_q J(u, q), \text{ such that, } \mathcal{H}(u, q) = 0 \quad (3.2)$$

In this problem, the function u is the solution of the equation $\mathcal{H}(u, q) = 0$, (e.g., elastic equilibrium) with parameters q (e.g., material properties, external forces, domain shape parameters) defined pointwise on the domain $\Omega_{\bar{q}}$. We use the weak form of the constraint $\mathcal{H}(u, q, w) = 0$, for any test function w , by integrating against arbitrary w from a suitable functional space on $\Omega_{\bar{q}}$. Equation $\mathcal{H}(u, q)$ may include local forces depending on u and its derivatives (elasticity) as well as forces involving integration over $\Omega_{\bar{q}}$ or its parts, as it is the case, e.g., with smoothed versions of contact forces.

In these equations, q is the collection of optimization parameters $\bar{q}, q^1 \dots q^{n_{par}}$. The first of these, \bar{q} plays a special role: it determines the shape of the domain on which the PDE is defined; it is a function on Ω_{ref} defining its deformation.

Each of the remaining optimization parameters q^m is either a function on $\Omega_{\bar{q}}$, or its part (e.g., boundary), or a globally defined scalar or vector; e.g., it may be a variable elasticity tensor defined per point of $\Omega_{\bar{q}}$, the friction coefficient of a pair of regions on its boundary, or the force applied by an actuator at a particular point.

Example. Consider a variable-coefficient Poisson equation $\nabla \cdot (c(x)\nabla u) = f$ and zero Neumann boundary conditions on a domain $\Omega_{\bar{q}}$ that can be changed by the optimization. Let the optimization objective be the squared H_1 seminorm of the solution on the domain. Then

- $q = [\bar{q}, c]$,

- The PDE functional is

$$\mathcal{H}(u, q, w) = \mathcal{H}(u, \bar{q}, c, w) = \int_{\Omega_{\bar{q}}} c \nabla u \nabla w - f w \, dx$$

.

- The objective is

$$J(u, \bar{q}) = \int_{\Omega_{\bar{q}}} \|\nabla u\|^2 \, dx$$

Discrete problem. Once the problem is discretized in a FE basis, we obtain a discrete problem:

$$\min_{\mathbf{q}} J(\mathbf{u}, \mathbf{q}), \text{ s.t., } \mathbf{h}(\mathbf{u}, \mathbf{q}) = 0 \quad (3.3)$$

where \mathbf{u} is the vector of FE basis coefficients of u and \mathbf{q} is the concatenation of the vectors of coefficients of $\bar{\mathbf{q}}, \mathbf{q}^1 \dots \mathbf{q}^m$, expanded in a suitable basis on their respective domains of definition. Notice that $\mathbf{h}(\mathbf{u}, \mathbf{q})$ is a vector valued function.

Example. For the variable-coefficient Poisson example above, discretized in FE basis with basis functions ϕ^ℓ , and with the shape function discretized with basis functions ξ^ℓ (generally different, but we can assume both to be piecewise linear for simplicity) on a triangulated domain $\Omega_{\text{ref}} \subset \mathbb{R}^2$, we have:

- $\mathbf{q} = [\bar{\mathbf{q}}, \mathbf{c}] = [x_1 \dots x_{n_N}, c_1 \dots c_{n_N}]$, where $x_\ell \in \mathbb{R}^2$ are vertices of the deformed domain $\Omega_{\bar{\mathbf{q}}}$, and c_ℓ are the coefficients of c in FE basis;
- $q(x) = \sum_{\ell=1}^{n_N} x_\ell \xi^\ell$.
- The PDE discretization is

$$\mathbf{h}(\mathbf{u}, \mathbf{q}) = S(\mathbf{q})\mathbf{u} - M(\mathbf{q})\mathbf{f},$$

where $S(\mathbf{q})$ is the matrix with entries

$$S_{m\ell} = \sum c_k \int_{\Omega_{\bar{\mathbf{q}}}} \phi^k(x) \nabla \phi^m(x) \cdot \nabla \phi^\ell(x) dx, \quad M_{m\ell} = \int_{\Omega_{\bar{\mathbf{q}}}} \phi^m(x) \phi^\ell(x) dx$$

with $\phi^\ell = \hat{\phi}^\ell \circ \bar{q}^{-1}$ and $\hat{\phi}^\ell$ is the FE basis on the Ω_{ref} .

- The discrete objective is:

$$J(\mathbf{u}, \mathbf{q}) = \mathbf{u}^T T(\mathbf{q}) \mathbf{u}$$

where

$$T(\mathbf{q})_{m\ell} = \int_{\Omega_{\bar{\mathbf{q}}}} \nabla \phi^m(x) \cdot \nabla \phi^\ell(x) dx$$

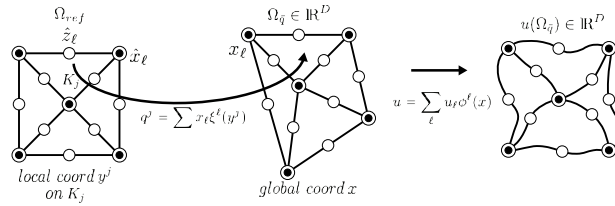


Figure 3.1: Notation for domains and maps we use, see Table 3.1.

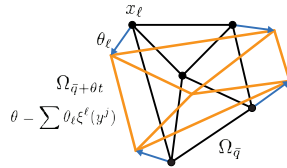


Figure 3.2: Domain perturbation θ , see Table 3.1.

3.4 ADJOINT-BASED OBJECTIVE DERIVATIVES IN GENERAL FORM

We now describe a general form of computing derivatives of an objective with respect to optimization parameters, assuming the problem is discretized e.g., using FEM spaces. We follow the adjoint approach, minimizing the computational expense.

First, consider the following Lagrangian for our constrained problem:

$$\mathcal{L} = J(\mathbf{u}, \mathbf{q}) \quad \{\text{objective term}\} \quad (3.4)$$

$$+ \mathbf{p}^T \mathbf{h}(\mathbf{u}, \mathbf{q}) \quad \{\text{physical constraint term}\} \quad (3.5)$$

We can compute the derivative with respect to parameters \mathbf{q} as:

$$d_{\mathbf{q}} \mathcal{L} \boldsymbol{\theta} = \partial_{\mathbf{q}} J \boldsymbol{\theta} + \partial_{\mathbf{u}} J \delta \mathbf{u} + \mathbf{p}^T \partial_{\mathbf{q}} \mathbf{h} \boldsymbol{\theta} + \mathbf{p}^T \partial_{\mathbf{u}} \mathbf{h} \delta \mathbf{u} \quad (3.6)$$

where $\delta \mathbf{u} = d_{\mathbf{q}} \mathbf{u} \boldsymbol{\theta}$ and where $\boldsymbol{\theta}$ is an arbitrary parameter perturbation vector, i.e., if parameters are changed by $\boldsymbol{\theta}$, up to higher order terms in $\|\boldsymbol{\theta}\|$, the functional changes by $d_{\mathbf{q}} \mathcal{L} \boldsymbol{\theta}$.

$d_{\mathbf{q}} \mathcal{L}$ is expensive to compute if the dimension of \mathbf{q} is large; a direct computation involves solving $\mathbf{h}(\mathbf{u}, \mathbf{q}) = 0$ once for each component of \mathbf{q} . In the adjoint approach only one solve is needed for any dimension of \mathbf{q} .

Isolating all terms multiplying $\delta \mathbf{u}$:

$$d_{\mathbf{q}} \mathcal{L} = \partial_{\mathbf{q}} J \boldsymbol{\theta} + \mathbf{p}^T \partial_{\mathbf{q}} \mathbf{h} \boldsymbol{\theta} + \left(\partial_{\mathbf{u}} J + \mathbf{p}^T \partial_{\mathbf{u}} \mathbf{h} \right) \delta \mathbf{u} \quad (3.7)$$

We can then eliminate the last term by finding \mathbf{p} such that $\partial_{\mathbf{u}} J + \mathbf{p}^T \partial_{\mathbf{u}} \mathbf{h} = 0$. This problem is called *the adjoint problem* and \mathbf{p} *the adjoint variable*:

$$\mathbf{p}^T \partial_{\mathbf{u}} \mathbf{h} = -\partial_{\mathbf{u}} J \quad (3.8)$$

Then, by plugging the solution \mathbf{p} of the adjoint PDE into the Lagrangian, we obtain the final shape

derivative:

$$d_{\mathbf{q}}J \boldsymbol{\theta} = d_{\mathbf{q}}\mathcal{L}(\mathbf{p}) = \partial_{\mathbf{q}}J \boldsymbol{\theta} + \mathbf{p}^T \partial_{\mathbf{q}}\mathbf{h} \boldsymbol{\theta} \quad (3.9)$$

Combining contributions from different forces together. Our discretized equation has the form

$$\sum_k \mathbf{h}^k(\mathbf{u}, \mathbf{q}) = 0$$

where \mathbf{h}^k is a contribution from each type of force (elasticity internal forces, contact forces, friction, and anything else we choose to add).

Similarly, the objective J is a sum of contributions from several objective components or constraints in penalty form:

$$J(\mathbf{u}, \mathbf{q}) = \sum_{\ell} J^{\ell}(\mathbf{u}, \mathbf{q})$$

Thus, the full parametric derivative has the form, for a perturbation $\boldsymbol{\theta}$:

$$d_{\mathbf{q}}J \boldsymbol{\theta} = \sum_{\ell} \partial_{\mathbf{q}}J^{\ell} \boldsymbol{\theta} + \sum_k \mathbf{p}^T \partial_{\mathbf{q}}\mathbf{h}^k \boldsymbol{\theta}$$

Similarly, the adjoint equation has the form

$$d_{\mathbf{q}}J \boldsymbol{\theta} = \sum_{\ell} \partial_{\mathbf{q}}J^{\ell} \boldsymbol{\theta} + \sum_k \mathbf{p}^T \partial_{\mathbf{q}}\mathbf{h}^k \boldsymbol{\theta}$$

$$\mathbf{p}^T \left(\sum_k \partial_{\mathbf{u}}\mathbf{h}^k \right) \boldsymbol{\psi} = - \sum_{\ell} \partial_{\mathbf{u}}J^{\ell} \boldsymbol{\psi}, \forall \boldsymbol{\psi} \quad (3.10)$$

Thus, for each force-optimization parameter pair, we need to get the expressions for

$$B^k(\mathbf{p}, \boldsymbol{\theta}) = \mathbf{p}^T \partial_{\mathbf{q}} \mathbf{h}^k \boldsymbol{\theta}, \quad A^k(\mathbf{p}, \boldsymbol{\psi}) = \mathbf{p}^T \partial_{\mathbf{u}} \mathbf{h}^k \boldsymbol{\psi} \quad (3.11)$$

for $\boldsymbol{\theta}$ going over basis vectors for this parameter type, \mathbf{p} going over adjoint variable components, and $\boldsymbol{\psi}$ over the test function basis vectors for the adjoint; i.e., two matrices of size $D_s n_N^z \times D_s n_N^z$.

For each objective-optimization parameter pair, $\partial_{\mathbf{q}} J^\ell \boldsymbol{\theta}$ and $\partial_{\mathbf{u}} J^\ell \boldsymbol{\psi}$, i.e., two vectors of size $D_s n_N^z$.

3.5 TIME-DEPENDENT PROBLEMS

In this section, we show how to transition from the adjoint equation for the static equation to the time-dependent. We do this in general form, only assuming that the force terms depend on solution and optimization parameters, but not explicitly on time.

Problem setup. We consider the following problem, already discretized in space but not yet in time:

$$\dot{\mathbf{u}} = \mathbf{v}; \quad M(\mathbf{q}) \dot{\mathbf{v}} = \mathbf{h}(\mathbf{u}, \mathbf{q}); \quad \mathbf{u}(0) = \mathbf{g}^u(\mathbf{q}); \quad \mathbf{v}(0) = \mathbf{g}^v(\mathbf{q})$$

where $M(\mathbf{q})$ is the mass matrix, which may also depend on parameters p .

For time discretization we use a BDF scheme of order m :

$$\dot{\mathbf{u}} \approx \frac{1}{\beta \Delta t} (\mathbf{u}^i + \sum_{j=1}^{\min(i,m)} \alpha_j^i \mathbf{u}^{i-j})$$

In general, α_j^i does not depend on i , except at the first $m - 1$ steps, when a higher-order scheme needs to be initialized with lower-order steps; more specifically, α_j^i is j -th coefficient of BDF i , for $1 \leq i < m$, and j -th coefficient of BDF m otherwise.

In the formulation above, $\mathbf{h}(\mathbf{u}, \mathbf{q})$ does not depend on velocities \mathbf{v} . If the dependence on velocities is needed, we discretize in time, and handle it as dependence on \mathbf{u} at different time steps; in the physical problem we solve, this happens only for damping forces.

The discrete system has the form

$$\begin{aligned} \mathbf{u}^i + \sum_{j=1}^{\min(i,m)} \alpha_j^i \mathbf{u}^{i-j} &= \beta_i \Delta t \mathbf{v}^i \\ M \left(\mathbf{v}^i + \sum_{j=1}^{\min(i,m)} \alpha_j^i \mathbf{v}^{i-j} \right) &= \beta_i \Delta t \mathbf{h}^i(\mathbf{u}^i, \mathbf{u}^{i-1}, \mathbf{q}) = \hat{\mathbf{h}}^i \end{aligned} \quad (3.12)$$

where M is the mass matrix.

Objectives. For time-dependent problems, we consider functionals of the form

$$J(\mathbf{u}, \mathbf{q}) = \int_{t=0}^T J(\mathbf{u}, t, \mathbf{q}) dt \quad (3.13)$$

where $J(\mathbf{u}, t, \mathbf{q})$ is a spatial functional, e.g., integral over the solid $\Omega_{\bar{\mathbf{q}}}(t)$ or its surface of some pointwise quantity depending on the solution and/or its derivatives pointwise. In discretized form,

$$J(\mathbf{u}, \mathbf{q}) = \sum_{i=0}^N w_i J_i(\mathbf{u}^i, \mathbf{q}) = \sum_{i=0}^N \hat{J}^i \quad (3.14)$$

where w_i are quadrature weights (e.g., all dt in the simplest case), and N is the number of time steps.

Remark on notation. We omit most of the explicit arguments in functions \mathbf{h} and J used in the expressions, to make the formulas more readable. The following is implied: $\mathbf{h}(\mathbf{u}^i, \mathbf{u}^{i-1}, \mathbf{q}, t_i) = \mathbf{h}_i(\mathbf{u}^i, \mathbf{q}) = \mathbf{h}_i$ and similarly for J_i .

Summary. Computing the derivative $d_q J$ requires the following components

- Derivatives $\partial_u J_i$, $\partial_u \mathbf{h}_i$, $\partial_q J_i$ and $\partial_q \mathbf{h}_i$ for each time step i . See Sections 3.7 to 3.8.2 for corresponding formulas.
- $\partial_q \mathbf{g}^u$ and $\partial_q \mathbf{g}^v$, derivatives of the initial conditions. See Section 3.5.3.

To compute the parametric derivative of J , the steps are as follows:

- Solve the forward system (3.12), and store the resulting solutions $\mathbf{u}^i, \mathbf{v}^i$, $i = 0 \dots N$ at every step.
- Initialize adjoint variables $\mathbf{p}_N, \mathbf{v}_N$ by solving (3.17) and (3.20).
- Perform backward time stepping using (3.16) (general BDF: (3.19)).
- At every step, evaluate derivative of the mass matrix $d_q M$, if applicable, and use formulas (3.18) and (3.22) to update $d_q J$.

3.5.1 IMPLICIT EULER

Discrete time-dependent Lagrangian. We write the time-dependent Lagrangian \mathcal{L} for the functional J viewing the equations for $\dot{\mathbf{v}}$ and $\dot{\mathbf{u}}$ as constraints with Lagrange multipliers \mathbf{p} and $\boldsymbol{\mu}$.

For the solution (\mathbf{u}, \mathbf{v}) for any optimization parameter values, the constraints are satisfied, $d_q J = d_q \mathcal{L}$, as the constraint terms identically vanish. The goal of introducing the adjoint variables is to eliminate the direct dependence of $d_q J$ on the displacement and velocity derivatives: $d_q \mathbf{u}^i$ or $d_q \mathbf{v}^i$. To achieve our objective, we expand the derivative $d_q \mathcal{L}$, and isolate the terms multiplying $d_q \mathbf{u}$ and $d_q \mathbf{v}$. By setting the sum of each of these two sets of terms to zero (which corresponds to our adjoint equations), we can find \mathbf{p} and $\boldsymbol{\mu}$ so that the derivative of the functional $d_q J$ does not directly depend on $d_q \mathbf{u}^i$ or $d_q \mathbf{v}^i$.

For clarity, we start with the simplest case, implicit Euler/BDF1:

$$\begin{aligned}\mathbf{u}^i - \mathbf{u}^{i-1} &= \Delta t \mathbf{v}^i \\ M(\mathbf{v}^i - \mathbf{v}^{i-1}) &= \Delta t \mathbf{h}^i = \hat{\mathbf{h}}^i\end{aligned}\tag{3.15}$$

In this case, the Lagrangian \mathcal{L} has the form

$$\begin{aligned}\mathcal{L} &= \sum_{i=0}^N \hat{f}^i && \{\text{objective terms}\} \\ &+ \mathbf{p}_0^T (\mathbf{v}^0 - \mathbf{g}^v) + \boldsymbol{\mu}_0^T (\mathbf{u}^0 - \mathbf{g}^u) && \{\text{initial condition terms}\} \\ &+ \sum_{i=1}^N \mathbf{p}_i^T (M(\mathbf{v}^i - \mathbf{v}^{i-1}) - \hat{\mathbf{h}}^i) + \boldsymbol{\mu}_i^T (\mathbf{u}^i - \mathbf{u}^{i-1} - \Delta t \mathbf{v}^i) && \{\text{PDE terms}\}\end{aligned}$$

Introducing a new variable $\boldsymbol{\nu}$ satisfying $\boldsymbol{\mu} = M^T \boldsymbol{\nu}$, adjoint equations in this case have the following form (see supplementary material for derivation):

$$\begin{aligned}\mathbf{p}_i - \mathbf{p}_{i+1} &= \Delta t \mathbf{v}_i \\ M^T (\mathbf{v}_i - \mathbf{v}_{i+1}) &= (\partial_{\mathbf{u}^i} \hat{\mathbf{h}}^i)^T \mathbf{p}_i + (\partial_{\mathbf{u}^i} \hat{\mathbf{h}}^{i+1})^T \mathbf{p}_{i+1} - (\partial_{\mathbf{u}^i} \hat{f}^i)^T\end{aligned}\tag{3.16}$$

Note that this systems is very similar to the forward time-stepping, with the following differences: it proceeds backwards, from \mathbf{v}_{i+1} to \mathbf{v}_i ; there is a single *linear* solve per time step, rather than a nonlinear solve as for the forward system. If we introduce \mathbf{p}_{N+1} and \mathbf{v}_{N+1} , with actual initial values \mathbf{p}_N and \mathbf{v}_N obtained from (3.16), the initial conditions can be taken to have a particularly simple form:

$$\begin{aligned}\mathbf{p}_{N+1} &= 0 \\ \mathbf{v}_{N+1} &= 0\end{aligned}\tag{3.17}$$

For $\mathbf{p}_0, \mathbf{v}_0$, which are a special case, as these correspond to initial conditions, and satisfy different equations compared to the rest, we have $M^T \mathbf{v}_0 = -(\partial_{\mathbf{u}^0} \hat{J}^0)^T + M^T \mathbf{v}_1 + \mathbf{p}_1^T \partial_{\mathbf{u}^0} \hat{\mathbf{h}}^1$ and $\mathbf{p}_0 = M^T \mathbf{p}_1$

Finally, the expression for $d_{\mathbf{q}}J$ is

$$\begin{aligned} d_{\mathbf{q}}J &= \partial_{\mathbf{q}} \hat{J}^0 - \mathbf{p}_0^T \partial_{\mathbf{q}} \mathbf{g}^v - \boldsymbol{\mu}_0^T \partial_{\mathbf{q}} \mathbf{g}^u - \mathbf{p}_1^T \partial_{\mathbf{q}} M \mathbf{v}^0 \\ &+ \sum_{i=1}^N \partial_{\mathbf{q}} \hat{J}^i - \mathbf{p}_i^T \partial_{\mathbf{q}} \hat{\mathbf{h}}^i + \Delta t \mathbf{v}_i^T \partial_{\mathbf{q}} M \mathbf{v}^i \end{aligned} \quad (3.18)$$

3.5.2 GENERAL BDF TIME INTEGRATION

For completeness, we include equations for BDFn adjoint system, also derived in supplementary material:

$$\begin{aligned} \left(\mathbf{p}_i + \sum_{j=1}^{\min(m, N-i)} \alpha_j^{i+j} \mathbf{p}_{i+j} \right) &= \beta_i \Delta t \mathbf{v}_i \\ M^T \left(\mathbf{v}_i + \sum_{j=1}^{\min(m, N-i)} \alpha_j^{i+j} \mathbf{v}_{i+j} \right) &= \\ (\partial_{\mathbf{u}^i} \hat{\mathbf{h}}^i)^T \mathbf{p}_i + (\partial_{\mathbf{u}^i} \hat{\mathbf{h}}^{i+1})^T \mathbf{p}_{i+1} - (\partial_{\mathbf{u}} \hat{J}^i)^T \end{aligned} \quad (3.19)$$

Similarly to the BDF1 case, by introducing $\mathbf{p}_{N+1}, \mathbf{v}_{N+1}$, the initial condition can be simplified as

$$\begin{aligned} \mathbf{p}_{N+1} &= 0 \\ \mathbf{v}_{N+1} &= 0 \end{aligned} \quad (3.20)$$

and similarly, the first (last in the adjoint solve) values need to be treated separately:

$$\boldsymbol{\mu}_0 = -(\partial_{\mathbf{u}} \hat{J}^0)^T - \sum_{j=1}^m \alpha_j^j \boldsymbol{\mu}_j + \mathbf{p}_1^T \partial_{\mathbf{u}^0} \hat{\mathbf{h}}^1, \quad \mathbf{p}_0 = - \sum_{j=1}^m \alpha_j^j M^T \mathbf{p}_j \quad (3.21)$$

Computing the derivative of J from the forward and adjoint solutions. From the adjoint variables, we can compute $d_q J = d_q \mathcal{L}$:

$$\begin{aligned}
d_q J = & -\mathbf{p}_0^T \partial_q \mathbf{g}^v - \boldsymbol{\mu}_0^T \partial_q \mathbf{g}^u \\
& + \sum_{i=0}^N \partial_q \hat{J}^i \\
& + \sum_{i=1}^N -\mathbf{p}_i^T \partial_q \hat{\mathbf{h}}^i + \beta_i \Delta t \mathbf{v}_i^T d_q M \mathbf{v}^i \\
& + \sum_{j=1}^m \alpha_j^j \mathbf{p}_j^T d_q M \mathbf{v}^0
\end{aligned} \tag{3.22}$$

Partial derivatives $\partial_q \hat{\mathbf{h}}, \partial_u \hat{\mathbf{h}}$ and $\partial_q \hat{J}_i, \partial_u \hat{J}_i$ are exactly the same as used in the construction of the system for static adjoint and computation of the functional. The differences, specific to time discretization, are:

- Mass matrix derivative $d_q M$. See supplemental material.
- Partial derivatives of the initial conditions with respect to parameters $\partial_q \mathbf{g}^v$ and $\partial_q \mathbf{g}^u$, for positions and velocities. See Section 3.5.3. Typically, a 3d position and velocity for the whole object (or angular velocity for the object rotating as a rigid body) are used as parameters, so these are trivial to compute.

3.5.3 INITIAL CONDITION DERIVATIVES

We need to compute partial derivatives of the initial conditions with respect to optimization parameters \mathbf{q} : $\partial_q \mathbf{g}^v$ and $\partial_q \mathbf{g}^u$, for positions and velocities. Notice that both $\partial_q \mathbf{g}^v$ and $\partial_q \mathbf{g}^u$ are discrete vector fields on domain $\Omega_{\bar{\mathbf{q}}}$. Consider we have one vector value \mathbf{q}^m per node of the domain.

If $(\mathbf{g}^v)_s = \mathbf{q}^m$, where $(\mathbf{g}^v)_s$ is initial condition at node s , the derivative with respect to q_m is simply the identity matrix ($\partial_{q^m} \mathbf{g}^v = I$). At the same time, it is the zero matrix w.r.t. any other q^{m^*} , with $m^* \neq m$. Same thing goes for \mathbf{g}^u .

3.6 PHYSICAL MODEL AND DISCRETIZATION

In this section, we summarize the physical model we use. The model is similar to the one used in [Li et al. 2020a], with some modifications, most importantly, addition of damping.

On discretization side, we use arbitrary-order Lagrangian elements and arbitrary-order BDF time stepping (although above order 6, BDF time stepping is not zero-stable).

The forces, which contribute to the PDE and need to be included in the adjoint equations and corresponding parametric gradient terms are:

- geometrically non-linear elasticity (with linear and neo-Hookean constitutive laws as options);
- contact forces in smoothed IPC formulation;
- friction forces also in smoothed IPC formulation;
- strain-rate proportional viscous damping for elastic objects;
- external forces such as gravity or surface loads.

Additionally, we support contact with stationary obstacles.

The right-hand side of the system of equations we solve on i -th time step can be written as

$$\begin{aligned} & \mathbf{h}^e(\mathbf{u}^i; \lambda(x), \mu(x)) + \mathbf{h}^c(\mathbf{u}^i) + \mathbf{h}^f(\mathbf{u}^i, \mathbf{u}^{i-1}; \mu(x, y)) \\ & + \mathbf{h}^d(\mathbf{u}^i, \mathbf{u}^{i-1}; \alpha(x), \beta(x)) \end{aligned}$$

where \mathbf{h}^e is the discrete elastic PDE term, \mathbf{h}^c and \mathbf{h}^f define contact and friction forces and \mathbf{h}^d defines damping. In greater detail, all these forces are defined in the next section, along with $\partial_{\mathbf{u}}\mathbf{h}$ and $\partial_{\mathbf{q}}\mathbf{h}$ for each.

The physical parameters \mathbf{q} of the model, with respect to which it can be differentiated, include: (possibly spatially variant) Lamé coefficients for elasticity $\lambda(x), \mu(x)$, friction coefficient between pairs of points $\mu(x, y)$ (we consider it fixed for each pair of objects, to reduce the number of variables involved), damping coefficients $\alpha(x), \beta(x)$, and all external forces.

3.7 PARAMETRIC DERIVATIVES OF FORCES

In this section, we derive expressions for $\partial_{\mathbf{u}}\mathbf{h}$ and $\partial_{\mathbf{q}}\mathbf{h}$ for specific forces needed for the adjoint equations and the final derivative formula respectively.

3.7.1 OBTAINING SHAPE DERIVATIVES ON PERTURBED DOMAIN

While it is conceptually straightforward to plug in FE basis expressions for $u = u \circ \bar{\mathbf{q}}$, and geometric maps $\bar{\mathbf{q}}$ into the PDE and functionals, obtaining expressions for all quantities as functions on the fixed reference domain Ω_{ref} which can be differentiated with respect to all parameters, doing this directly yields unwieldy expressions in the case of shape derivatives.

For this reason, we perform most of the derivations for shape derivatives considering functions on $\Omega_{\bar{\mathbf{q}}}$, where we can perform all manipulations in coordinates on $\Omega_{\bar{\mathbf{q}}}$ and substitute expressions

involving the basis on Ω_{ref} in the end. If the domain does not change, approaches using Ω_{ref} and $\Omega_{\bar{\mathbf{q}}}$ are similar in complexity.

To compute $\partial_{\mathbf{q}} h$, or $\partial_{\mathbf{q}} J$, we consider the perturbed domain $\Omega_{\bar{\mathbf{q}}+\theta\epsilon}$, and compute the full derivative as limit of, e.g.

$$\frac{1}{\epsilon}(J(u_{\bar{\mathbf{q}}+\theta\epsilon}, \bar{\mathbf{q}} + \theta\epsilon) - J(u_{\bar{\mathbf{q}}}, \bar{\mathbf{q}}))$$

as $\epsilon \rightarrow 0$. In the resulting expression, the terms not containing derivative $\delta u(x)$ correspond to $\partial_{\mathbf{q}} J\theta$, and the terms containing derivatives of $u(x)$ are transformed to $\partial_{\mathbf{u}} J\psi$ by substituting ψ instead of $\delta u(x)$.

3.7.2 VOLUME FORCES

Many forces in continuum mechanics have the general weak form

$$\mathcal{H}^v(u, w, q) = \int_{\Omega_{\bar{\mathbf{q}}}} f^v((\nabla u(x), q(x)) : \nabla w \, dx \quad (3.23)$$

with the components of the vector $\mathbf{h}^v(\mathbf{u})$ obtained as $\mathcal{H}^v(u, \phi^\ell, q)$, for all basis functions ϕ^ℓ . In our case, elastic forces, irrespective of the constitutive law used, belongs to this category.

In these expressions $f^v(\nabla u, q)$ is a tensor of dimension $D_d \times D_s$; e.g., for elasticity, $D_d = D_s$, and this expression is the stress tensor, as function of ∇u .

If the force is associated with a volume energy density $W^v(\nabla u, q)$ or surface density $W^s(u, q)$, associated forces have the form above, specifically, $f^v(\nabla u, q) = \nabla_1 W^v$. (Here, ∇_1 means the gradient with respect to the first parameter, which in this case is ∇u).

We also formulate damping in a similar way, as explained in more detail below, except at each timestep W^v depends on displacements u^i and u^{i-1} at the current and previous time steps. The formulas for A^v and B^v in this case are obtained in exactly the same way as for dependence on u^i

only, separately for u^i and u^{i-1} , corresponding to $\partial_{\mathbf{u}_i} \mathbf{h}^i$ and $\partial_{\mathbf{u}_{i-1}} \mathbf{h}^i$ respectively.

To obtain matrices A^v and B^v corresponding to $\partial_{\mathbf{u}} \mathbf{h}^v$ and $\partial_{\mathbf{q}} \mathbf{h}^v$ (3.11), we split $\partial_{\mathbf{q}} \mathbf{h}^f$ into $\partial_{\bar{\mathbf{q}}} \mathbf{h}^f$ and $\partial_{\mathbf{q}^1} \mathbf{h}^f$, the shape and non-shape parameter derivatives, assuming f depends on a single volume vector of parameters $q = \mathbf{q}^1$ (e.g., Lamé constants). We treat these two types of parameters separately, as $\bar{\mathbf{q}}$ affects the domain of integration but not the integrand, and conversely, q affects the integrand but not the domain.

Shape derivatives. For the shape derivative contribution, we obtain the following forms (the derivation and explicit form of matrix entries can be found in supplementary material).

$$\begin{aligned} B^v(\theta, p) = & \int_{\Omega_{\bar{\mathbf{q}}}} -f(\nabla u) \nabla \theta^T : \nabla p \\ & - (\nabla_1 f(\nabla u) : (\nabla u \nabla \theta)) : \nabla p + (f(\nabla u) : \nabla p) \nabla \cdot \theta \, dx \end{aligned} \quad (3.24)$$

$B^v(\theta, p)$ is linear in θ and p , and we convert it to a matrix form by substituting basis functions for θ and p .

The contribution to the left-hand side of the adjoint equation is

$$A^v(\psi, p) = \int_{\Omega_{\bar{\mathbf{q}}}} (\nabla_1 f(\nabla u) : \nabla \psi) : \nabla p \, dx \quad (3.25)$$

Non-shape volumetric parameter derivatives. We assume that the force depends on $q = q(x)$, a function of the point in $\Omega_{\bar{\mathbf{q}}}$, defined by its values \mathbf{q} at the same nodes as the solution, and interpolated using the same basis ϕ .

In this case, the relevant forms are:

$$B^v(\theta, p) = \int_{\Omega} (\partial_q f \cdot \theta) : \nabla p \, dx$$

The contribution to the left-hand side of the adjoint equation is identical to the shape derivative case.

In our implementation we consider two versions of elastic forces, both defined by Lamé parameters specified as functions on Ω_{ref} : $q(x) = [\lambda(x), \mu(x)]$. The only quantities we need are derivatives of $f(\nabla u)$ with respect to ∇u , and material parameters.

Linear elasticity. For linear elasticity, we replace f^v with

$$f^e(\nabla u, \mathbf{q}) = \sigma(\nabla u, \mathbf{q}) = C(\mathbf{q}) : \varepsilon(\nabla u) = \frac{1}{2} C(\mathbf{q}) : (\nabla u^T + \nabla u),$$

with $C_{ijkl}(\lambda, \mu) = \lambda \delta_{ij} \delta_{kl} + \mu(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})$.

For computing A^e and B^e we need partial derivatives of f^e with respect to relevant parameters:

$$\nabla_1 f^e(\nabla u, \boldsymbol{\lambda}, \boldsymbol{\mu}) = C$$

$$\partial_{\lambda} f^e(\nabla u, \boldsymbol{\lambda}, \boldsymbol{\mu})_{ij} = \delta_{ij} \delta_{kl} \varepsilon_{kl}$$

$$\partial_{\mu} f^e(\nabla u, \boldsymbol{\lambda}, \boldsymbol{\mu})_{ij} = (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \varepsilon_{kl}$$

Neo-Hookean elasticity. For Neo-Hookean elasticity, the following formula is used for computing stress:

$$f^e(\nabla u, \mathbf{q}) = \mu(F(\nabla u) - Q(\nabla u)) + \lambda \log(\det(F(\nabla u))) Q(\nabla u)$$

where $F(\nabla u) = \nabla u + I$ and $Q(\nabla u) = F(\nabla u)^{-T}$.

We can then compute derivatives of $f(\nabla u)$

$$\nabla_1 f^e(\nabla u, \mathbf{q})_{ijkl} = \mu(\delta_{ik}\delta_{jl} + Q_{il}Q_{kj}) + \lambda(Q_{ij}Q_{kl} - \log(\det(F)) Q_{il}Q_{kj})$$

$$\partial_\lambda f^e(\nabla u, \mathbf{q}) = F(\nabla u) - Q(\nabla u)$$

$$\partial_\mu f^e(\nabla u, \mathbf{q}) = \log(\det(F(\nabla u))) Q(\nabla u)$$

Damping. For damping, we have material parameters controlling shear and bulk damping α, β .

We use the strain-rate proportional damping described in [Brown et al. 2018]. Given deformation gradient $F = \nabla u + I$, the Green strain tensor $E = \frac{1}{2}(F^T F - I)$ is rotation-invariant. The viscous Piola-Kirchhoff stress is of the form

$$P(\nabla u, \nabla \dot{u}) = F(2\alpha \dot{E} + \beta \text{Tr}(\dot{E})I)$$

where \dot{E} denotes the time derivative, and the weak form of the corresponding force

$$\mathcal{H}^d(u, \dot{u}, w) = \int_{\Omega_{\bar{q}}} P(\nabla u, \nabla \dot{u}) \nabla w dx$$

In our case, to fit this force into our differentiable formulation, we discretize \dot{F} using as $\dot{F}^i = \frac{1}{\Delta t}(F^i - F^{i-1})$; this yields a force expression of the form

$$\mathcal{H}^d(u^i, u^{i-1}, w) = \int_{\Omega_{\bar{q}}} P(\nabla u^i, \nabla u^{i-1}) \nabla w dx$$

which is identical to (3.23), except it depends on both ∇u^i and ∇u^{i-1} .

As a consequence, expressions for $A^d(\psi, p)$ and $B^d(\theta, p)$ are obtained in the same way as in (3.25) and (3.24), except two pairs of matrices are obtained, one for ∇u^i the other for ∇u^{i-1} , using $\nabla_1 P$ and $\nabla_2 P$ as $\nabla_1 f$ respectively.

Contact. The contact forces are added to the formulation using the formulation of [Li et al. 2020a], with a small change. While the original formulation is introduced in a discrete form, with minimal changes, [Li et al. 2022] it can be derived as a linear finite-element discretization of a continuum formulation.

The contact formulation uses log barrier function $b(y)$, where b is a truncated log barrier function, approaching infinity, if $y \rightarrow 0$, and vanishing for $y \geq \hat{d}$ for some small distance d . For any pair of primitives k (vertices, edges and faces) of the surface mesh $\partial\Omega_{\bar{\mathbf{q}}}$, the distance $d_k(\mathbf{x}^d)$ is computed. C denotes the set of primitive pairs in contact, i.e., pairs of primitives with $d_k < \hat{d}$.

The contact forces are derived from the following potential, which can be obtained as a discretization of an integral over the surface:

$$E(\mathbf{u}, \bar{\mathbf{q}}) = \kappa \sum_{k \in C} b(\underbrace{d_k(\bar{\mathbf{q}} + \mathbf{u})}_{\mathbf{x}^d}) A_k = \sum_{k \in C} W_k(\mathbf{u}, \bar{\mathbf{q}}) A_k,$$

where $\kappa > 0$ is a conditioning parameter controlling the barrier stiffness and A_k corresponds to the sum of surface areas associated with each primitive in k (1/3 of the sum of areas of incident triangles for vertices and edges, and the area for triangles); distances d_k depend on the initial positions $x \in \Omega_{\bar{\mathbf{q}}}$.

We define $F_k^c(\mathbf{u}, \bar{\mathbf{q}}) = \partial_{\mathbf{u}} W_k(\mathbf{u}, \bar{\mathbf{q}}) = \kappa b'(d_k(\mathbf{u} + \bar{\mathbf{q}})) \partial_{\mathbf{x}^d} d_k$.

The contact force is then

$$\mathbf{h}^c = \sum_{k \in C} F_k^c(\mathbf{u}, \bar{\mathbf{q}}) A_k$$

The terms B^c and A^c have the form

$$B^c(p, \theta) = \sum_k (\partial_{\bar{\mathbf{q}}} F_k^c \theta \cdot p + F_k^c \cdot p \partial_{\bar{\mathbf{q}}} A_k) A_k$$

$$A^c(p, \psi) = \sum_k \partial_{\mathbf{u}} F_k^c \psi \cdot p A_k$$

where

$$\partial_{\mathbf{u}} F_k^c = \kappa(b''(\partial_{\mathbf{x}^d} d_k)(\partial_{\mathbf{x}^d} d_k)^T + b' \partial_{\mathbf{x}^d}(\partial_{\mathbf{x}^d} d_k))$$

$$\partial_{\bar{\mathbf{q}}} F_k^c = \partial_{\mathbf{u}} F_k^c M^*$$

and $\partial_{\bar{\mathbf{q}}} A_k$ corresponds to the gradient of the area term, which varies depending on the type of primitive pairs corresponding to k .

The matrix M^* is upsampling matrix to bring dimension of $\bar{\mathbf{q}}$ to the same as discrete solution \mathbf{u} , as the surface geometry \bar{q} is represented in the basis ξ_ℓ , for which we always use piecewise-linear elements, and u can be represented in a Lagrangian basis of any order. The upsampling is performed by linear interpolation from \hat{x}_ℓ to nodes \hat{z}_ℓ .

Friction. In general, the friction coefficient $\gamma(x_1, x_2)$ is a function of surface material points in $\partial\Omega_{\bar{\mathbf{q}}}$. As a simplification, in our implementation, we assume that each pair of objects (m, n) , in the simulation has a single coefficient $\gamma_{m,n}$, which can vary through the optimization. To simplify notation, we use γ_{k_1, k_2} for a pair of primitives k_1 and k_2 to indicate the friction coefficient between objects these primitives belong to.

We follow IPC definition of friction [Li et al. 2020a]. The key feature it has for our purposes is that it is a differentiable function of displacements, which determine the contact forces, and relative

velocities, which we discretize in this formulation by $u^{i-1} - u^i$, where i is the time step, so the function depends on displacements at *two* sequential time steps.

The friction force for each active pair of primitives k is defined as:

$$F_k^f(\mathbf{u}^{i-1}, \mathbf{u}^i) = -\gamma_{k_1, k_2} N_k T_k f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} \quad (3.26)$$

where N_k is the contact force magnitude, T_k is a tangential frame matrix, constructed from as described in [Li et al. 2020a], and $\tau_k = \tau_k(\mathbf{u}^{i-1}, \mathbf{u}^i)$ is the relative tangential sliding defined below.

$$\mathbf{h}^f = \sum_{k \in C} F_k^f(\mathbf{u}, \bar{\mathbf{q}}) A_k$$

It is very important to notice that our method for solving friction is implemented with a lagging formulation, where solution u_{lagged} from previous solution of the Newton method solve is used as input for both T_k and N_k (together with the contacting set), in order to obtain current time step solution u^i . In our time dependent case, we always use a single lagged step and assume $u_{\text{lagged}} = u^{i-1}$. See below the formula showing how each term depends on u^{i-1} and u^i .

$$\tau_k = T_k(\underbrace{\bar{\mathbf{q}} + \mathbf{u}^{i-1}}_{\mathbf{x}^p})^T (\mathbf{u}^i - \mathbf{u}^{i-1})$$

$$f_\eta(y) = \begin{cases} -\frac{y^2}{\eta^2} + \frac{2y}{\eta} & y \in [0, \eta) \\ 1 & y \geq \eta \end{cases}$$

To compute the shape derivative term $B^f = \sum_k (F_k^f)_x \theta \cdot p A_k + F_k^f \cdot p \partial_{\bar{\mathbf{q}}} A_k A_k$, we need to compute

the derivative of each f_k term with respect to components x_ℓ of $\bar{\mathbf{q}}$, obtaining:

$$\begin{aligned}
\partial_{x_\ell} F_k^f = & -\gamma_{k_1, k_2} T_k f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} \times \\
& \left(\kappa \frac{N_k}{\|N_k\|} \cdot (b''(\partial_{\mathbf{x}^d} d_k)(\partial_{\mathbf{x}^d} d_k)^T + b' \partial_{\mathbf{x}^d}(\partial_{\mathbf{x}^d} d_k)) M^* \right)_\ell + \\
& -\gamma_{k_1, k_2} N_k (\partial_{\mathbf{x}^p} T_k M^*)_ell f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} + \\
& -\gamma_{k_1, k_2} N_k T_k \frac{\tau_k}{\|\tau_k\|} \left(f'_\eta \frac{\tau_k}{\|\tau_k\|} \cdot ((\partial_{\mathbf{x}^p} T_k M^*)_ell^T (\mathbf{u}^i - \mathbf{u}^{i-1})) \right) + \\
& -\gamma_{k_1, k_2} N_k T_k f_\eta(\|\tau_k\|) \left(\left(\frac{I_2}{\|\tau_k\|} - \frac{\tau_k \tau_k^T}{\|\tau_k\|^3} \right) (\partial_{\mathbf{x}^p} T_k M^*)_ell^T (\mathbf{u}^i - \mathbf{u}^{i-1}) \right)
\end{aligned} \tag{3.27}$$

The derivative with respect to friction coefficient values is easily obtained as the force is linear in friction coefficients. If \mathbf{q} is a vector of friction coefficients,

$$\partial_{q_\ell} F_k^f = \begin{cases} -N_k T_k f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} & \text{if } q_\ell \text{ corresponds to } \gamma_{k_1, k_2} \\ 0 & \text{otherwise} \end{cases}$$

To compute the term on the left side of the adjoint equation ($A^f(\psi, p) = \sum_k (F_k)_u \psi \cdot p A_k$), we need to compute the derivative of each F_k term with respect to u^i and u^{i-1} :

$$\begin{aligned}
\partial_{u^i} F_k^f = & -\gamma_{k_1, k_2} N_k T_k \frac{\tau_k}{\|\tau_k\|} \left(f'_\eta \frac{\tau_k}{\|\tau_k\|} \cdot (T_k^T)_\ell \right) + \\
& -\gamma_{k_1, k_2} N_k T_k f_\eta(\|\tau_k\|) \left(\left(\frac{I_2}{\|\tau_k\|} - \frac{\tau_k \tau_k^T}{\|\tau_k\|^3} \right) (T_k^T)_\ell \right)
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
\partial_{u_\ell^{i-1}} F_k^f &= -\gamma_{k_1, k_2} T_k f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} \times \\
&\quad \left(\kappa \frac{N_k}{\|N_k\|} \cdot (b'' \nabla d_k \nabla d_k^T + b' \nabla^2 d_k) \right)_\ell + \\
&\quad - \gamma_{k_1, k_2} N_k (\partial_{\mathbf{x}^p} T_k)_\ell f_\eta(\|\tau_k\|) \frac{\tau_k}{\|\tau_k\|} + \\
&\quad - \gamma_{k_1, k_2} N_k T_k \frac{\tau_k}{\|\tau_k\|} \left(f'_\eta \frac{\tau_k}{\|\tau_k\|} \cdot (\partial_{\mathbf{x}^p} T_k)_\ell^T (\mathbf{u}^i - \mathbf{u}^{i-1}) \right) + \\
&\quad - \gamma_{k_1, k_2} N_k T_k f_\eta(\|\tau_k\|) \left(\left(\frac{I_2}{\|\tau_k\|} - \frac{\tau_k \tau_k^T}{\|\tau_k\|^3} \right) (\partial_{\mathbf{x}^p} T_k)_\ell^T (\mathbf{u}^i - \mathbf{u}^{i-1}) \right)
\end{aligned} \tag{3.29}$$

3.8 OBJECTIVE DERIVATIVES

In this section, we define $\partial_{\mathbf{q}} J$ and $\partial_{\mathbf{u}} J$ terms needed for gradient computation (3.9).

3.8.1 GENERAL FORMS OF OBJECTIVES

Consider general types of objectives, which will be used for a number of specific objectives in Section 3.8.2. This includes inequality constraints in penalty form.

For each different objective J^o , we obtain vectors $R^o(\psi)$ and $S^o(\theta)$ corresponding to $\partial_{\mathbf{u}} J^o$ and $\partial_{\mathbf{q}} J^o$, which are necessary to compute the adjoint solution and the final shape derivative. The vectors $\partial_{\mathbf{u}} J^o$ and $\partial_{\mathbf{q}} J^o$ are obtained by plugging in the basis functions ϕ_ℓ into R^o and S^o .

Objectives depending on gradient of solution and shape.

Consider an objective that depends on both the solution of the PDE and the domain :

$$J^o(\nabla u, \bar{q}) = \int_{\Omega_{\mathbf{q}}} j(\nabla u, x) dx \tag{3.30}$$

In this case, as derived in the supplementary document,

$$S^o(\theta) = \int_{\Omega_{\bar{q}}} -\nabla_1 j : \nabla u \nabla \theta + \nabla_2 j \cdot \theta + j(\nabla u, x) \nabla \cdot \theta \, dx \quad (3.31)$$

and

$$R^o(\psi) = \int_{\Omega_{\bar{q}}} \nabla_1 j : \nabla \psi \, dx \quad (3.32)$$

Objectives depending on solution and shape. We also an objective that depends on both the solution of the PDE and the domain :

$$J^o(u, \bar{q}) = \int_{\Omega_{\bar{q}}} j(u, x) dx \quad (3.33)$$

In this case,

$$S^o(\theta) = \int_{\Omega_{\bar{q}}} \nabla_2 j \cdot \theta + j(u, x) \nabla \cdot \theta \, dx \quad (3.34)$$

and,

$$R^o(\psi) = \int_{\Omega_{\bar{q}}} \nabla_1 j \cdot \psi \, dx \quad (3.35)$$

3.8.2 SPECIFIC OBJECTIVES

L_p norm of stress. For $p = 2$ this objective measures the overall average stress, and for high p ,

e.g., $p = 6$, L_p -norm of stress approximates maximal stress:

$$J^\sigma = \left(\int_{\Omega_{\bar{q}}} \|\sigma(\nabla u)\|_F^p dx \right)^{1/p} \quad (3.36)$$

where $\sigma(\nabla u) = f(\nabla u)$ represents stress, which depends on ∇u . Following the chain rule:

$$\partial_{\mathbf{q}} J^\sigma \theta = \frac{1}{p} ((J^\sigma)^p)^{\frac{1-p}{p}} \partial_{\mathbf{q}} (J^\sigma)^p \theta$$

Since $(J^\sigma)^p$ is on the form $\int_{\Omega_{\bar{q}}} j(\nabla u, x) dx$, we can use the expressions for the general objective (3.30), for which we just need the derivative $\nabla_1 j$, since $\nabla_2 j = 0$:

$$\nabla_1 j = p \|\sigma\|^{p-2} \sigma : \nabla f(\nabla u)$$

Weighted difference from target deformations.

$$J^{trj}(x, u) = \int_{\Omega_{\bar{q}}} w(g(x, \bar{q})) \|x^d - x^{trg}(g(x, \bar{q}))\|^2 dx \quad (3.37)$$

where $x^d = x + u$, the deformed state of the object, g is a function mapping the current position x to our original shape prior to optimization, weight w determines relative importance of points, and x^{trg} is the target configuration, defined as function on Ω_{ref} .

J^{trj} is a function of x and u , and using the formulas for the general objective (3.33), compute S^{trj} and R^{trj} as

$$\nabla_1 j = \nabla_2 j = 2w(g(x))(x^d - x^{trg}(x))$$

If we define only the shape on the boundary as the target, then we have:

$$J^{btrj} = \int_{\partial\Omega_{\bar{q}}} w(g(x, \bar{q})) \|x^d - x^{trg}(g(x, \bar{q}))\|^2 dx$$

Formulas for the derivatives follow the same idea, with:

$$\begin{aligned} S^{btrj} &= \int_{\partial\Omega_{\bar{q}}} \nabla_2 j \cdot \theta + j(u, x) \nabla_s \cdot \theta dx \\ R^{btrj} &= \int_{\partial\Omega_{\bar{q}}} \nabla_1 j \cdot \psi dx \end{aligned}$$

where ∇_s denotes the surface derivative.

Target center of mass trajectory. A related objective is the deviation of the center of mass of the object from a target trajectory:

$$J^{ctr} = \left\| \frac{\int_{\Omega_{\bar{q}}} \rho(x) x^d dx}{\int_{\Omega_{\bar{q}}} \rho(x) dx} - x^{ctr} \right\|^2 = \|\Delta^{ctr}\|^2 \quad (3.38)$$

Following the chain rule:

$$\partial_q J^{ctr} \theta = 2 \|\Delta^{ctr}\| \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \|\Delta^{ctr}\|$$

As $\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \|\Delta^{ctr}\| = \frac{\Delta^{ctr}}{\|\Delta^{ctr}\|} \cdot \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \Delta^{ctr}$, it is sufficient to compute $\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \Delta^{ctr}$:

$$\begin{aligned} \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \frac{\int_{\Omega_{\bar{q}}} \rho x^d dx}{\int_{\Omega_{\bar{q}}} \rho dx} - x^{ctr} &= \\ &= \frac{\int_{\Omega_{\bar{q}}} \rho dx \int_{\Omega_{\bar{q}}} \rho (\theta + \delta u) + \rho x^d \nabla \cdot \theta dx - \int_{\Omega_{\bar{q}}} \rho x^d dx \int_{\Omega_{\bar{q}}} \rho \nabla \cdot \theta dx}{\left(\int_{\Omega_{\bar{q}}} \rho dx \right)^2} \end{aligned}$$

Separating contributions for shape derivative and adjoint system, we have the following for Δ^{ctr} :

$$S^{ctr}(\theta) = \frac{\int_{\Omega_q} \rho \, dx \int_{\Omega_q} \rho \, \theta + \rho x^d \nabla \cdot \theta \, dx - \int_{\Omega_q} \rho \, x^d \, dx \int_{\Omega_q} \rho \nabla \cdot \theta \, dx}{\left(\int_{\Omega_q} \rho \, dx \right)^2}$$

$$R^{ctr}(\psi) = \frac{\int_{\Omega_q} \rho \, \psi \, dx}{\left(\int_{\Omega_q} \rho \, dx \right)^2}$$

Height. This functional aims to maximize the height of the center of mass:

$$J^{z_{\max}} = - \frac{\int_{\Omega_q} \rho(x) x_z^d \, dx}{\int_{\Omega_q} \rho(x) \, dx} \quad (3.39)$$

where u_z is the z (vertical) component of the solution (displacement) u , x_z is the z component of the original position x .

Using the quotient rule, we obtain:

$$S^{z_{\max}}(\theta) = \frac{- \int_{\Omega_q} \rho \, dx \int_{\Omega_q} \rho \theta_z + \rho x_z^d \nabla \cdot \theta \, dx + \int_{\Omega_q} \rho x_z^d \, dx \int_{\Omega_q} \rho \nabla \cdot \theta \, dx}{\left(\int_{\Omega_q} \rho \, dx \right)^2}$$

$$R^{z_{\max}}(\psi) = - \frac{\int_{\Omega_q} \rho \, \psi_z \, dx}{\int_{\Omega_q} \rho \, dx}$$

Upper bound for volume. This is a constraint on the volume, most relevant in shape optimization, defined in penalty form:

$$J^V = \varphi(V(\Omega_q) - V_t) = \varphi(\Delta^V) \quad (3.40)$$

where V corresponds to the volume of shape Ω_q ($\int_{\Omega_q} dx$), V_t to the target volume and $\varphi(z)$ is a

function that penalizes quadratically values above our target, as follows:

$$\varphi(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z^2 & \text{otherwise} \end{cases} \quad (3.41)$$

Our derivative is then:

$$\partial_{\mathbf{q}} J^V \theta = \varphi' \partial_{\mathbf{q}} \Delta^V \theta$$

Following the expressions for the general objective (3.33), we can compute the shape derivative for J^V by building vectors R^V and S^V with $\nabla_1 j = \nabla_2 j = 0$, since $j(u, x) = 1$.

Upper bound for stress. Similarly, we can impose an approximate upper bound on stress via a penalty:

$$J^{\sigma_t} = \int_{\Omega_{\bar{\mathbf{q}}}} \varphi(\|\sigma\| - s_t) dx \quad (3.42)$$

where s_t is the stress magnitude target. This means that we are penalizing quadratically elements with higher stress than expected. As our L_p stress energy, our integrand $\varphi(\|\sigma\| - s_t)$ depends only on ∇u and we have:

$$\begin{aligned} \nabla_1 j &= \varphi' \frac{f(\nabla u)}{\|\sigma\|} : \nabla f(\nabla u) \\ \nabla_2 j &= 0 \end{aligned}$$

Scale-invariant smoothing.

$$J^{\text{smooth}} = \sum_{i \in B} \|s_i\|^p \quad s_i = \frac{\sum_{j \in N(i) \cap B} (v_i - v_j)}{\sum_{j \in N(i) \cap B} \|v_i - v_j\|} \quad (3.43)$$

where B contains the indices of all boundary vertices, $N(i)$ contains the indices of all neighbor vertices of vertex i , and v_i is the position of vertex i . The value of p can be adjusted to obtain smoother surfaces at the cost of less optimal shapes, normally we use $p = 2$. This term is scale-invariant and pushes the triangles/tetrahedra of the mesh toward equilateral.

Material parameter spatial smoothing.

$$J^{\lambda, \mu \text{ smooth}} = \sum_{t \in T} \sum_{t' \in \text{Adj}(t)} \left(1 - \frac{\lambda_{t'}}{\lambda_t}\right)^2 + \left(1 - \frac{\mu_{t'}}{\mu_t}\right)^2 \quad (3.44)$$

where T is the set of all triangles/tetrahedra, $\text{Adj}(t)$ is the set of triangles/tetrahedra adjacent to t . λ, μ are the material parameters defined per triangle.

3.9 IMPLEMENTATION

FE Solver. We implemented our solver in C++ using the PolyFEM library [Schneider et al. 2019] for the forward solve, the IPC Toolkit [Ferguson et al. 2020] for computing contact and friction potentials, and Pardiso [Alappat et al. 2020; Bollhöfer et al. 2020, 2019] for solving linear systems.

Optimization. To minimize the objective function we use the L-BFGS implementation in [Wieschollek 2016], with backtracking line search. If the line search fails to find a valid descent step that decreases the energy, we reset the Hessian approximation to identity and take a gradient descent step instead.

Meshing. Shape optimization might negatively affect the elements shape, and for large defor-

mation introduce inverted elements. After every optimization iteration, we evaluate the element quality using the scaled Jacobian quality measure [Knupp 2001], and we optimize the mesh if it is below a threshold experimentally set to 10^{-3} .

For 2D examples, we keep the mesh boundary fixed and we regenerate the interior using GMSH [Geuzaine and Remacle 2009]. For 3D examples, we similarly fix the boundary and then use the mesh optimization procedure of fTetWild [Hu et al. 2020b] to improve the quality of the interior until its quality is above the threshold.

The reason for fixing the boundary is that our optimization objectives (Section 3.8.1) often depend on quantities on the boundary: if the boundary is remeshed, we will need a bijective map between the two boundaries. Meshing methods providing this map exist [Jiang et al. 2020], but their integration in our framework, while trivial from a formulation point of view, is an engineering challenge that we leave as future work.

Reproducibility. The reference implementation of our solver and applications will be released as an open source project.

3.10 RESULTS

We partition our results in three groups depending on the type of the dofs used in the objective function: shape, initial conditions, or material. For each group we provide a set of examples of static and dynamic scenes of increasing complexity. We run our experiments on a workstation with a Threadripper Pro 3995WX with 64 cores and 512 Gb of memory. For a selection of problems, we validate our results with physical experiments using items fabricated in silicon rubber (we use 1:1 SMOOTH-ON OOMOO 30 poured into a 3D printed PVA mold) or 3D printed PLA plastic.

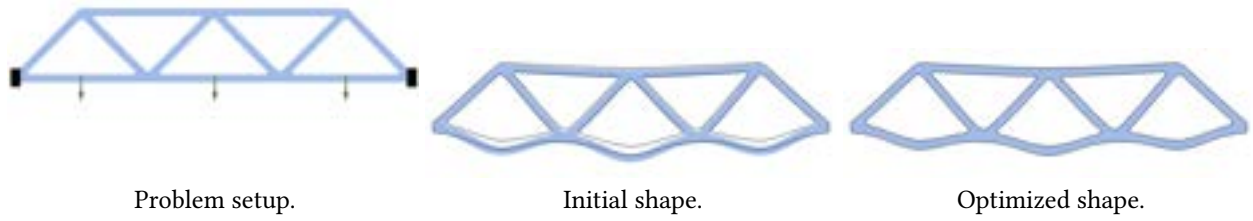


Figure 3.3: The result of the shape optimization (blue surface) matches the target shape (wire-frame).

We additionally provide a video showing the intermediate optimization step for all the results in the paper as part of our additional material.

Color Legend. We use a colorscale from blue to red to indicate von Mises stress, green arrows to indicate Neumann boundary conditions, and black square to indicate nodes that have a Dirichlet boundary condition. To reduce clutter, we use a uniform gray to indicate objects with a uniform Dirichlet boundary condition on all nodes.

To avoid singularities in the optimization we add, to the objective function, a boundary smoothing term (3.43) in all our shape optimization experiments, and a material regularization term (3.44) to all our material optimization experiments.

3.10.1 SHAPE OPTIMIZATION WITHOUT CONTACT

We start our analysis with shape optimization problems without contact or friction forces.

Static: Bridge. We fabricate a 2D solution to verify the correctness of our formulation and implementation. Starting from the shape of a bridge (Figure 3.3) we run a forward linear elasticity simulation with the two sides fixed and gravity forces. We now perturb the geometry of the rest pose and solve a shape optimization problem to recover the original rest pose, i.e. we remove the perturbation we introduced by minimizing the objective in (3.37).

We use the same model for a more challenging problem (Figure 3.4): we use the same Dirichlet

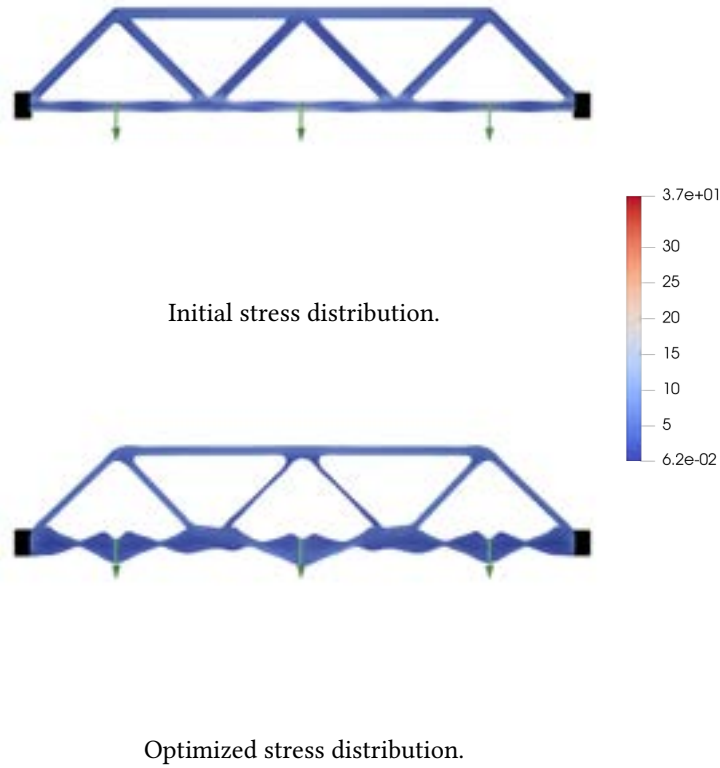


Figure 3.4: Result of shape optimization to minimize the average stress.

conditions and material model, replace the gravity forces by 3 Neumann conditions on the lower beams, and minimize the average stress (3.36). To avoid trivial solutions we add a constant volume constraint (Section 3.8.2).

Static: 3D Beam. Moving to 3D (Figure 3.5), we perform static optimization of maximum the stress using Neo-Hookean materials on a beam standing on a fixed support at the center (all black nodes have zero Dirichlet boundary conditions), and with two side loads applied as Neumann boundary conditions. The maximum stress is reduced from 2,788 to 1,531.

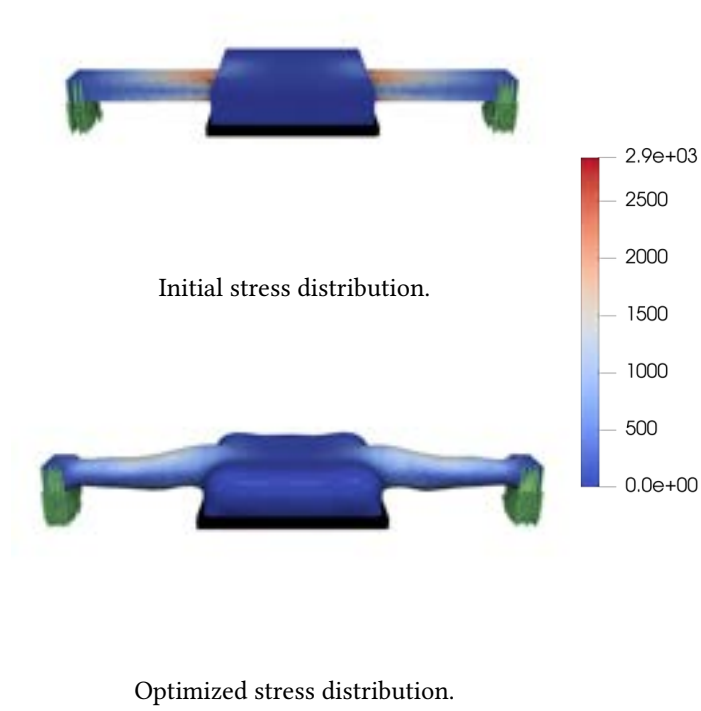


Figure 3.5: Result of stress minimization on a beam standing on a platform, with two loads on its sides.

3.10.2 SHAPE OPTIMIZATION CONTACT

Our framework supports contact and transient friction forces between objects without requiring explicit definition of contact pairs.

Static: Interlocking. We borrow the experimental setup used in [Tozoni et al. 2021]: we optimize

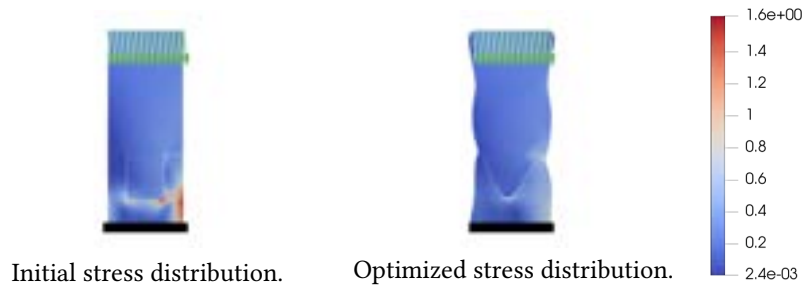


Figure 3.6: Result of shape optimization to minimize the maximum stress.

the shape of two interlocking 2D parts (Figure 3.6) to minimize maximum the stress (3.36). For this example we use a non-linear material. The bottom part is fixed and a force pointing downright is applied to the top. Figure 3.6 shows how the shape changes to reduce the maximum stress from 1.624 to 1.003.

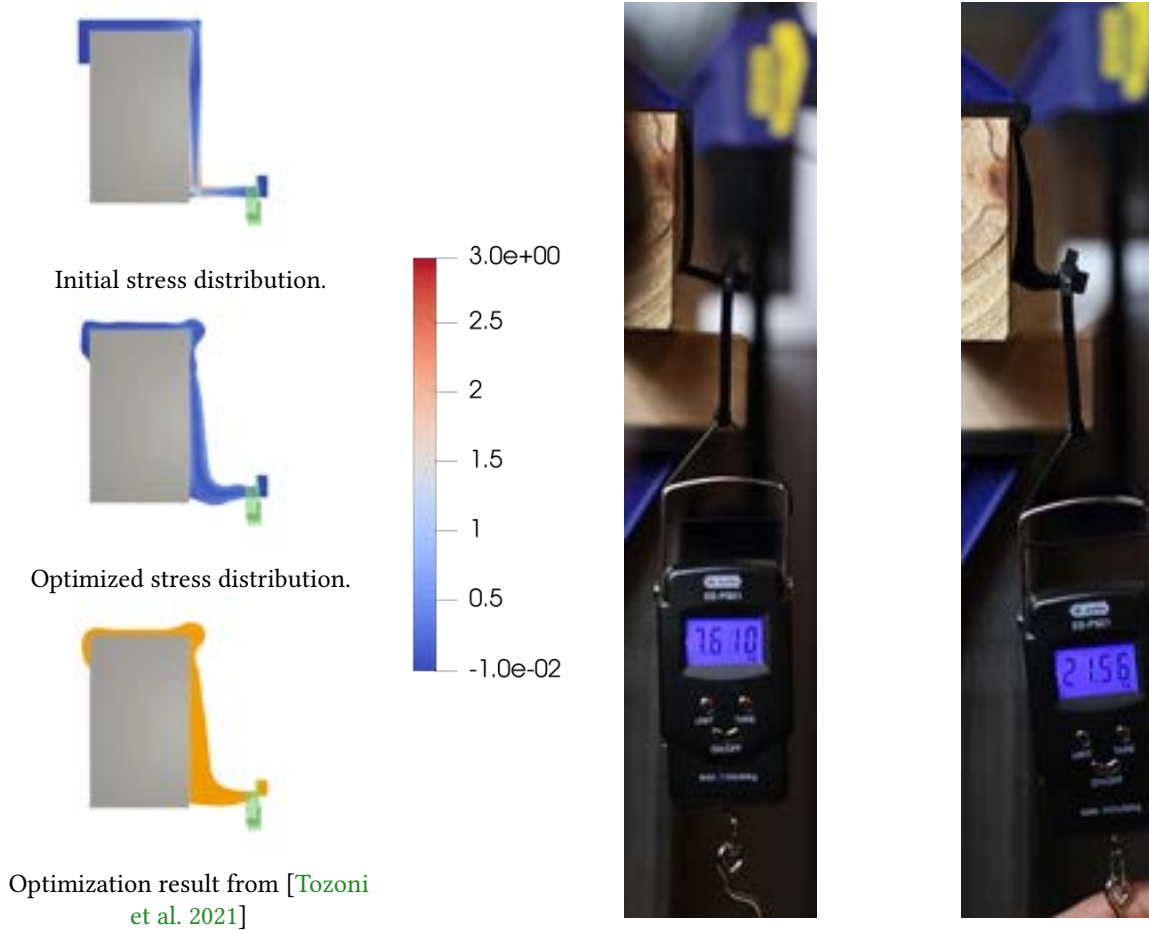


Figure 3.7: Shape optimization of a hook to reduce stress concentration (left). Fabricated results with maximum load before failure (right).

Static: 2D Hook. To physically validate our shape optimization results we reproduce the experiment in [Tozoni et al. 2021, Figure 21], where a hook is optimized to minimize the maximum stress (3.36) when a load is applied to one of its ends (Figure 3.7). The grey block is fixed with zero Dirichlet conditions on all nodes. We physically validate that the optimized shape is able to withstand a load of over $3\times$ the unoptimized shape before breaking (Figure 3.7). The hook has

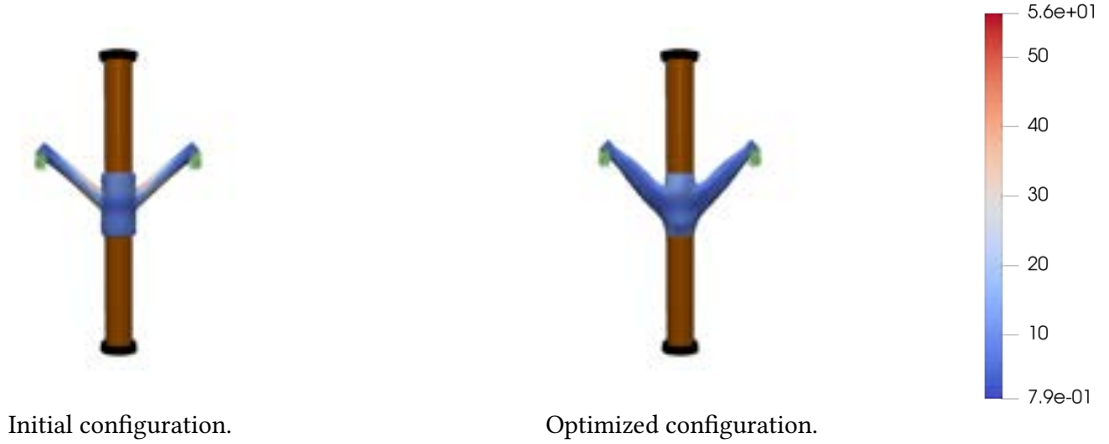


Figure 3.8: Result of shape optimization of a hanger to reduce stress concentration.

been fabricated using an Ultimaker 3 3D printer, using black PLA plastic. Despite the different contact model, the result is quite similar to the one presented in [Tozoni et al. 2021]: our approach has the advantage of not requiring manual specification of the contact surfaces.

Static: 3D Hanger. We also reproduce the experiment [Tozoni et al. 2021, Figure 29]: a coat hanger is composed of two cylinders and a hanger keeping the together. The shape of the hanger is optimized to minimize the maximum internal stress (3.36) when two loads are applied on its arms (Figure 3.8). The maximal stress is reduced from 56.451 to 17.888.

Transient: Bouncing ball. As a demonstration of shape optimization in a transient setting, we run a forward non-linear simulation of a ball bouncing on a plane and use its trajectory as the optimization goal (3.38). We then deform the initial shape into an ellipse and try to recover the original shape (Figure 3.9).

Transient: Sliding Ball. We optimize the shape of a ball sliding down a ramp to minimize the internal stress (3.36). Perhaps unsurprisingly, the ball gets flattened on the side it contacts with the ramp as this leads to a major reduction of max stress, from 16, 274 to 5, 432.

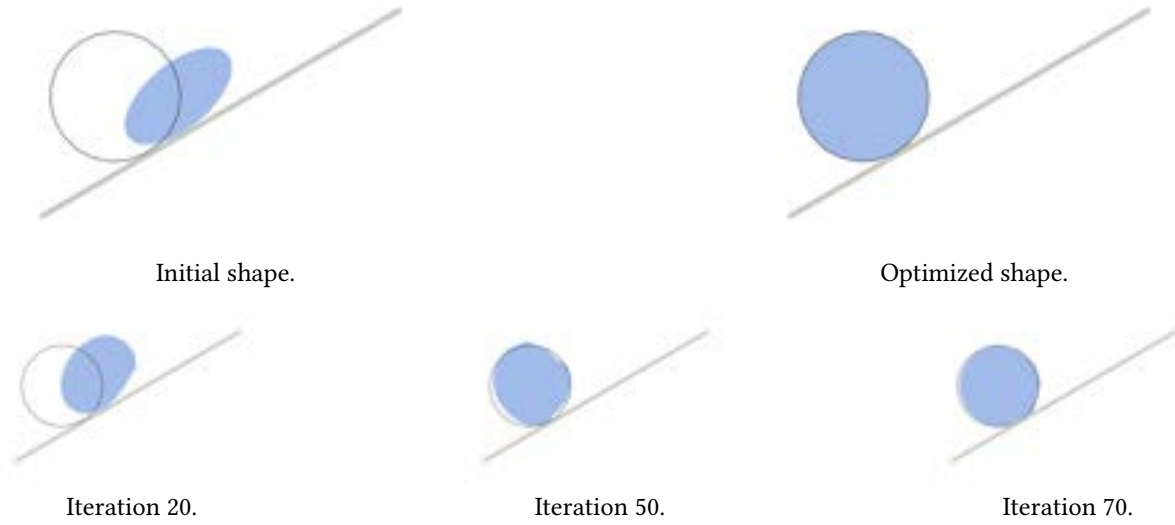


Figure 3.9: The result of the shape optimization (blue surface) matches the desired trajectory (wire-frame).

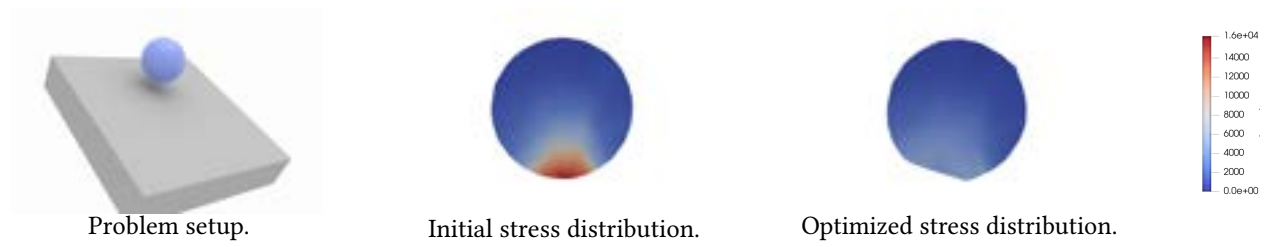


Figure 3.10: Result of shape optimization to reduce stress.

3.10.3 INITIAL CONDITIONS

Our formulation supports the optimization of objectives depending on the initial conditions. We show two examples, the first involving an object sliding on a ramp with a complex geometry, and the second simulating a game of pool, using bunnies instead of spheres.

Transient: Puzzle Piece. We synthesise a trajectory using a forward simulation, and we then perturb the initial conditions and try to reconstruct them minimizing (3.37), with an additional integration over time (Figure 3.11). The puzzle piece uses a Neo-Hookean material.



Initial trajectory.



Iteration 3.



Iteration 5



Optimized result

Figure 3.11: Optimizing the initial velocity of a bouncing puzzle. Target is shown as a black outline while the trajectory being optimized is blue.

Transient: Bunny Pool. We use our solver to optimize the throw (initial position and velocity) of a bunny on a pool table to hit and displace a second bunny into the corner (Figure 3.12), minimizing (3.38). This example involves complex contact between the bunnies and the pool table, and also friction forces slowing down the sliding after contact.

3.10.4 MATERIAL OPTIMIZATION

Finally, we look at material optimization problems, where our differentiable simulator is used to estimate the material properties of an object from observations on its displacement.



$t = 0$



$t = 0.6$



$t = 1.4$



$t = 2.15$

Figure 3.12: The initial conditions of the green bunny (left) are optimized to push the blue bunny in the corner of the pool table (right).

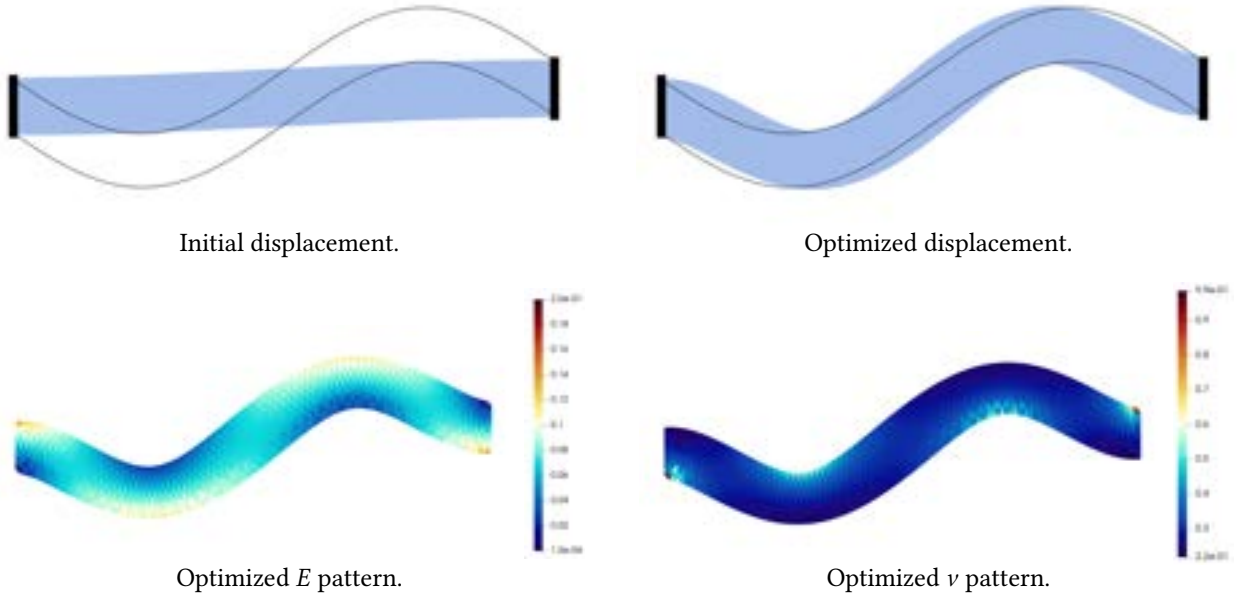


Figure 3.13: Optimized material parameters to obtain a displacement (blue surface) in y -direction similar to a sine function for a linear material model (wire-frame).

Static: Sine. We optimize the material of a bar to match the shape of a sine function (wire-frame) when Dirichlet boundary conditions are applied at its ends (3.37). The rest shape of this bar is a rectangle $[-4, 4] \times [-0.3, 0.3]$, the left and right surfaces are fixed by Dirichlet boundary condition of $u_y = 0.7 \sin(x + u_x)$ and $u_x = -\text{sign}(x)$, and no body force is applied. Figure 3.13 shows that deformed bar is aligned with a sine function.

Static: Bridge. We assign material parameters $\lambda = 160, \mu = 80$ to a bridge shape and run a linear forward simulation to obtain the target displacement u^* (Figure 3.14 in gray), using the same set of boundary conditions as Figure 3.3. We initialize the optimization using uniform material $\lambda = 100, \mu = 50$ and minimize (3.37).

Static: Cube. We set up a physical experiment with a silicon rubber cube compressed by a vise. The deformation is acquired using an HP 3D scanner, and a set of marker points is manually extracted from the scan. We minimize (3.37) to find the material parameters which produce the observed displacements. We found that the material parameter that lead to smaller error is

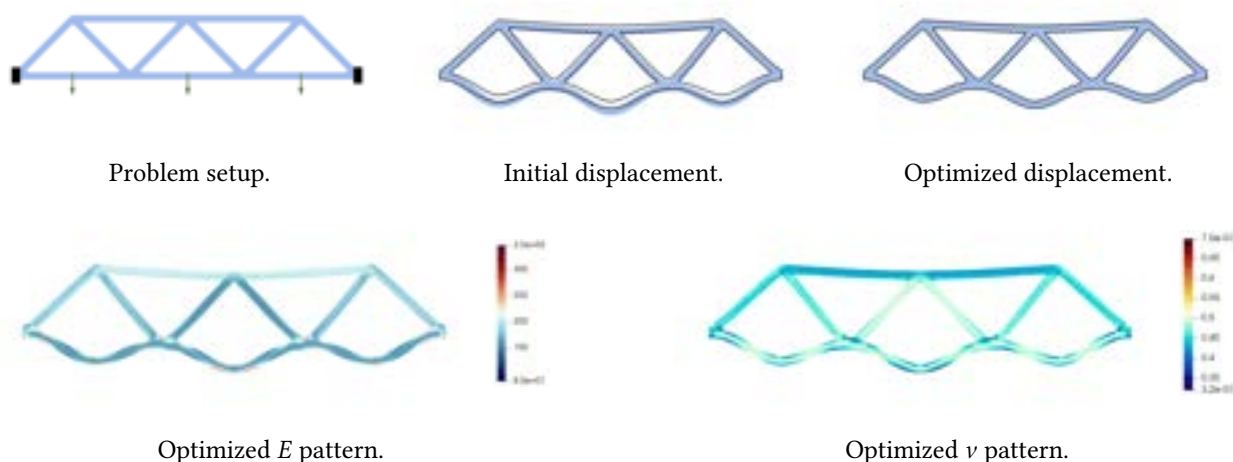


Figure 3.14: Optimization of the materials of a bridge (blue surface) to match a forward simulation (wire-frame).

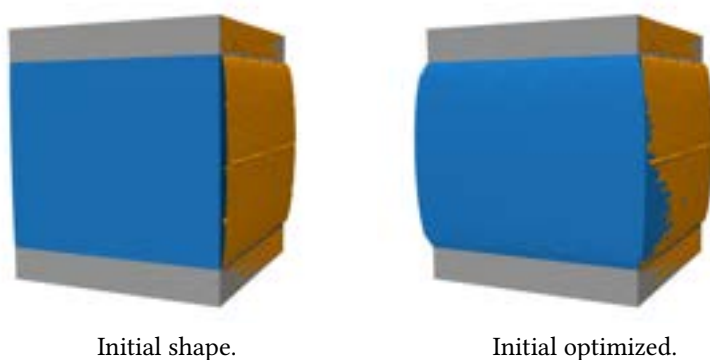


Figure 3.15: Material optimization (blue) to match real data (orange).



$\nu = 0.4817$ (the Young's modulus does not affect its deformation in this setting) and the L2 error in markers position is $1.48353 \times 10^{-5} m^2$.

Static: Micro-Structure. We repeat the same experiments with the complex geometry of a micro-structure tile from [Panetta et al. 2017c]. This is a challenging example, as the micro-structure beams come in contact after compression. Our solver can find material properties $E = 2.27 \times 10^5 \text{Pa}$, $\nu = 0.348$ with a L2 error on the markers of $7.9 \times 10^{-5} m^2$.

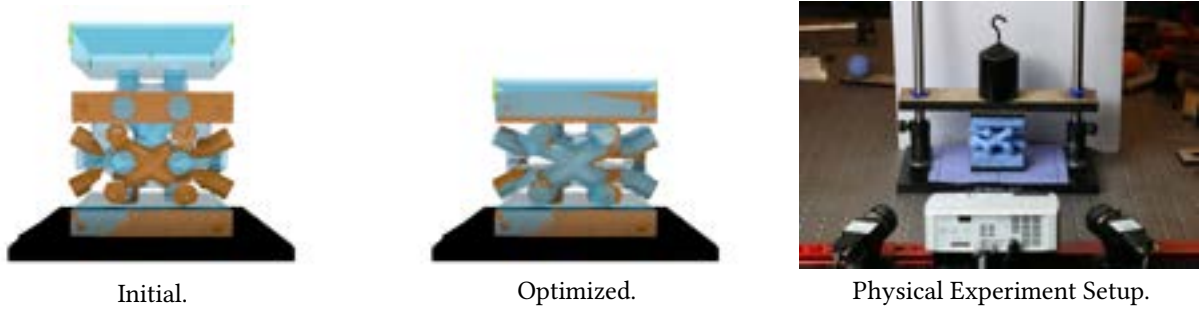


Figure 3.16: We acquire the deformation of a silicon micro-structure using a 3D scanner, and use the observed displacement, created by a controlled load, to reconstruct its material parameters. The simulated mesh (transparent blue) matches well the 3d scan data (orange). The optimization is initialized with $E = 1e6\text{Pa}$, $\nu = 0.3$ and converges to $E = 2.27 \times 10^5\text{Pa}$, $\nu = 0.348$.



Figure 3.17: Non-linear transient simulation of a kangaroo (blue surface) bouncing on a plane to match a target shape (wire-frame).

Transient: Kangaroo. As an example of reconstruction of material parameters from a transient simulation, we run a forward simulation to obtain a transient non-linear target displacement. Then we minimize (3.37) to reconstruct the material parameters (Figure 3.17). The initial material parameters are $E = 3 \times 10^6\text{Pa}$, $\nu = 0.5$, and the target material parameters are $E = 10^7\text{Pa}$, $\nu = 0.3$.

Transient: Sliding Puck. A similar experiment is repeated for the friction coefficient using a more challenging object geometry (Figure 3.18) of a puck sliding over a curved ramp. The initial friction coefficient is $\mu = 0.5$, and the target friction coefficient is $\mu = 0.2$.



Figure 3.18: Optimization of the friction (blue surface) to match the target shape (wire-frame).

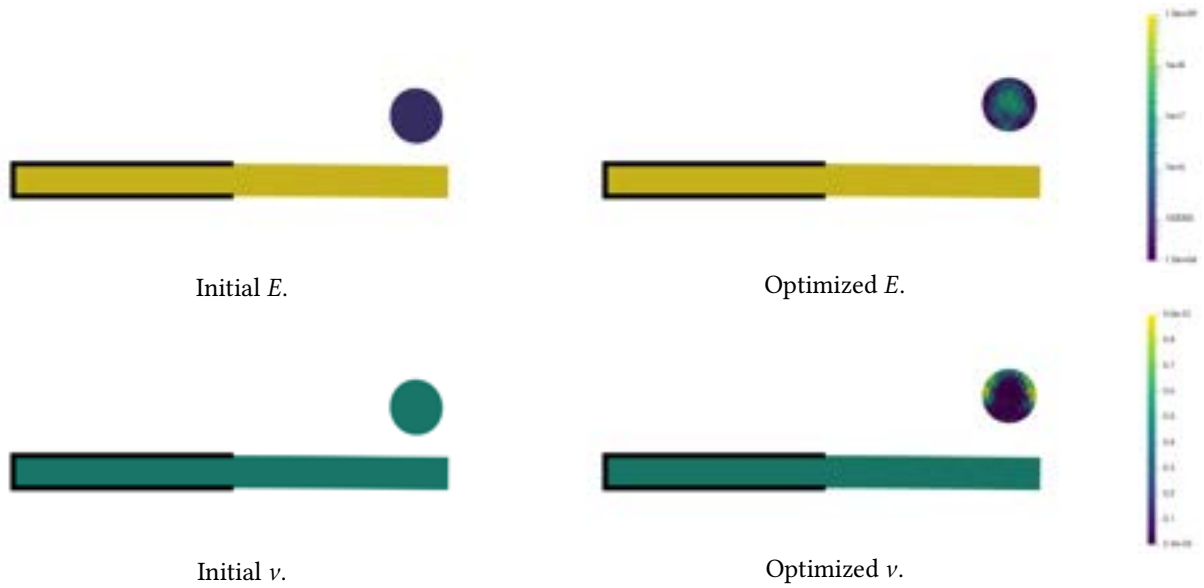


Figure 3.19: Material optimization to increase the bouncing height.

Transient: Bouncing Ball. We show that the height of the bounce of a ball can be optimized by changing the material parameters (Figure 3.19). The optimization was done in λ and μ , although we show the results using E and ν to give intuition. Initial material parameters for the ball and plank were $E = 10^5 \text{Pa}$, $\nu = 0.48$ and $E = 10^9 \text{Pa}$, $\nu = 0.48$, respectively and the elasticity model used was NeoHookean. Note that we added a smoothing term to the optimization to increase smoothness in the material parameters.

Transient: Physical Experiment Bouncing Ball. Finally, we show that we can optimize for the material parameters, friction coefficient, and damping parameters of a silicone rubber ball

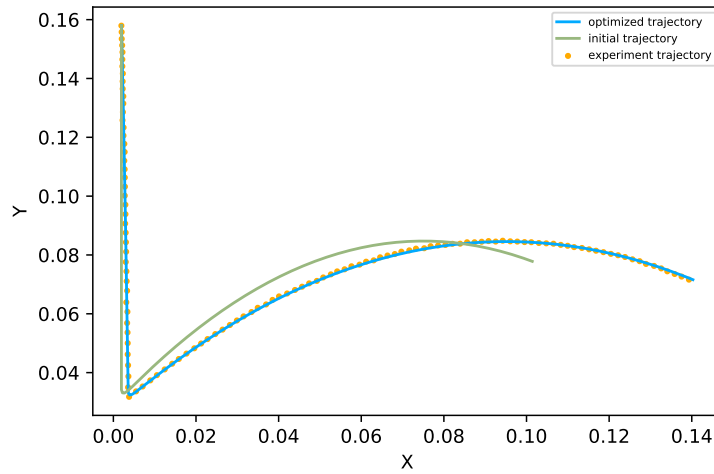
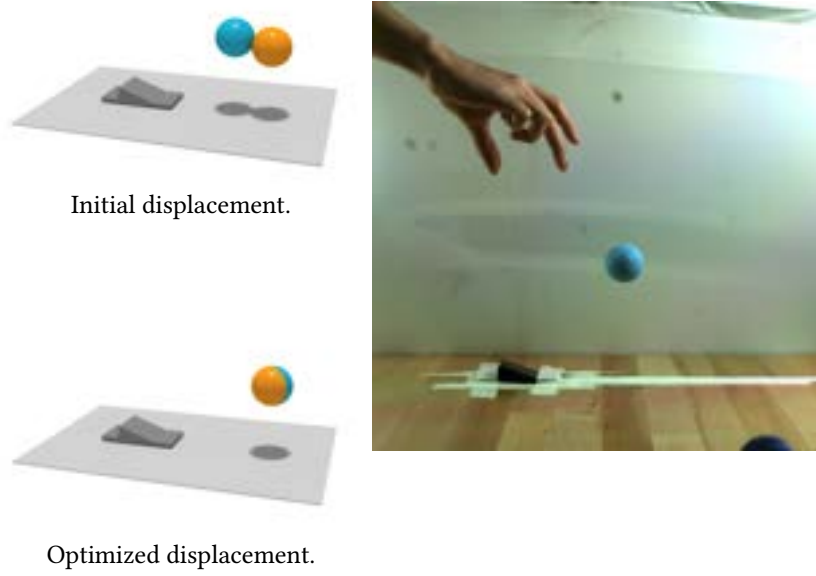


Figure 3.20: Material optimization (blue) to match experimental data (orange).

bouncing on an incline, using trajectory data from a physical experiment. The real world dynamics of the ball are captured using a high speed camera and used to formulate a functional based on (3.38), which penalizes differences between the observed and simulated barycenter of the ball. The material model used is NeoHookean and we match initial conditions by optimizing for them using the observed barycenters of the ball before it hits the ground.

3.11 CONCLUDING REMARKS

We introduced a generic, robust, and accurate framework for PDE-constrained optimization problems involving elastic deformations of multiple objects with contact and friction forces. Our framework supports customizable objective functions and allows for optimization of functionals involving the geometry of the objects involved, material parameters, contact/friction parameters, and boundary/initial conditions. The formulation applies to both 2D and 3D problems.

We believe the benefits of our analytic derivation of the adjoint system (efficiency, generality, guarantee of converge under refinement) outweighs its downsides (complexity of derivation, difficulty in implementation, and requirement of an explicit FE mesh). We plan to extend our approach to a wider set of PDE-constrained problem and to further optimize it for common use cases in material design and robotics. In particular we would like to explore the following directions:

1. Add support for periodic boundary conditions, which are required for the design of microstructure families [Tozoni et al. 2020].
2. Add support for rigid and articulated objects (i.e. allow the material stiffness to be infinite). We plan to incorporate the IPC formulation introduced in [Ferguson et al. 2021] to improve performance in design problems involving rigid objects.
3. Many robotics problems involve manipulation of plastic objects or interaction with fluids: adding support for additional physical models will widen the applicability of our simulator.
4. We designed our system to provide accurate modeling of elastic, contact, and friction forces, as the majority of PDE-constrained applications requires accurate simulations faithfully reproducing the behaviour observable in the real works. However, there are applications

where this is not necessary, and in these cases it would be possible to either use simpler elastic models, or reduce the accuracy of the collision/friction forces by using proxy geometry. This is commonly done in graphics settings, and it would be interesting to add this option to our system to accelerate its performance.

4 | A LOW-PARAMETRIC RHOMBIC MICROSTRUCTURE FAMILY FOR IRREGULAR LATTICES

This chapter, which is based on a paper published in SIGGRAPH 2020 [[Tozoni et al. 2020](#)], investigates a two-scale topology optimization approach where a curved shape can first be split into close to rhombic shapes and then filled with microstructures (one in each cell) according to a smooth material to geometry map.

This work was done in collaboration with Jérémie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, and Denis Zorin. My role in this paper was designing the algorithm for, and computing the smooth map of microstructures from material to geometric parameters. In addition, I also participated in designing and running experiments for specific shapes, analyzing results and the accuracy of our method, fabricating our examples using laser cutting, and running physical experiments.

New fabrication technologies have significantly decreased the cost of fabrication of shapes with highly complex geometric structure. One important application of complex fine-scale geometric structures is to create variable effective elastic material properties in shapes manufactured from

a single material. Modification of material properties has a variety of uses, from aerospace applications to soft robotics and prosthetic devices. Due to its scalability and effectiveness, an increasingly common approach to creating spatially varying materials is to partition a shape into cells and use a parametric family of small-scale geometric structures with known effective properties to fill the cells.

We propose a new approach to solving this problem for extruded, planar microstructures. Differently from existing methods for two-scale optimization based on regular grids with square periodic cells, which cannot conform to an arbitrary boundary, we introduce cell decompositions consisting of (nearly) rhombic cells. These meshes have far greater flexibility than those with square cells in terms of approximating arbitrary shapes, and, at the same time, have a number of properties simplifying small-scale structure construction. Our main contributions include a new family of 2D cell geometry structures, explicitly parameterized by their effective Young’s moduli E , Poisson’s ratios ν , and rhombic angle α with the geometry parameters expressed directly as smooth spline functions of E , ν , and α . This family leads to smooth transitions between the tiles and can handle a broad range of rhombic cell shapes. We introduce a complete material design pipeline based on this microstructure family, composed of an algorithm to generate rhombic tessellation from quadrilateral meshes and an algorithm to synthesize the microstructure geometry. We fabricated a number of models and experimentally demonstrated how our method, in combination with material optimization, can be used to achieve the desired deformation behavior.

4.1 INTRODUCTION

Advances in fabrication of highly complex geometry using additive manufacturing and other technologies resulted in new opportunities for shape design. In particular, *small-scale*, topologically and geometrically complex structures make it possible to achieve variable effective material properties using a single material for fabrication, including material properties not easily obtained

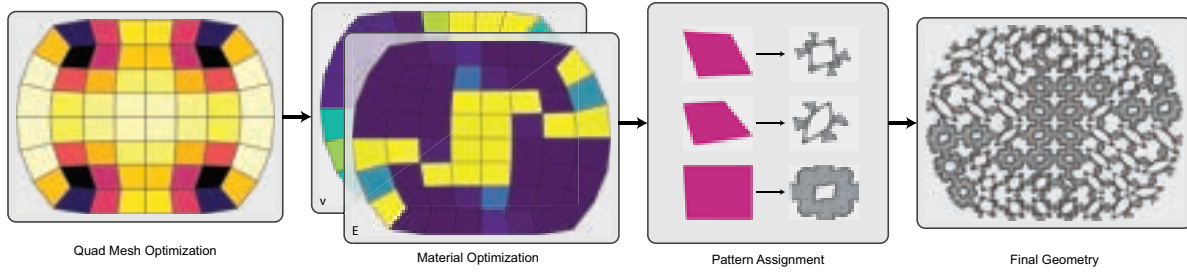


Figure 4.1: The cells of a quadrilateral mesh are optimized to become quasi-rhombic; then material properties (variable Young’s moduli and Poisson’s ratio) are assigned to the cells. The assigned material properties are used to evaluate the geometric parameters of a tileable microstructure, encoded with a smooth spline map.

by other means (such as negative Poisson’s ratio) or material properties needed for precise control of deformation behavior of a shape. Realizing the potential of microstructures requires automatic generation and high-level control of the geometry, absent from commonly used geometric modeling tools.

A variety of approaches were developed for generating small-scale structures. Global methods, like topology optimization, are flexible but require very expensive computations to obtain high-quality results for fine-scale structures (cf. [Aage et al. 2017]). Other approaches partition the problem into different scales, using a variety of methods to generate a small-scale structure locally from a coarse-scale assignment of spatially-varying target properties. In this paper we present a method for design of small-scale structure families supporting an approach of the second type; specifically,

- Partition an input shape into quadrilateral cells, possibly with irregular connectivity; each cell is assigned target elastic properties;
- Assign to each cell a geometric microstructure, chosen from a family of such structures directly parameterized by their effective elastic properties and cell shapes.

The main goals for the choice of the microstructure family include: (1) cover a broad range of material properties, (2) be simple enough for fabrication at a small scale (i.e. avoiding thin

features and small holes), (3) handle a range of cell shapes, (4) be easily tileable without significant modifications, (5) depend smoothly on the target effective elasticity tensor to avoid discontinuities in transitioning between varying material properties, and (6) be efficient to compute to enable tiling of large lattices.

The question of designing such microstructures is extensively studied in the literature, although advances in design of practical families covering large ranges of material properties are more recent. Existing techniques fall somewhat short of meeting the requirements above. First, these methods are exclusively based on periodic cell tilings with squares, regular triangles, or hexagons: only a single cell shape is used. This restricts the shapes that can be tiled to those constructed of this single cell type. Other shapes require either cutting cells, or deforming cells; both changes to the cell shape lead to a significant deviation from the intended deformation behavior (Section 4.8).

Most closely related previous work constructs large libraries of cell geometries [Panetta et al. 2015a; Schumacher et al. 2015], with a separate shape or topology optimization performed for a dense set of values of material properties (Young’s modulus and Poisson’s ratio combinations for isotropic materials), and a specific base material. While this is generally adequate, if variable cell shapes are used, the size of the needed library will become prohibitively large (and still be limited to a discrete set of shapes). Furthermore, a different set of microstructures needs to be computed if some aspects of the properties of the base material change. These approaches do not provide the guarantee of smooth dependence of the resulting geometry on the material properties, and, as a consequence, a large number of samples are needed and interpolation between samples cannot be used.

In this paper, we address these problems for extruded, planar microstructures, aiming to meet the requirements listed above. The main aspects of our approach include the following.

- We choose to use *rhombic* cells, among different classes of tileable cells, since they have the

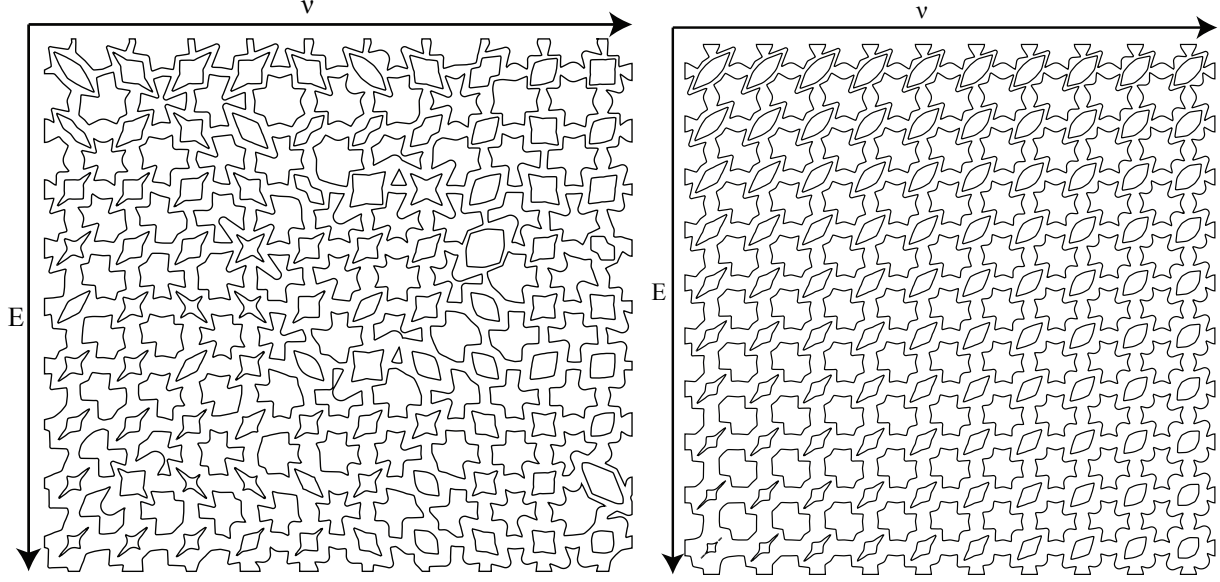


Figure 4.2: An array of microstructures with Young’s modulus E varying from 0.005 to 0.1 and ν varying from -0.15 to 0.5. Top: obtained by pointwise inverse homogenization of [Panetta et al. 2017a]; Bottom: our family.

flexibility to approximate arbitrary boundaries and can be connected in arbitrary orientations while being sufficiently simple to define microstructures for all members of the family in a compact way.

- We develop a parametric family of structures (Figure 4.2) completely described by: (1) Eight geometric parameters, defined as smooth spline functions $\mathcal{P}(E, \nu; \alpha)$ of material properties (Young’s modulus E and Poisson’s ratio ν in the isotropic case) and the cell shape parameter (rhombus angle α). (2) The domain of this parameterization in (E, ν) space is defined by three simple linear constraints $\ell(\alpha)$ dependent on the angle. This enables efficient optimization of material properties.
- Our family is tileable, i.e., the structures for adjacent cells connect with little or no modifications. Our tiles have matching topology and positions in the boundary. In addition, due to the smoothness of our mapping functions, if the transition of materials is smooth, the change in the connection region will also be smooth and almost no difference exists in the

shape between neighboring cells.

- We demonstrate that the set of geometries is *universal*, i.e., can be used for any base material (although the ranges of realizable material properties do change).
- We present an algorithm for optimizing a quadrangulation of a planar domain to minimize the deviation of quads from the rhombic shape and demonstrate that this approach yields nearly-rhombic cells so that our microstructure family can be used.

We will make the data for computing microstructure geometric parameters, and constructing microstructure geometry from these parameters, publicly available.

4.2 RELATED WORK

We build on the foundation of previous work on construction of microstructure families and mesh quality optimization.

Periodic homogenization. Homogenization is a central tool in our construction. We use an extension of the FEM-based formulation used in [Panetta et al. 2015a, 2017a], which, in turn, goes back to [Allaire 2002a], and is widely used in the literature. [Schumacher et al. 2018a] presents an efficient homogenization approach tailored to rod structures. In these papers, regular lattices are used. We instead perform homogenization on a range of rhombic cell shapes, by transforming the problem to a domain where the cell is square, and replacing the target tensor with a transformed one. Transformations of the homogenized material properties when the base material is changed are considered in [Cherkaev et al. 1992].

Microstructure design and optimization. Many papers considered different aspects of microstructure design, see e.g., books [Cioranescu and Donato 1999; Milton 2002b; Torquato 2002a; Allaire 2002a; Cherkaev 2000], and references in [Schumacher et al. 2015] and [Panetta et al. 2015a],

which represent topology and shape optimization approaches to microstructure optimization respectively. We refer to these methods as *pointwise inverse homogenization*, as they construct microstructures separately for each combination of effective material parameters.

Other examples of topology-optimization based work can be found in [Bendsøe 1989; Bendsøe and Sigmund 2003; Nakasone and Silva 2010; Chen et al. 2018]. Initially, mostly the problem of identifying extremal microstructures are considered (i.e., microstructures with properties at the boundary of the ranges that can be achieved). Recent works, e.g., [Schumacher et al. 2015; Panetta et al. 2015a; Zhu et al. 2017; Ostanin et al. 2018] consider the question of constructing families spanning a broad range of elastic properties. In particular, [Ostanin et al. 2018] shows that near-optimal ranges of isotropic material behaviors can be achieved in 2D with hexagonal and triangular cells and a small number of microstructure parameters. We discuss the differences from this work in Section 4.4. [Milton et al. 2017] provides a characterization of achievable elastic tensors in terms of energies in 2D and 3D.

Most of the works designing microstructure families use an ad hoc approach to connecting structures corresponding to adjacent cells. [Garner et al. 2019] takes a more systematic approach for structures obtained using topology optimization by adding additional terms to the functional. A recent concurrent work [Martínez et al. 2019] introduces a metric that can be used to interpolate between a variety of microstructures on 2D regular grids, allowing to create smooth variation of material properties similar to our construction (but limited to regular lattices). [Konaković-Luković et al. 2018] uses a special type of 2D triangular auxetic structure to effect conformal surface deformations. This method requires domain meshing with triangles close to regular. Similarly, a recent paper [Malomo et al. 2018] uses 2D spiral microstructures for controlling deformation of sheets into a target shape. Varying geometric properties of spirals allows for a restricted control over material properties.

As an alternative to periodic microstructures, [Martínez et al. 2016; Martínez et al. 2017] construct randomized printable structures with control over Young’s moduli both for isotropic and anisotropic target properties, but cannot independently control the Poisson’s ratio. [Ion et al. 2016, 2019] describe a simple set of small-scale two-dimensional structures that can be assembled into mechanisms, and a computational design tool creating this type of mechanisms. While a range of properties can be obtained by changing the basic structure parameters their properties are difficult to control precisely.

A number of works apply the two-scale approach in a different way in 2D, using a simple rectangular cell structure and obtaining a directional field and scalar fields for geometric parameters defined on a regular grid. A field-aligned coarse mesh is extracted from the directional field and then filled with microstructures with parameters determined by the scalar fields [Groen et al. 2019; Groen and Sigmund 2017; Gil-Ureta et al. 2019]. Compared to these works, we do not require the mesh cell orientations to be physically meaningful: in our case the effective material distribution and meshing are independent.

Global topology optimization. In [Wu et al. 2016; Aage et al. 2017; Liu et al. 2018], topology optimization was scaled up to high-resolution uniform and adaptive 3D grids. [Wu et al. 2018] performs high-resolution topology optimization with additional constraints to create an evenly distributed porous small-scale structure minimizing compliance for specific loading scenarios. The two-scale methods based on microstructures can always be combined with topology optimization to improve efficiency or resolution achievable in a given time.

Fabrication. [Lin et al. 2004b,a; Hollister 2005; Kang 2010] have demonstrated fabrication of optimized microstructures in the context of bone scaffold and fusion cage design. [Greaves et al. 2011; Schwerdtfeger et al. 2011; Bückmann et al. 2012; Andreassen et al. 2014] have shown the possibility of manufacturing auxetic materials. The idea of manufacturing objects with spatially

varying properties using tileable structures also appears in [Hiller and Lipson 2009]. [Bickel et al. 2010] designs and fabricates objects satisfying an input deformation by optimizing for the best combination of stacked layers of their multi-material 3D printer’s base materials. [Skouras et al. 2013b] applies discrete material optimization to achieve desired deformations of complex characters with actuation, fabricating the results with multi-material printing.

Fabrication constraints. [Qian 2016; Gaynor and Guest 2016; Langelaar 2016] have made recent progress in incorporating undercut/overhang angle constraints in the topology optimization framework. However, these method enforce the constraints only approximately, requiring parameter tuning, and add nonlinearities to the problem that hinder convergence [Gaynor and Guest 2016]. [Panetta et al. 2015a] enforces approximate printability via constraints on the skeleton defining the structure. In our planar case, there is no need to impose fabrication constraints other than requiring that the geometry should stay connected, and the minimal feature size is large enough for the chosen fabrication process.

Mesh optimization. The optimization of the element shape of discrete meshes has been studied in many disciplines. In the context of finite element simulations, the shape is optimized to reduce the distortion introduced by the geometric map [Livesu et al. 2015; Brewer et al. 2003]. Similarly, for texture mapping and quadrilateral meshing applications the distortion of a map from a surface to a plane is minimized by either evolving a Tutte’s parameterization [Hormann and Greiner 2000; Sander et al. 2001; Sorkine et al. 2002; Degener et al. 2003; Schüller et al. 2013; Aigerman et al. 2014; Poranne and Lipman 2014; Smith and Schaefer 2015; Fu et al. 2015; Kovalsky et al. 2016; Shtengel et al. 2017; Zhu et al. 2017] or recovering from a possibly inverted initial guess [Lipman 2012; Aigerman and Lipman 2013; Kovalsky et al. 2015; Fu and Liu 2016]. These maps are also commonly used for mesh deformation applications [Sorkine and Alexa 2007; Bouaziz et al. 2012], and special constraints are often used in architectural geometry to generate meshes with planar faces [Bouaziz et al. 2014; Deuss et al. 2015; Poranne et al. 2013; Jiang et al. 2015; Tang et al. 2014].

We introduce an algorithm to optimize a quadrilateral mesh to have rhombic elements, a requirement that, to the best of our knowledge, has never been studied before. Our algorithm follows the paradigm introduced in ShapeUp [Bouaziz et al. 2012], alternating the projection to the space of rhombic shapes and a continuous optimization, and it relies on the solver introduced in [Rabinovich et al. 2017b] to speed up convergence.

4.3 PROBLEM FORMULATION AND SOLUTION OVERVIEW

Problem. Our goal is to approximate a distribution of (potentially variable) material properties given by a spatially variable symmetric elasticity 4-tensor $\bar{C}(\mathbf{x})$ on a two-dimensional polygonal domain Ω , by partitioning it into constant-material cells, and assigning a microstructure to each cell. We assume that a single *base material* with known elastic properties given by the tensor C^{base} is used for all small-scale geometry.

The geometry of each cell Q is chosen so that its *homogenized* or *effective* elasticity tensor matches that of the assigned material $\bar{C}(Q)$.

Informally, the effective elasticity tensor can be understood as follows. If the cell Q were repeated periodically, a block of material consisting of sufficiently large number of such cells would behave as if it were made of a homogeneous material with elasticity tensor $\bar{C}(Q)$. In this work, we focus on isotropic target material distributions, for which the elasticity tensor is defined by a pair of parameters $M = (E, \nu)$, Young’s modulus and Poisson’s ratio.

We focus on isotropic materials since the effective orthotropic elasticity tensors that rhombic cells can produce have principal axes aligned with the diagonals of the rhombi; while for some purposes this broader space can still be used, the space of possible materials becomes mesh-dependent, complicating and restricting material optimization. In contrast, considering isotropic materials makes the result much more mesh-independent.

As mentioned in Section 4.1, our goal is to design the geometry in individual cells to have the following properties: (1) cover a broad range of material properties, (2) be simple enough for fabrication at small scale (i.e. avoiding thin features and small holes), (3) handle a range of cell shapes, (4) be easily tilable without significant modifications, (5) depend smoothly on the target effective elasticity tensor to avoid discontinuities in transitioning between varying material properties, and (6) be efficient to compute to enable tiling of large lattices.

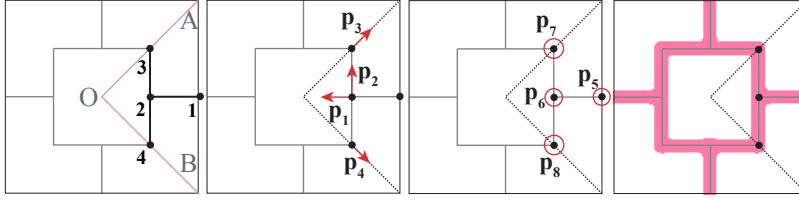


Figure 4.3: Parameters defining our elementary cell geometry: 4 displacement parameters (two nodes 3,4 have only one free displacement and the node 1 is fixed, due to symmetry), and 4 radii, one per node. The displacements are specified in barycentric coordinates with respect to the triangle OAB.

Notation. We use *rhombic cells* R and a specific parametric cell geometry structure, which respects the two reflection symmetries of the rhombus. The shape of the rhombic cell is described by a single parameter, $\alpha \leq \pi/2$, the smaller angle of the rhombus.

The geometry of each cell is defined by a vector of parameters $\mathbf{p} \in \mathbb{R}^n$, $n = 8$, (Figure 4.3); the geometry is generated from parameter values using a 2D version of the method of [Panetta et al. 2017a], described in Appendix 4.10.1. The reasons for this choice are summarized in Section 4.4. We assume that for each parameter a range $[p_i^{min}, p_i^{max}]$ is provided. The n -dimensional box of admissible parameter values is denoted B_p . In particular, for *any* choice of parameters, the resulting homogenized elasticity tensor is *orthotropic*, and completely determined by four components, e.g., Young’s moduli E_1, E_2 along two directions, shear modulus G , and one of two Poisson’s ratios ν_{12} .

The function $H(\mathbf{p}; \alpha) : B_p \rightarrow \mathbb{R}^4$, the *homogenization function*, computes the effective elasticity tensor components of a cell with angle α from its geometry parameters. We describe a way to compute this function on arbitrary parallelogram cells, including rhombic, in Section 4.5. We use

$H_E(\mathbf{p})$ and $H_\nu(\mathbf{p})$ to denote the Young's modulus and Poisson's ratio corresponding to $H(\mathbf{p})$ when it is isotropic.

Solution overview. Our approach is composed of two main components:

- We construct a parametric family of cell structures solving the *inverse homogenization problem* for a range $D(\alpha) \subset H(B_p; \alpha)$ of material properties $(E, \nu) \in D(\alpha)$, where E is the Young's modulus and ν is the Poisson's ratio, for a range of cell angles α , $[\alpha_{min}, \pi/2]$. This family is described by the *material-to-geometry* map $\mathcal{P}(E, \nu; \alpha) : D(\alpha) \rightarrow B^p$.
- We optimize a quadrangulation to obtain quasi-rhombic cells, so that the map $\mathcal{P}(E, \nu, \alpha)$ can be used to fill in the small-scale geometry.

The difficulty of the problem of inverse homogenization is due to several factors. The homogenization map $H(\mathbf{p})$ is straightforward, but expensive, to compute, because it requires several finite-element solves as a part of its computation. More fundamentally, for a fixed α , we need to invert the map $H : B_p \subset \mathbb{R}^n \rightarrow \mathbb{R}^4$ on a two-dimensional subspace $\text{Iso} \subset \mathbb{R}^4$ of isotropic materials. This is challenging, since the inverse is far from unique. Last but not least, the map depends on the base material properties C^{base} , in addition to the homogenized parameters and cell geometry.

We introduce a novel, smooth, closed form material-to-geometry map $\mathcal{P}(E, \nu; \alpha, C^{\text{base}})$ covering a broad range of material properties and rhombus angles α , that can be used for an *arbitrary* isotropic base material with tensor C^{base} . This map is uniquely determined by selecting a suitable four-dimensional subspace of the geometric parameter space \mathbb{R}^n , and can be represented in a very compact form by a set of 3D spline.

In the next sections, we discuss our choice of cell shape (Section 4.4), review homogenization (Section 4.5) and show how it can be extended to rhombic cells. Then we explain how we solve the inverse homogenization problem (Section 4.6), and how to compute a tiling with quasi-rhombic

cells (Section 4.7).

4.4 CHOICE OF CELL SHAPE AND STRUCTURE

The choice of cell structure (i.e., topology of small-scale geometry, and parameters defining the geometry) is not unique: many choices may have similar behavior. We outline the heuristics we have used to select our structure for quadrilateral cells and briefly compare to alternatives.

Our choice of structure follows the general procedure outlined in [Panetta et al. 2015a], applied to planar square cells. On the one hand, we want to minimize the number of parameters and topological complexity of the structure, as complex cells are difficult or impossible to manufacture. On the other hand, we want to maximize the *coverage* that can be achieved by the structure, that is, the range of material properties (E, ν) . To obtain a rough prediction of coverage, we run a coarse sampling of geometric parameter space, evaluating the material parameters for each sample. In these sweeps, we enforce square symmetry (Figure 4.5), so that the resulting elastic tensor has only three free parameters (E, ν, G) , and we look at the coverage we get in the projection to the (E, ν) plane as an indicator of what one can expect for inverse homogenization.

Among simplest topology patterns, the specific one we have chosen has the largest area covered. Expanding this area requires increasingly complex topology which negatively affects manufacturability.

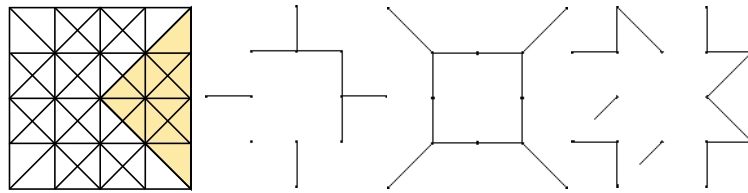


Figure 4.4: We consider structures consisting of edges of the graph depicted on the left, with degrees of freedom consisting of radii at nodes and node displacements. Right: several examples of structures with few edges/simple topology.

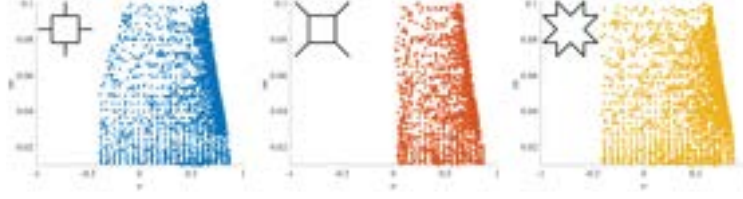


Figure 4.5: Coverage in (E, ν) space for three cell structures, with smaller and larger coverage.

Choice of cell shape. As we would like to partition arbitrary shapes into cells, using square cells is not possible: in general, we cannot conform to an arbitrary boundary without introducing cells of other shapes. At the same time, the cells need to be close to periodically tileable, i.e., if we use quadrilateral cells, close to parallelograms. Parallelograms have only central symmetry, which does not restrict the possible homogenized (averaged) elasticity tensors much; on the other hand, *rhombic* cells have two reflectional symmetries with respect to diagonals, and, as a result (Section 4.6), its elasticity tensor is orthotropic, as long as the small-scale geometry satisfies the same constraint. This considerably simplifies the construction of the material-to-geometry properties map, by decreasing the dimension of the space of possible materials.

As an alternative, regular triangular and hexagonal cells as shown in [Ostanin et al. 2018] can span a large range of material properties, and have an important advantage of being isotropic *by construction*, i.e., *any* geometry can be used on a cell as long as it has the symmetries of the cell (regular triangle or regular hexagon). While this significantly simplifies the problem of constructing geometries with target properties for regular cells, using distorted triangular or hexagonal cells as we need to tile an arbitrary shape, nullifies this advantage.

For triangular cells, we can enforce an additional symmetry by requiring the cells to be isosceles; in a periodic tiling, this is equivalent to using rhombic cells. At the same time, triangular cells are more restrictive in terms of choosing structure topology, if it needs to have the symmetry of the regular triangle. Finally, and perhaps most importantly, due to non-existence of regular tetrahedral tilings in 3D, methods based on quadrilateral tilings generalize better to 3D. For these,

reasons, we choose rhombic cells.

To define the actual microstructure geometry, we use an implicit function parametrized by an edge skeleton vertices and radii, using the method of [Panetta et al. 2017a]. This method, on the one hand, allows for explicit control of topology through the skeleton connectivity, on the other hand, allows for merges of parts of the structure when these run into each other due to parameter choices, which is critical for robustness of the optimization process.

4.5 HOMOGENIZATION AND SHAPE OPTIMIZATION

We briefly review the standard homogenized tensor computation on square cells and show how it extends to rhombic cells. We also discuss how the inverse homogenization problem can be solved using optimization.

4.5.1 HOMOGENIZATION AND SHAPE OPTIMIZATION ON SQUARE CELLS

Our formulation follows [Panetta et al. 2015a]. Suppose that a planar domain Ω is tiled by identical square cells Q , each with the same microstructure ω . At a large scale, we can consider averaged deformations: in the limit of infinitesimal cells, these deformations can be viewed as linear on each cell, and the actual deformation as a sum of the averaged one and a local fluctuation on the cell. The averaged deformation is the elastic deformation when the solid Ω is viewed as a completely filled volume with variable material properties defined by cell small-scale geometry, in the limit of zero cell size. The averaged deformation $\bar{\mathbf{u}}$ satisfies the *macroscopic* elasticity equation

$$-\nabla \cdot [\bar{\mathbf{C}} : \varepsilon(\bar{\mathbf{u}})] = \bar{\mathbf{f}} \quad \text{in } \Omega, \quad (4.1)$$

where the elasticity tensor $\bar{\mathbf{C}}$ is the effective elasticity 4-tensor with entries \bar{C}_{ijkl} and the $\varepsilon(\mathbf{u}) := \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the linearized Cauchy strain tensor of the averaged deformation. The colon

notation is used for contraction over the last two indices: $A : B = A_{ijkl}B_{kl}$, in the Einstein summation notation.

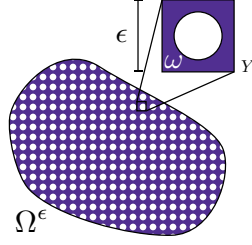


Figure 4.6: A tiling of a domain Ω with base cell Y having geometry ω [Panetta et al. 2015a].

On a single cell, in the limit of zero cell size in a periodic tiling, we can view elastic deformations as having a constant averaged strain $\varepsilon(\bar{\mathbf{u}})$ and a periodic “microscopic fluctuation” component \mathbf{w} , having zero average strain by periodicity. We consider fluctuations \mathbf{w} for three constant average strains forming a basis for all possible constant strains $e^{11} = \mathbf{e}_1 \otimes \mathbf{e}_1$, $e^{22} = \mathbf{e}_2 \otimes \mathbf{e}_2$, and $e^{12} = \mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1$. Then the expressions for the components of \bar{C} , are given by [Panetta et al. 2017a]:

$$\bar{C}_{ijkl} = \frac{1}{|Q|} \int_{\omega} C_{ijpq}^{\text{base}} [\varepsilon(\mathbf{w}^{kl}) + e^{kl}]_{pq} \, d\mathbf{x}. \quad (4.2)$$

To obtain the microscopic fluctuation term, the force balance equation in the cell Q is solved for each of the three basis strains:

$$-\nabla \cdot \left(C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \right) = 0 \quad \text{in } \omega, \quad (4.3a)$$

$$\hat{\mathbf{n}} \cdot \left(C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \right) = 0 \quad \text{on } \partial\omega \setminus \partial Q, \quad (4.3b)$$

$$\mathbf{w}^{kl}(\mathbf{x}) \text{ } Q\text{-periodic}, \quad (4.3c)$$

$$\int_{\omega} \mathbf{w}^{kl}(\mathbf{x}) \, d\mathbf{x} = \mathbf{0}. \quad (4.3d)$$

Inverse homogenization and shape derivatives. The conceptually straightforward, but technically difficult, way to solve the inverse homogenization problem for a given target tensor \bar{C}^*

is by shape optimization, i.e. by varying the geometry ω using shape parameters \mathbf{p} as variables to minimize a functional. In the case of inverse homogenization, the functional penalizes the difference of the homogenized elasticity tensor and the target \bar{C}^* :

$$J(\mathbf{p}) = \|\bar{C}(\mathbf{p}) - \bar{C}^*\|^2 + wE^{reg}(\mathbf{p}), \quad (4.4)$$

with the norm taken to be e.g., the Frobenius norm. A combination of regularizing terms can be used as E^{reg} , the most common being the volume of the structure defined by \mathbf{p} and the proximity to the initial solution [Panetta et al. 2015a, 2017a].

Unfortunately, due to non-uniqueness of the solution, resulting geometry parameters typically do not depend smoothly on the input material properties which has a number of negative consequences. We compare direct pointwise optimization to our approach in Section 4.8.

To minimize the functional (4.4) efficiently, we need derivatives of $J(\mathbf{p})$ with respect to the geometry parameters \mathbf{p} . Note that this requires differentiating $\bar{C}(\mathbf{p})$; the computation of \bar{C} requires solving elasticity PDEs on a domain depending on \mathbf{p} . This type of derivatives are called *shape derivatives*; these are computed by solving an *adjoint* PDE, which in our case is identical to elasticity, but with redefined volume forces. The details of the computation can be found in [Panetta et al. 2017a].

Discretization. To solve the PDEs (4.3), as well as similar equations needed for shape derivative computation, we use a standard FEM discretization. The domain is remeshed at each iteration using marching squares to extract the boundary and Triangle [Shewchuk 1996b] to mesh the domain.

We discretize the cell problems with quadratic triangle elements, which we found essential for accurate stress and homogenized tensor evaluation. We use straight-edged elements (subparametric

FEM) for representing the geometry to simplify meshing and the shape derivative formulas (edge nodes are placed at the edge midpoints).

In our experiments, running 2D homogenization usually took under 1s for meshes with an order of 4000 vertices. The mesh resolution (and hence the number of vertices) is determined by the maximal triangle area constraint (flag `-a` in `Triangle`) and the marching squares grid size (used to compute the boundary polygon), which we chose to be 5×10^{-4} and 256 respectively for most of our experiments. Figure 4.7 shows an example of the resolution we used.

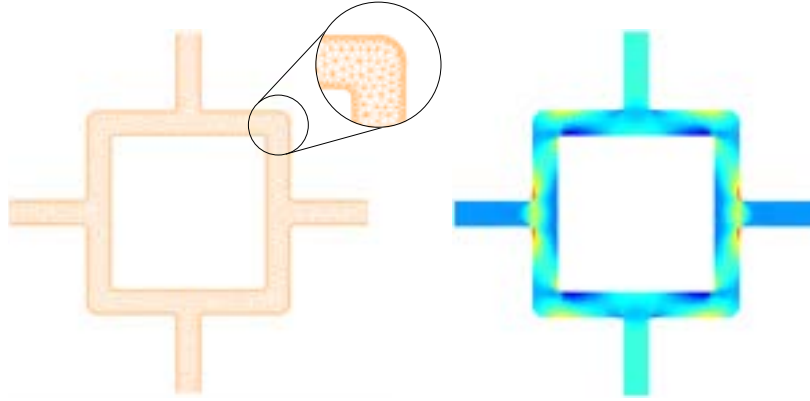


Figure 4.7: Left: An example of the cell geometry discretization. Right: fluctuation strain norm obtained from (4.3) for the constant strain e^{11} .

4.5.2 DEFORMED STRUCTURE HOMOGENIZATION

Computing the homogenization map $H(\mathbf{p})$ is the essential step in our process for obtaining a parametric family of structures. We use a modification of the standard homogenization method on square cells to handle arbitrary rhombi. Homogenization for an arbitrary rhombic cell (more generally parallelogram) can be transformed to homogenization on a square cell by a change of basis for elasticity tensors C^{base} and \bar{C} .

Suppose we have a cell R (rhombus with unit sides) and we want to write all our equations on a domain Q (unit square). Let F be the affine map $Q \rightarrow R$ and let its inverse be G .

Proposition 1. *Let $(\mathbf{e}_1, \mathbf{e}_2)$ be a nonorthogonal basis aligned with the sides of R ; in this coordinate system, the domain is a unit square, i.e., the map F maps $(\mathbf{e}_1, \mathbf{e}_2)$ to the unit coordinate vectors. If displacements \mathbf{u} satisfy the elasticity equation on domain R , $-\nabla \cdot [C : \varepsilon(\mathbf{u})] = \mathbf{f}$, then if $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{f}}$ are the displacements and forces expressed in the coordinate system $(\mathbf{e}_1, \mathbf{e}_2)$, they satisfy the elasticity equation*

$$-\nabla \cdot [C' : \varepsilon(\tilde{\mathbf{u}})] = \tilde{\mathbf{f}}, \quad (4.5)$$

where the transformed tensor C' components are given by

$$C'_{ijkl} = G_{pi}G_{qj}G_{rk}G_{sl}C_{pqrs}. \quad (4.6)$$

The proposition is verified directly by the change of variables in (4.1) and (4.3).

Note that the components of C can be expressed in terms of the components of C' using the inverse relation:

$$C_{ijkl} = F_{pi}F_{qj}F_{rk}F_{sl}C'_{pqrs}. \quad (4.7)$$

This observation applies both to the microscopic equation on the domain in R filled with the base material, as well as to the homogenized equation. This leads to the following 3-step procedure for determining the homogenized properties of a rhombic cell:

- Transform the base material tensor C^{base} to the coordinate system $(\mathbf{e}_1, \mathbf{e}_2)$ using (4.6).
- Compute the homogenized tensor \bar{C}' on Q , as described in Section 4.5.1.
- Transform \bar{C} back to the original coordinate system, using (4.7).

This allows us to solve the *forward* problem: given the geometry in a rhombic cell, compute the homogenized elasticity tensor. As the problem is reduced to square-cell problems, shape derivatives are computed exactly in the same way, and the problem is discretized using the same approach.

We describe our solution to the inverse problem in the next section.

4.6 MATERIAL-TO-GEOMETRY MAP CONSTRUCTION

In this section, we explain the steps for constructing a map \mathcal{P} from isotropic material parameters to geometry parameters. We assume in most of the exposition that C^{base} is fixed, and show how this assumption can be removed in Section 4.6.3. We use $E = 1$ and $\nu = 0$ for the base material, but the specific choice does not matter.

4.6.1 ORTHOTROPY

The fact that the rhombic cells always have orthotropic elasticity tensor is of critical importance in our construction, since it determines the dimension of the image of $H(\mathbf{p}, \alpha)$. We consider this fact in more detail and define the measures of deviation from *isotropy* tailored for orthotropic materials.

The rhombus has two reflection symmetries with respect to its diagonals, and we consider geometries that by construction have the same symmetries. This means that the periodic structure obtained by repeating the elementary cell R is invariant with respect to these two transformations. As a consequence, the elasticity tensor \bar{C} is also invariant with respect to these transformations. This implies *orthotropy*:

Proposition 2. *[Love 1944] If an elasticity tensor is invariant with respect to reflections about two*

orthogonal axes, then it is orthotropic with respect to these axes and in the coordinate system aligned with these axes has the form (in Voigt notation)

$$\begin{pmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{pmatrix} \quad (4.8)$$

This means, in particular, that only two additional constraints need to be satisfied to obtain an isotropic elastic tensor:

$$C_{11} = C_{22}; \quad 2C_{33} = (C_{11} + C_{22})/2 - C_{12} \quad (4.9)$$

We use two anisotropy measures, corresponding to these two constraints:

$$a_1 = \frac{|C_{11} - C_{22}|}{\max(C_{11}, C_{22})}, \quad a_2 = \frac{|(C_{11} + C_{22})/2 - C_{12} - 2C_{33}|}{\max(C_{11}, C_{22}, C_{33})}. \quad (4.10)$$

We note that principal directions of the tensor are exactly the diagonals of the rhombus, so cannot be set independently of the mesh directions.

4.6.2 MAP CONSTRUCTION FOR $\alpha = \pi/2$ (SQUARES)

Recall that the material-to-geometry map $\mathcal{P}(E, \nu; \alpha)$ for a given α , is not uniquely defined; we describe a simple way to obtain a unique initial map on a smaller initial domain $D(\alpha)$, for $\alpha = \pi/2$, which we then expand. For clarity, we drop the dependence on α in this section.

We explain the general form of construction, to clarify how it can be used for other types of structures (e.g., 3D), and explain how it applies in the specific case of planar rhombic structures. Recall that $p = 8$ is the number of geometric parameters and we denote $m = 4$ the number of material parameters for 2D orthotropic materials.

We start with the homogenization map $H : B_p \rightarrow \mathbb{R}^m$, $p \geq m$. In general, a $p - m$ -dimensional submanifold of B_p is mapped to each orthotropic elasticity tensor in $H(B_p)$. *Iso* corresponds to the set of possible isotropic materials in the space of material parameters \mathbb{R}^m , e.g., it is defined by equations $a_1 = 0$ and $a_2 = 0$ for orthotropic materials.

Our key idea is to restrict H to a carefully chosen m -dimensional affine subspace $V \subset \mathbb{R}^p$, for which H has large coverage, i.e. $H(V \cap B_p) \cap \text{Iso}$, the set of isotropic material properties covered by geometries in V , is not too far from $H(B_p) \cap \text{Iso}$. We refer to V as the *transversal subspace*.

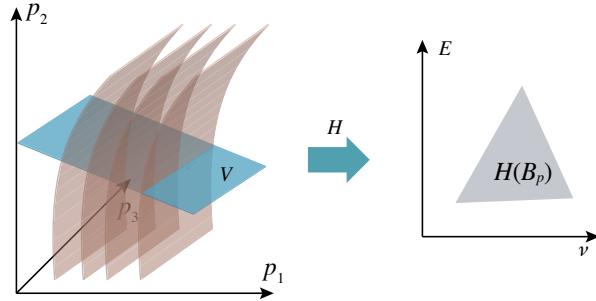


Figure 4.8: Transversal subspace V illustrated for the simplified case of a 3-parametric structure and 2-dimensional property space (E, v) (In our construction, we consider spaces that are 8 and 4-dimensional respectively). The curved surfaces are isosurfaces of $E(\mathbf{p})$. The map H , restricted to V , is bijective.

$H' : V \rightarrow \mathbb{R}^m$, the restriction of H to V , is locally injective near all points where it is non-degenerate.

The map $\mathbf{p}^{\text{ortho}} : D = H(B_p) \rightarrow \mathbb{R}^p$, can be constructed by inverting H' as described below, and its restriction to isotropic materials $\text{Iso} \cap D$ yields the desired material-to-geometry map \mathbf{p} . Note that if non-isotropic orthotropic materials are desired, this map also can be used to compute these.

As the $H(\mathbf{p})$ depends on \mathbf{p} smoothly, almost everywhere we expect $\mathcal{P}(E, \nu)$ to be smooth as well: the only places where a jump is possible is in the areas where the differential of $H(\mathbf{p})$ is degenerate, or at the boundary of $H(B_p)$, which naturally defines the domain D for $\mathcal{P}(E, \nu)$. In practice, we have not observed the degeneracies of $\mathcal{P}(E, \nu)$ for the shape and material parameter ranges that we have considered, and the smooth approximation we obtain matches the samples closely.

Defining the transversal subspace V . We start with identifying a m -dimensional affine subspace containing a large range of geometric configurations with isotropic elasticity tensor, i.e., has large isotropic coverage. A direct approach to this would be to sample H sufficiently densely on B_p , select points for which the anisotropy measures a_1 and a_2 are small, and fit an m -dimensional hyperplane to this set, e.g., via PCA. Unfortunately, densely sampling the p -dimensional set (even for $p = 8$) is prohibitively expensive, as each evaluation of H requires meshing and solving three elasticity problems at sufficiently high resolution.

By considering a coarse sampling of B_p , we experimentally observed that, for our setting, B_p contains an *axis-aligned* hyperplane subset of dimension p' , $p \geq p' \geq m$, which has good coverage. Intuitively, this corresponds to fixing the geometry parameters that can be fixed without decreasing the range of material properties; we refer to these as *redundant*. This reduction leads to a practical two-stage algorithm: first, remove the redundant parameters that have the least effect on coverage. Once no dimensions can be dropped, use denser sampling and PCA for the final reduction to m dimensions. In our case, $p' = 5$, so only one extra dimension needs to be eliminated.

Next, we discuss these two steps in greater detail:

1. *Finding redundant parameters.* We compute a coarse regular grid sampling $q^j = H(\mathbf{p}^j)$ of the full homogenization map $H : B_p \rightarrow \mathbb{R}^m$. Each geometric parameter p^i is uniformly sampled at values c_k^i , $k = 1 \dots n_i$. In our case, a total of $540k$ data points were collected (see

more information in Appendix 4.10.2). Then we extract a subset of \mathbf{p}^j with the deviation from anisotropy sufficiently small (in our case 0.005 for both a_1 and a_2). Consider slices π_{ik} consisting of all samples \mathbf{p}^j with a fixed value of $p_i^j = c_k^i$: by direct check of the coverage area of each slice, we identify the geometric parameters i for which there is a k such that coverage of π_{ik} is close to complete coverage, i.e., the coverage area obtained with the initial sweep, where we use all parameters as variables. Figure 4.9 shows the observed coverage for a sequence of expanding sets of fixed parameters.

For the rhombic structures, we observe that the offset parameters p_1, p_2 , and the thickness parameter p_5 all have such values c_k^i (0.5, 0.25, and 0.3 respectively). This is a particularly convenient set of parameters to fix, since this allows us to merge cells *perfectly* as the radius on the boundary is fixed and thus the same for all cells. Fixing p_1, p_2 , and p_5 reduces the

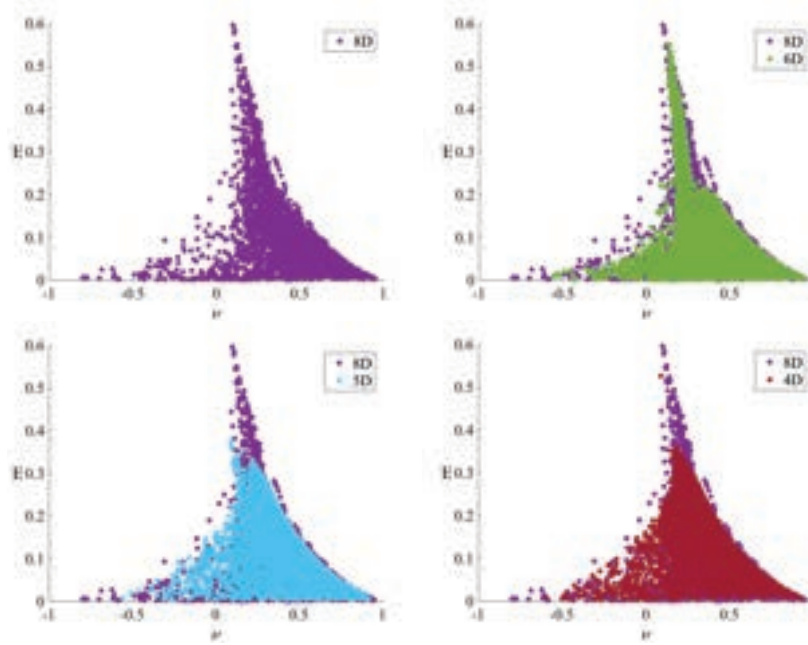


Figure 4.9: Material properties coverage for the subset of samples of B_p with constant values of p_1 and p_2 fixed (6D space, samples shown in green), p_1, p_2, p_5 fixed (5D space, samples shown in blue) and V (4D space, shown in red), compared to the full set of samples \mathbf{p}_j , shown in purple.

dimension of the geometry space to just 5, leading to a 5D affine subspace V^{5D} . It remains

to reduce the dimension by 1.

2. *Reduce to m dimensions by PCA.* The restriction of the map H to $V' \cap B_p$, where V' is the p' -dimensional subspace obtained by dropping dimensions, can be sampled much more densely (we use 15 samples per dimension for 5-dimensional V'). To obtain the m -dimensional transversal space V , we approximate $H(\mathbf{p})$ with a linear map on B_p : $H(\mathbf{p}) \approx Q\bar{\mathbf{p}} + H_0$, where Q is a $m \times p'$ matrix, and $\bar{\mathbf{p}}$ is obtained from \mathbf{p} by discarding the fixed coordinates. We compute Q and H_0 using PCA. Let W be the matrix spanning the nullspace of Q , i.e., $QW = 0$, and W has dimension $p' \times (p' - m)$ and maximal rank. For all points $\bar{\mathbf{p}} + Wu$, where u is $(p' - m)$ -dimensional, the value of the map is constant. Then the condition $W^T(\bar{\mathbf{p}} - \bar{\mathbf{p}}_0) = W^T\bar{\mathbf{p}} + c_0 = 0$, where $\bar{\mathbf{p}}_0$ is a point in V' , defines an m -dimensional subspace $V \subset V'$, perpendicular to the constant-value affine spaces, on which $Q\bar{\mathbf{p}} + H_0$ is one-to-one.

For rhombic structures, we obtain $p' - m = 1$, and W has dimensions $p' \times 1$, i.e., it is a vector in this case. W is approximated well by a vector with nonzero components $v_3 = 1$ and $v_7 = 1$, and with $c_0 = -0.82$.

This gives us a complete description of $V = \{\mathbf{p} \mid p_1 = 0.5, p_2 = 0.25, p_5 = 0.3, p_8 = 0.82 - p_4\}$. Geometrically, V is described as the space of structures where node 2 is fixed at the default position at barycentric coordinates $(0.5, 0.5, 0)$, the radius of node 1 connecting the cell to other cells is fixed at 0.3, and the radius and displacement of node 4 are related linearly.

Inverse of H' on the isotropic subspace. First, we construct a piecewise-linear interpolant of the samples of H' . The map H' is defined as the restriction of H to $V \cap B_p$. We sample H' on a regular grid in V . For the rhombic structure, we use p_3, p_4, p_6, p_8 as coordinates for sampling. We sample at the points of the grid $q_j = (E_j, v_j)$ contained inside B_p .

As samples q^j are on a regular m -dimensional grid, they form mD cubes, which we then subdivide into m -simplices. We discard all simplices with one of the two possible orientations (as H' is

not guaranteed to be injective everywhere, simplices may have both orientations; we choose the predominant orientation to be the “correct” one). The map is defined by linear interpolation on each simplex.

For each sample q on a regular 2D grid of samples in Iso , we search for m -simplex S_ℓ in V , such that $H(S_\ell)$ contains q . While the simplices found in this way may be not unique, in practice we do not observe this problem. The value of $\mathcal{P}(q)$ is obtained by interpolating the values of \mathbf{p} at the corners of each simplex S_ℓ . Finally, we fit a B-spline to the sampled values of q with Laplacian regularization. We use 20×20 control points for rhombic structures.

Increasing coverage of the material-to-geometry map. Constraining the map H to V makes the map invertible, and we observe the inverse to be smooth. However, this also restricts the coverage, as the inverse image $H^{-1}(Iso \cap H(B_p))$ in B_p is not necessarily contained in V . We can expand the range by the following procedure using optimization-based inverse homogenization, similar to [Zhu et al. 2017].

For each regular-grid sample point q_j in (Iso) , for which $\mathcal{P}(q_j)$ is not defined, but which is adjacent to a point q_ℓ where \mathcal{P} is defined, we initialize the shape optimization for the functional (4.4) using the spline approximation to \mathcal{P} obtained above evaluated at q_j . Then we optimize using all 8 geometry parameters as variables (and not only the 4 independent variables used in the sampling procedure), thus taking the value of $\mathcal{P}(q_j)$ out of V . If the optimization converges to a value sufficiently close, using a tolerance of 0.005 for Poisson’s ratio and 0.001 for Young’s modulus, we include the additional point in the set, and refit the splines to the new set of points.

We note that, in principle, this process may suffer from the same flaws as the original inverse homogenization of [Panetta et al. 2015a] (cf. Figure 4.10) as the map $\mathcal{P}(E, \nu; \alpha)$ may become non-smooth for lower values of α . However, the procedure defined above prevents this from happening, and the map $\mathcal{P}(E, \nu; \alpha_{min})$ remains smooth (Figure 4.18).

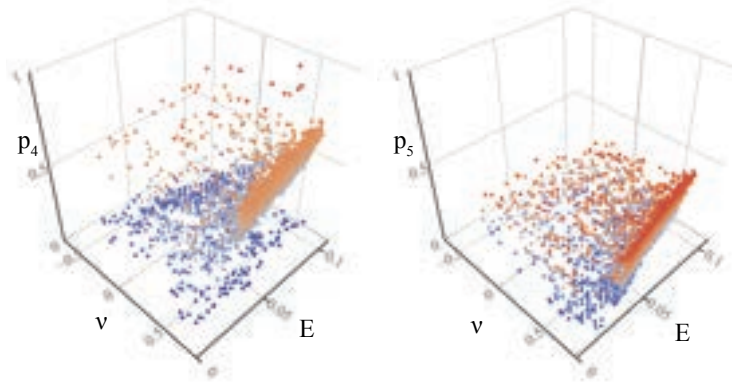


Figure 4.10: Plots of parameters p_1 and p_4 for pointwise inverse homogenization. Note the lack of smoothness; other parameters also have a varying degree of noise. The color in the charts represents the geometric parameter value, varying from blue to red.

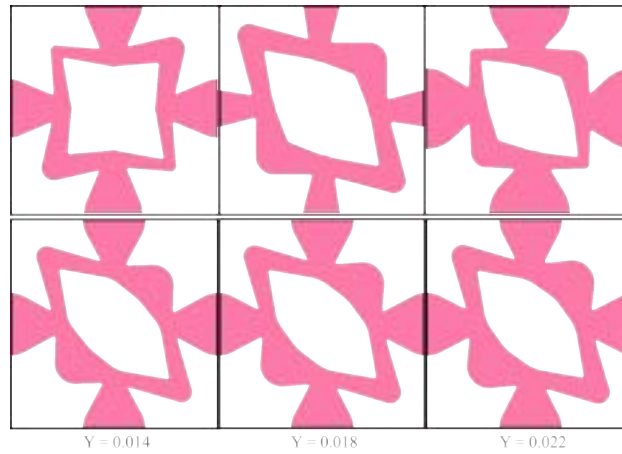


Figure 4.11: Changes of geometry as a result of small changes in Young's modulus Top: obtained by pointwise inverse homogenization [Panetta et al. 2017a]; Bottom: our family.

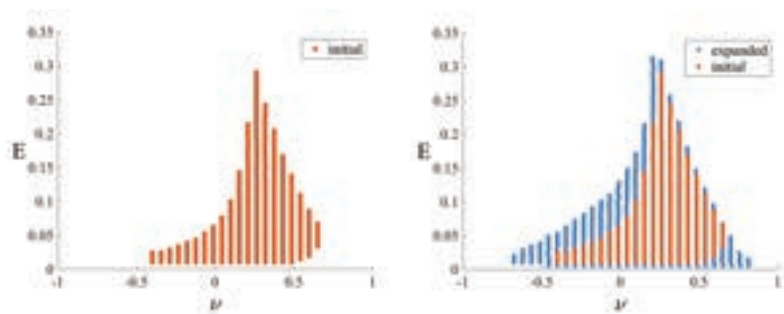


Figure 4.12: Expanding the initial material-to-geometry map coverage. Left: original coverage; Right: expanded coverage.

Bounds for the domains $D(\alpha)$. For applications using optimization of material properties on a partitioned domain (Section 4.8), it is useful to ensure that the values used in the optimization stay within the coverage zone of the material-to-geometry function \mathbf{p} . To keep these constraints efficient, we approximate the coverage area by a convex polygon bounded by a set of a maximum of 6 lines: 2 horizontal lines defining minimum and maximum Young's modulus E_{min} and E_{max} , 2 vertical lines for minimum and maximum Poisson's ratio ν_{min} and ν_{max} , and 2 additional slanted lines, s_l and s_r , that allow for larger non-rectangular domains (see Figure 4.13).

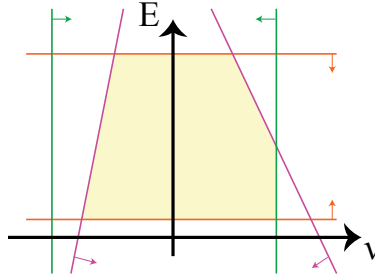


Figure 4.13: Lines defining the shape of our convex domain.

The lines are defined to maximize a weighted sum of the area and the height (maximum Young's modulus) of the convex polygon while keeping it completely inscribed in the domain coverage region. Below, we explain this process in more detail.

In our coverage data, we first identify two sets of points on the boundary of our covered area. For each value of Young's modulus, the minimum achieved Poisson's ratio is added to B_l , while the maximum should be in B_r . Then, we iteratively run the following two steps, trying different values for lines ν_{min} , ν_{max} , E_{min} and E_{max} at each time:

1. *Filtering.* We filter out boundary points in B_l and B_r that are outside the square defined by ν_{min} , ν_{max} , E_{min} and E_{max} , creating new boundary sets B'_l and B'_r .
2. *Optimization.* We optimize the slanted lines as follows. We create one linear constraint for each boundary point b in B'_l (resp. B'_r), making sure that slanted line s_l (resp. s_r) is to the

right (resp. left) of b . After building our constraint matrix, we run the `fmincon()` solver in MATLAB to optimize our objective function (composed of area and height of the convex domain) given our new linear constraints.

At the end, the objective function values obtained for all iterations are compared and we select the 6 lines corresponding to the best value achieved. In our own case, we decided to fix minimum and maximum Young's modulus (E_{min} and E_{max}) to 0.005 and 0.32, due to our observed coverage. However, we tried multiple values for maximum and minimum Poisson's ratio (ν_{min} , ν_{max}). Notice that these values can also be chosen according to the application.

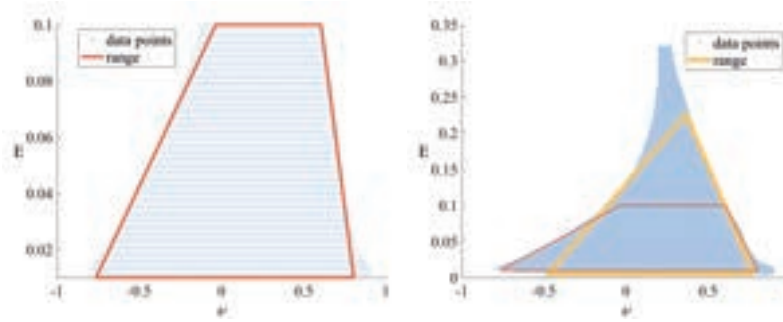


Figure 4.14: Domain coverage for $\alpha = \pi/2$, with approximating linear bounds. Left: resulting polygon when restricting minimum and maximum Young's modulus to 0.01 and 0.1; Right: convex polygon (in yellow) when Young's modulus minimum and maximum values are unlimited. We show a comparison with the bounds in the constrained case (in red). Depending on the Young's modulus range the application needs, a different polygon may be obtained in the optimization. If the application requires larger stiffness values, the polygon on the right should be preferred. On the other hand, if a larger range on Poisson's ratio values is needed, the polygon in red is a better option.

4.6.3 ARBITRARY ANGLES AND BASE MATERIALS

We extend the map to all angles in the range $[\alpha_{min}, \pi/2]$. We use $\alpha_{min} = \pi/4$ since the coverage becomes very small below this value. In our experiments, the range $[\pi/4, \pi/2]$ proved to be sufficient, as, after quad optimization, the minimum angle for quads was larger than $\pi/4$. It is however possible that for complex geometries smaller angles are needed.

We perform the extension incrementally for a sequence of angles $\alpha_i = \pi/2 - i d$, where $d = 1.25^\circ$.

Using the geometric parameters for α_{i-1} as a starting point, we use shape optimization to obtain a point for the same target value (E, ν) but for α_i , with a regularization term $\|\mathbf{p} - \mathbf{p}_0\|^2$ penalizing deviation from the initial value of each parameter, where \mathbf{p} and \mathbf{p}_0 represent the current and the initial parameter values during the optimization, respectively.

Once the whole range of angles is covered, we fit a 3D spline in variables (E, ν, α) to the whole set of points. The resulting change of geometry parameters as a function of α is smooth (Figure 4.15).

For spline fitting, we use least-squares on the whole cube. An extra regularization term, the Laplacian of the function at grid points, is added in the cost, which promotes smoothness and eliminates the null space corresponding to the regions with a few or no data points.

Complete map description. The final map \mathcal{P} is defined as:

- a 3D uniform spline approximation on a $12 \times 12 \times 12$ grid of 8 geometric parameters, for a total of 8×12^3 coefficients;
- a 1D splines in α for each of the coefficients defining the lines of the convex polygon approximating the domain coverage region $D(\alpha)$.

The complete description of the map is relatively compact, requiring less than 14k coefficients. The plots for the eight components are shown in Figure 4.17 for $\alpha = \pi/2$, while Figure 4.18 shows plots for two of the parameter when $\alpha = \pi/4$. In addition, Figure 4.11 illustrates (non-)smoothness of the map obtained by pointwise shape optimization, compared to our map, for small changes of E and ν for three choices of (E, ν) .

Change of the base material. In principle, we would need to compute a new map \mathcal{P} each time the base material changes, as the homogenization map H depends on the choice of material. However, two observations show that \bar{C} for any base material (E', ν') can be obtained from a base material (E, ν) by a simple transformation.

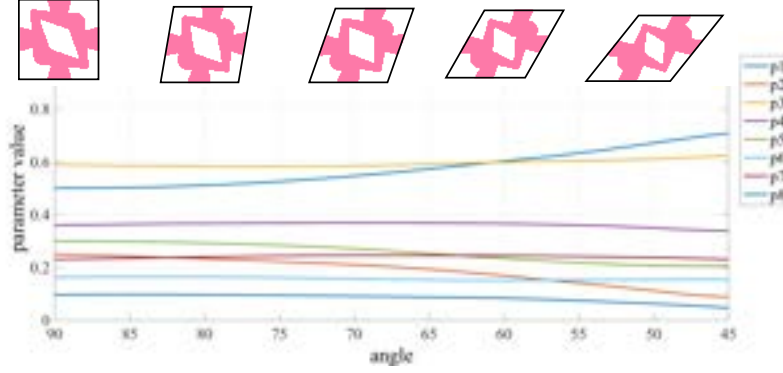


Figure 4.15: Dependence of the geometry parameters \mathbf{p} on the angle, for $E = 0.076$ and $\nu = 0.263$ (using base material $E = 1$, $\nu = 0$).

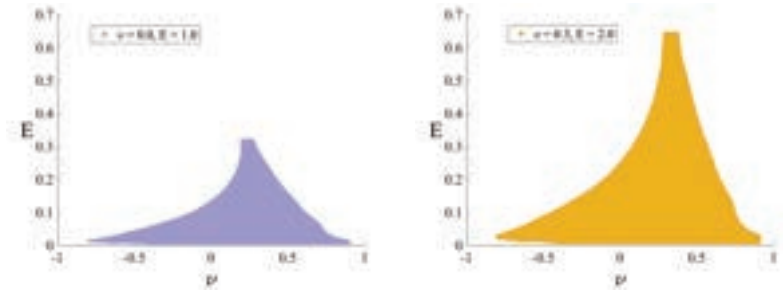


Figure 4.16: Left: coverage for $\nu = 0$, $E = 1$. Right: coverage for $\nu = 0.3$, $E = 2$.

The first simple observation is that the expressions for the components of \bar{C} and the solution for (4.3) depend linearly on the base elastic tensor C^{base} , i.e., if the base Young's modulus E is replaced with sE , the homogenized one E^H becomes sE^H .

The less trivial part are the changes due to changes in the Poisson's ratio. Fortunately, if ν is replaced with ν' , the new homogenized elasticity tensor \bar{C}' is characterized by the CLM theorem, a corollary of which for isotropic materials can be formulated as follows.

Theorem 4.1. [*Cherkaev et al. 1992*] Suppose a periodic 2D medium with holes is filled with material with Young's modulus E and Poisson's ratio ν , and the effective (homogenized) properties of this medium are E_H and ν_H . Suppose ν is replaced with ν' . Then the effective Young's modulus is preserved,

and the Poisson's ratio changes as

$$\nu'_H = \nu_H - (\nu - \nu') \frac{E_H}{E_b}.$$

This immediately leads to the change of variables in the function $H(E, \nu)$ that corresponds to the change of base material properties $(E, \nu) \rightarrow (E', \nu')$:

$$\begin{aligned} E'_H &= \frac{E'}{E} E_H, \\ \nu'_H &= \nu_H - (\nu - \nu') \frac{E' E_H / E}{E'} = \nu_H - (\nu - \nu') \frac{E_H}{E}. \end{aligned} \tag{4.11}$$

Discussion of alternative approaches. The closest works on 2D microstructures are [Ostani et al. 2018] and [Martínez et al. 2019], which introduce low-parametric families of structures. However, both approaches are limited to regular grids in a profound way: they rely on symmetries of triangular/hexagonal grids to produce isotropic effective properties, a requirement that is lost on deformed lattices with fewer symmetries.

Regular quad/hex grids are used by topology optimization-based methods [Schumacher et al. 2015; Zhu et al. 2017], and in the family of structured proposed in [Panetta et al. 2015a, 2017a]. While these methods could be adapted to rhombic tiles by applying a bilinear map to their structures, this construction introduces large errors (see Section 4.8 for more details).

Our construction is based on the optimization techniques introduced in [Panetta et al. 2015a, 2017a], by addressing two of their limitations: (1) we lift the regular lattice cell assumption (for the 2D planar case), directly targeting rhombic tiles and (2) we address the lack of smoothness of \mathcal{P} (cf. Figures 4.17 and 4.11).

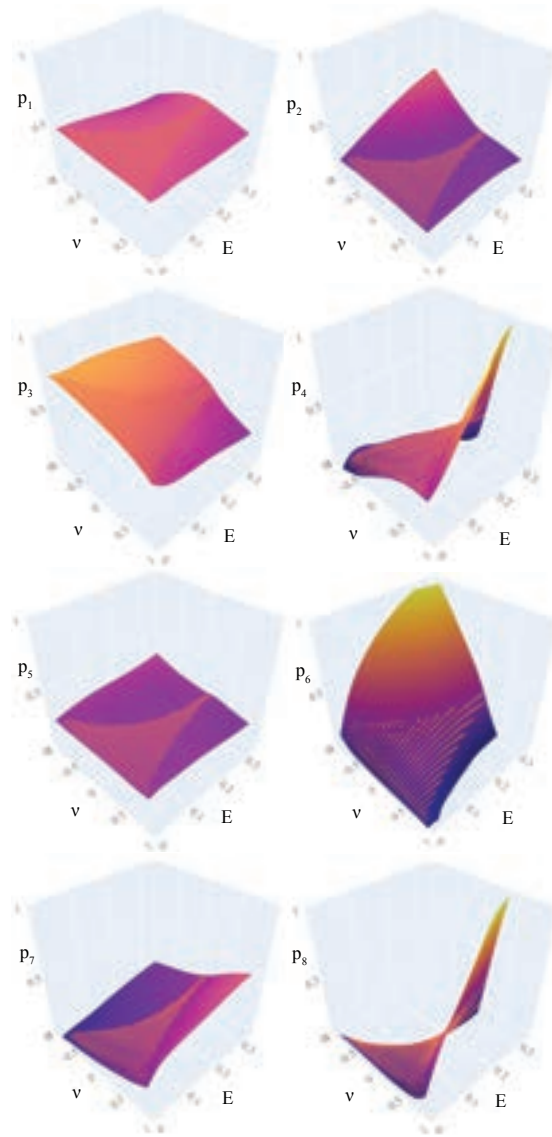


Figure 4.17: Plots of parameters $p_1 \dots p_8$ as functions of (E, v) for $\alpha = \pi/2$. The color of the surface corresponds to parameter values.

4.7 QUAD MESH OPTIMIZATION

We tessellate the polygonal domain Ω into (nearly-)rhombic elements by starting with a quadrangulation of Ω and optimizing the shape of its cells.

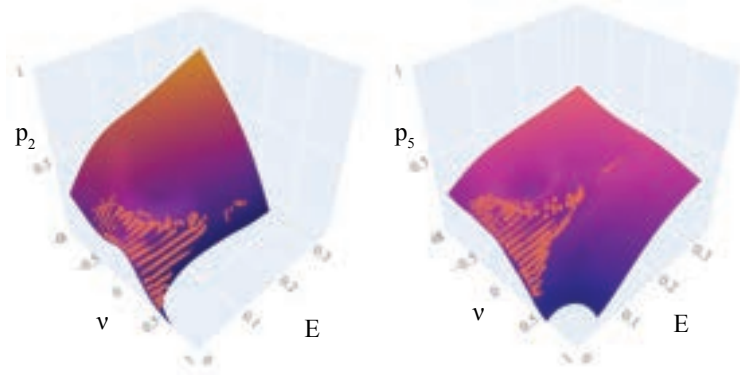


Figure 4.18: Plots of parameters for $\alpha = \pi/4$. The color of the surface corresponds to the parameter values.

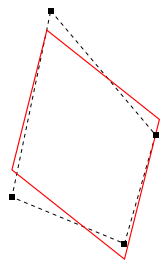
Quad mesh generation. Our algorithm can be applied to any planar quadrilateral mesh: for the examples in the paper, we used a combination of meshes automatically generated using the *Integer Grid Maps* algorithm [Bommes et al. 2013] (Figure 4.34, disk), *Pattern-Based Quadrangulation* [Takayama et al. 2014] (Figure 4.34, bar), and manually designed using Blender (Figure 4.34, pliers, ghost). Generally, denser meshes are easier to optimize for our algorithm since they have more degrees of freedom.

Quad to rhombi projection.

Our algorithm uses an alternating optimization inspired by ShapeUp [Bouaziz et al. 2012]: we compute a target rhombic shape for each quad (see inset), and then optimize the vertex positions to match the target shape of each element in least-squares sense. To obtain the ideal rhombic shape R for a quad F , we fit a linear transformation T of a canonical unit square Q (a parallelogram) to F , after translating its barycenter to the origin:

$$\min_T \sum_i \|f_i - Tq_i\|^2, \quad (4.12)$$

where f_i and q_i are the vertices of F and Q respectively. Letting θ_T be the smaller angle of the parallelogram, we define the unit target rhombic shape R_u as the rhombus with unit length edges and minimal angle $\max(\theta_T, \theta_m)$, where θ_m is the smallest angle covered by our parameter sweep



(45 degrees). Capping the minimal angle pushes the optimization toward rhombic shapes within the reachable range of our parametric family. The scaled target rhombic shape R is computed by translating (by a vector t) and scaling (by a diagonal matrix A) R_u to best fit the quad F :

$$\min_{A,t} \sum_i \|f_i - Ar_i - t\|^2, \quad (4.13)$$

where r_i and f_i are the vectors of vertices of R_u and F respectively.

Exponential rhombic distortion energy. Equipped with a target reference for each quad, we define a distortion energy by penalizing the deviation of each element from its ideal rhombic shape using the symmetric Dirichlet energy [Smith and Schaefer 2015]:

$$\mathcal{D}(J_F) = \|J_Q\|_F^2 + \|J_Q^{-1}\|_F^2,$$

where J_F is the Jacobian of the bilinear map mapping F to its target rhombic shape R . Ideally, we would like to minimize the L_∞ norm of the distortion integrated over the entire domain Ω , which is however a challenging problem to solve numerically. We opt for an approximation, using a continuation approach [Rabinovich et al. 2017b] minimizing the following exponential energy for increasing values of s :

$$E_{\mathcal{D}}(x) = \sum_{Q \in \Omega} A_Q e^{s\mathcal{D}(J_Q)} \quad (4.14)$$

where A_Q is the area of the quad Q .

Boundary projection. Without additional constraints, minimizing (4.14) would change the shape of the boundary of Ω (Figure 4.19). We add a penalty term to avoid changes to the boundary, while allowing nodes to slide on it:

$$E_{\partial}(x, y) = \sum_{x_i \in \partial\Omega} \|x_i - y_i\|^2, \quad (4.15)$$

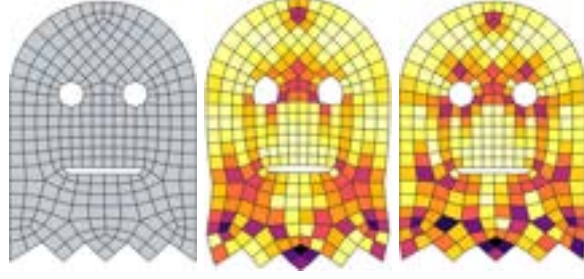


Figure 4.19: A quadrilateral mesh (left) is optimized with (right) and without (middle) the boundary term. The shape of the eyes and the mouth is lost in the middle image, while it is preserved on the right. The color for each quad maps to α (the minimum angle of the cell).

where x_i are the coordinates of a boundary vertex of a point on the boundary of the quad mesh, and $y_i = \Phi(x_i)$ is the closest point on the boundary of Ω .

Optimization. We create a rhombic mesh by minimizing the combined energy:

$$E(x) = (1 - \lambda)E_{\mathcal{D}}(x) + \lambda E_{\partial}(x, \Phi(x)), \quad (4.16)$$

with $\lambda = 10000$ a weight balancing the two terms. We use the solver proposed in [Rabinovich et al. 2017b] to find vertex positions minimizing eq. (4.16). This solver uses a local/global optimization strategy in order to minimize eq. (4.16). In the local step, the target rhombi are rotated to minimize $E_{\mathcal{D}}(x)$ (eq. (4) in [Rabinovich et al. 2017b]), and we compute $y_i^k = \Phi(x_i^k)$, the projection on the boundary $\partial\Omega$ for border vertices after iteration k . In the global step, the Jacobian J_Q in $E_{\mathcal{D}}$ is replaced by $J_Q^k = J_Q(x^k)$, $E_{\partial}(x, \Phi(x))$ is replaced by $E_{\partial}(x, y^k)$, and E is minimized wrt x .

4.8 EVALUATION AND EXAMPLES

In this section, we explore the advantages of our approach, compared to "naive" application of existing techniques, and its limitations. Specifically, we explore the following features of our method:

- the use of boundary-aligned, irregular quad meshes instead of regular grids with cut cells;
- the optimization of the cell geometry structure taking cell shape into account, compared to remapping geometry optimized for square cells;
- continuous dependence of geometric parameters on material parameters, compared to pointwise inverse homogenization used in previous work;
- mesh independence and the effects of deviation from rhombicity on the accuracy of the method.

Finally, we show how our method handles a number of test problems. Many of the comparisons are performed on models with variable material properties generated using material optimization, as in [Panetta et al. 2015a]. Specifically, the deformation on one part of the boundary is given, and the material properties for interior cells are optimized to obtain a user-defined target deformation on an other part of the boundary. For example, we use a plier shape for several examples; the prescribed deformation moves the handles together, and the target deformation moves the jaws together. In the optimization, the cells of the mesh are used as finite elements with constant material properties.

Comparison with regular grids with cut cells. The simplest approach that allows one to use existing techniques with microstructures optimized for squares is to overlay a regular grid on the geometry of the shape and remove parts of the cells outside the input shape boundary. We used a regular grid with square side length equal to the average edge length of our rhombic geometry to obtain approximately the same number of cells. The material optimization was performed with the same boundary conditions and target deformations but for each mesh separately (cut regular grid vs rhombic cells), rather than, e.g., interpolating target material properties from one mesh to the other. For the purposes of FEM simulation used in the material optimization, cut

cells are triangulated. In both cases we assign the geometry for each cell based on the computed material. As an additional step for the regular cell grid, we crop the final fine-scale geometry using the original object’s boundary. We run the deformation simulation on full fine-scale geometry for comparison. Figure 4.20 shows an example with significant deformations compared to the ground truth. We note that one feature of our method, the continuous dependence of geometric parameters on material parameters remains useful even for objects that can be meshed with regular grids, and improves accuracy.

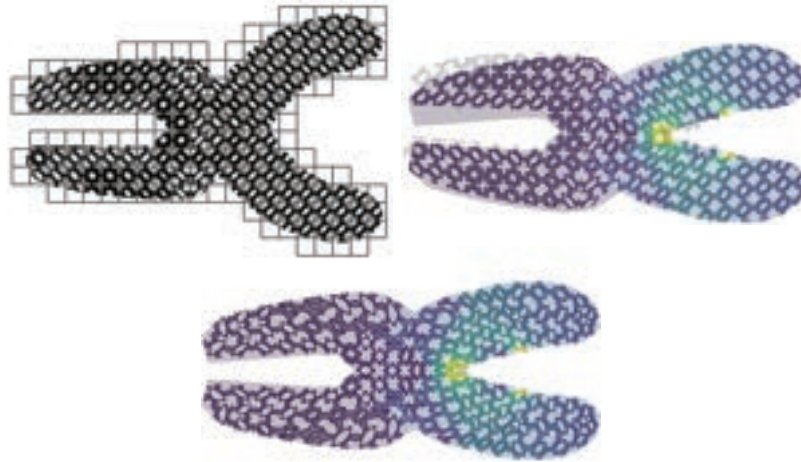


Figure 4.20: Top left: example of a cropped regular grid of perfect square cells. Top right: deformation using the model produced with square grid as shown on the left. Bottom: result using our new method.

To experiment with higher resolution quad meshes, we applied one level refinement on each of the two configurations shown in Figure 4.20, keeping the same material distribution. The result is presented in Figure 4.21. For both methods, the error is reduced, although it is still considerably higher when using regular grids with cut cells. Notice, however, that (although being a valid experiment) refinement is not a very practical solution in real world scenarios since most fabrication technologies have resolution limitations.

Comparison with remapping square cell structures. In principle, the structures obtained for square cells can be directly applied to rhombic cells, by simply applying a bilinear transformation to the geometry. Figure 4.22 shows the effect of not using structures specialized to set with specific



Figure 4.21: Left: deformation result after one level of quad mesh refinement when using our new method. Right: result using regular grid with cut cells.

angles: the increase in the error is substantial.

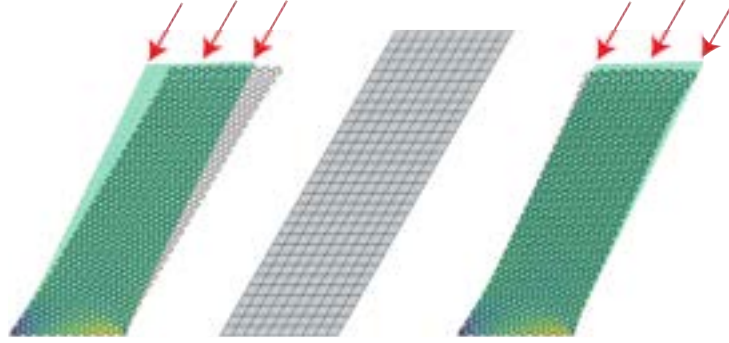


Figure 4.22: Middle: non-deformed solid shape. Left: deformation of the structure using geometry optimized for $\alpha = \pi/2$ everywhere; the solid shape in the background shows the reference deformation of a single-material solid. Right: using correct cell geometry with parameters $\mathcal{P}(\alpha)$ everywhere. The microstructure has material parameters $E = 0.02$ and $\nu = -0.4$. Bottom part is fixed (0 Dirichlet), and on the top part we apply a force parallel to the slanted side (at an angle of $4\pi/3$).

Comparison with pointwise inverse homogenization. In this experiment, the same mesh and the same variable material properties, resulting from optimization, are used with two different ways of assigning a structure to each cell (Figure 4.23). The first approach performs the nearest-neighbor lookup in the database of structures obtained by pointwise shape optimization [Panetta et al. 2017a], the other using our smooth parametric family $\mathcal{P}(E, \nu; \alpha)$. In both cases, the structures are mapped to the arbitrary quads of the mesh by a bilinear map. We observe that the result is closer to the reference simulation using a variable-material solid. Figure 4.24 shows that the $E - \nu$ space coverage of two methods is close.

Mesh independence. If all cells are assigned the same effective material properties, the resulting deformation should not change significantly as we change the domain discretization.

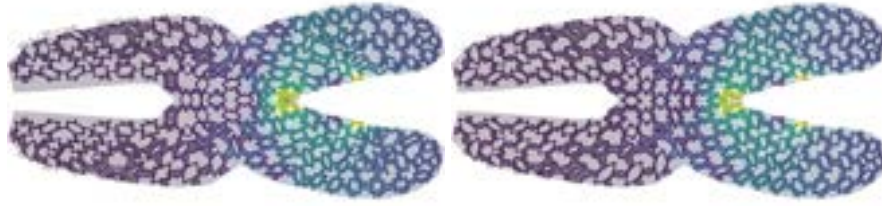


Figure 4.23: Left: deformation of an object constructed using a table of structures obtained using pointwise inverse homogenization as in [Panetta et al. 2017a]. Right: deformation of the object constructed using our family, with the same boundary conditions. The gray shape in the background shows the ground truth obtained by simulation on a solid quad mesh with materials assigned per quad.

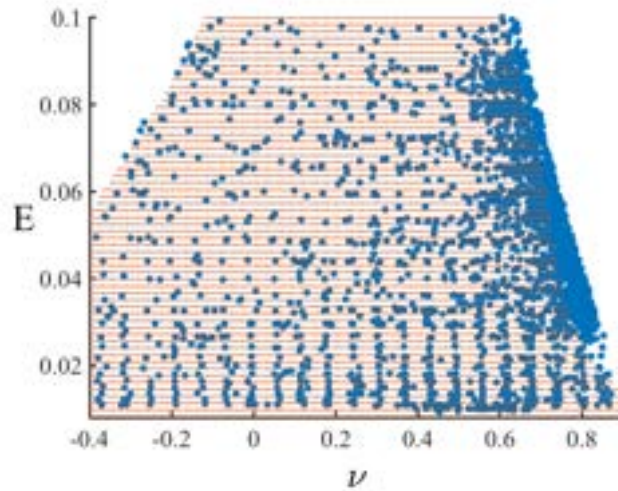


Figure 4.24: Coverage in (E, ν) plane for $\alpha = \pi/2$ obtained by pointwise shape optimization (blue) in the range $E = 0.01 \dots 0.1$ and $\nu = -0.4 \dots 0.9$. The coverage of our family is shown in orange.

We performed two experiments to evaluate the effect of changing the mesh. The first experiment (Figure 4.25) confirms that the deformations obtained for an irregular mesh on a bar, which requires different rhombic structures for different cells, are nearly the same as for a regular mesh with identical structure, optimized for the same square cell everywhere.

In the second experiment, we study how deviation from rhombic shape affects the result, when the cell shape is intentionally distorted to be far from rhombic. We create a *twisted structure* – a compound cell subdivided into 4 non-square subcells, and compute the properties of the material

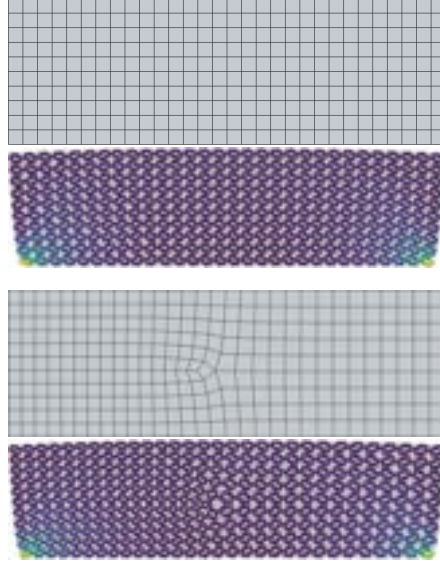


Figure 4.25: Top: deformation of a shape with constant assigned materials and regular connectivity. Bottom: deformation of the same shape, with irregular mesh connectivity. The deformed pattern (simulated using FEM on a fine mesh) is compared to the simulated homogeneous mesh with equivalent material properties ($E = 0.05$, $\nu = 0.4$). In both cases, a uniform vertical load is applied on the top, while we keep a 0 Dirichlet condition at the bottom.

composed of such compound cells, using homogenization on the whole cell. Specifically, starting from a 4×4 square quad mesh on the cell, with each quad assigned $E = 0.05$, $\nu = 0.0$, we gradually rotate the cross formed by the edges incident at the central vertex of the mesh (5 degrees at a time), and optimize the shape of the elements of the twisted mesh (keeping the cross fixed, Figure 4.26). Figure 4.27 shows how the Frobenius norm of the distance between the target elasticity tensor used to obtain the structure in each subcell and the homogenized elasticity tensor of the compound cell changes with rotation angle. In the plot, we also show the error we observe if we use the structure optimized for $\alpha = \pi/2$ everywhere.

Notice how the error stays stable when we increase the rotation angle, which is not the case when using only the structures for $\alpha = \pi/2$ in all quads. We emphasize that this experiment was intentionally conducted with a very coarse grid of subcells. The error of our method further decreases for a finer grid: if we subdivide each quad and placing the same initial structure in all

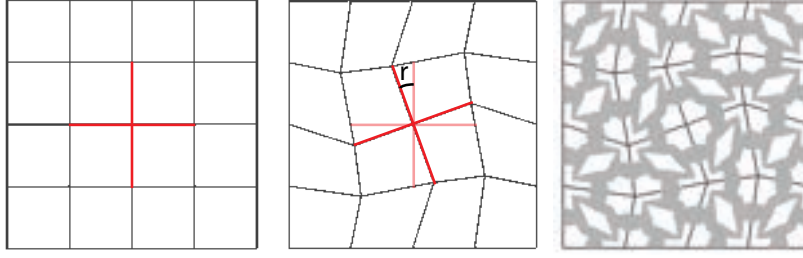


Figure 4.26: An example of the twisted structure. Left: initial mesh with perfect squares. Middle: twisted mesh by a rotation r of 20 degrees. Right: final structure after rotation.

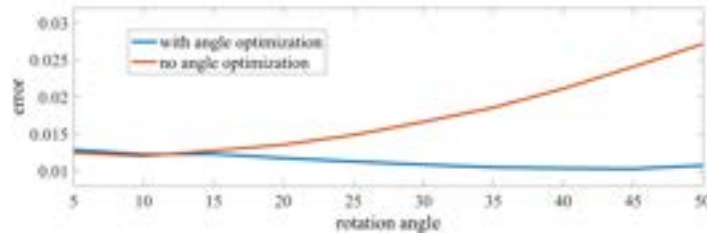


Figure 4.27: Plot of twisting accuracy when the rotation angle r is increased. The plot shows the Frobenius norm of the distance between the target elasticity tensor and the actual elasticity tensor obtained by running homogenization on the compound pattern.

sub-quads (see Figure 4.28). For example, for a rotation angle of 10 degrees, we had an initial error of 0.0123 with our method, while, with one level of refinement, the error is almost halved, decreasing to 0.0069, and two levels decrease the error to 0.0040.

Non-rhombicity error. In our approach, all cells are approximated by rhombi for the purposes of computing the cell structure. To study the effect of deviation of cell shapes from rhombic, we experimented with a simple example shown in Figure 4.29. Using a 2×2 quad grid in a compound cell, and starting from 4 perfect squares, we move the center vertex in the direction of the top-right corner by an increasing distance d in each axis, which increases the non-rhombicity of every cell. For each d , the homogenized properties of the material consisting of compound cells is computed.

We compute then, for each value of d , the *rhombic errors* of each of the 4 quads as the sum of the distances from the quad vertices to the vertices of the closest rhombus (which can be obtained as

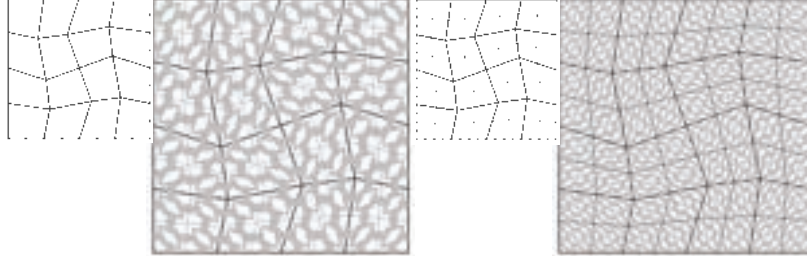


Figure 4.28: Refinement of the twisted structure. Left: one level of quad refinement. Right: two levels of quad refinement.

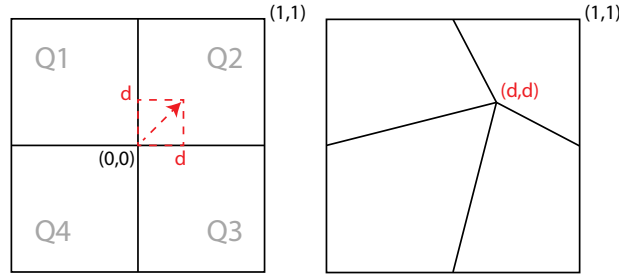


Figure 4.29: Non-rhombic cell deformation: the center vertex is slowly translated from $(0,0)$ to $(1,1)$, increasing non-rhombicity of cells and distance from the homogenized properties of the target cell to target material properties.

explained in Section 4.7). We also compute the *target error* as the Frobenius norm of the difference between the homogenized elasticity tensor for the compound cell and the target one. Results are shown in Figure 4.30. The distance to target material changes smoothly with the total rhombic error. Moreover, to analyse quantitatively the quality of our quad meshes and to show how much damage non-rhombicity could cause in accuracy, we computed the average rhombic error for our meshes in Figure 4.34, normalizing the results with the squared root of the average area of quads. The resulting numbers for pliers, ghost, disk and bar models can be seen on the right side of Figure 4.30. Notice that these numbers would indicate a low target error.

Scalability. Using the same example shown in Figure 4.29 with d fixed at 0.25, we also experimented with the scalability of our solution by running the entire pipeline (quad optimization, material optimization, material to geometry mapping and final mesh construction) on different levels of refinement on the initial quad mesh. The result is shown in Figure 4.31. Notice that our

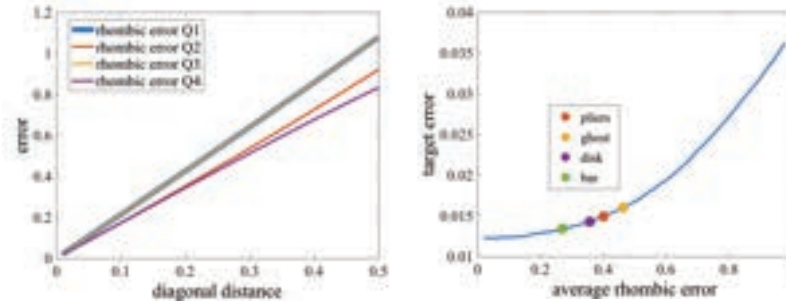


Figure 4.30: Plots of accuracy in non-rhombicity experiment. Left: plot shows how rhombic error varies when d value is increased. Notice that the change is almost linear and, as expected, error for Q1 and Q3 are the same. Right: smooth variation of target error when rhombic error increases. Dots indicate corresponding error for our example models in Figure 4.30.

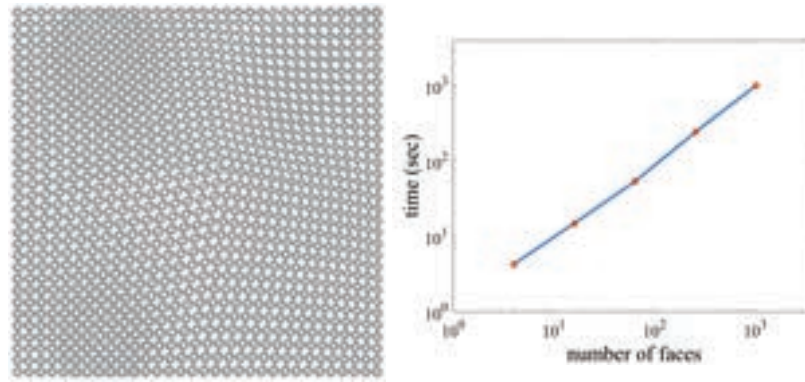


Figure 4.31: Scalability experiment. Left: example of optimized structure with 4 levels of refinement on the initial mesh, which means a final number of 1024 faces. Right: Plot (in log–log scale) showing how our running time grows with the number of quad faces. For this experiment, we used a machine with 20 cores and 32GB of RAM.

running time scales linearly with the number of cells of the quad mesh.

Merge error. Once the geometry is defined per cell, it needs to be merged into a single geometry for the whole object. For our choice of pattern (but without using our smooth spline fit), adjacent cells share a single node and the structures of two cells may have different geometric parameters (radii) assigned to the shared node: these radii need to be averaged, affecting the properties of cells. In contrast, with our microstructures family, if the transition of materials is smooth, the change in radii will also be smooth and almost no averaging will be required. In fact, for square cells ($\alpha = \pi/2$), the radius at the boundary will always be very close to 0.3 (see Figure 4.17), because of

the way we fix p_5 in our map construction, as described in Section 4.6.2.

Figure 4.32 shows an example where there is a significant mismatch between radii assigned to the shared node by two cell structures that have same material properties ($E = 0.0315$, $\nu = 0.75$ and anisotropy measurements a_1 and a_2 in the order of 10^{-5} and 10^{-4} , respectively). After the merge, the new homogenized properties can be obtained by building a new square cell as shown on the right side of Figure 4.32. The new material has significantly different properties, not being isotropic anymore ($a_1 = 0.028$ and $a_2 = 0.037$).

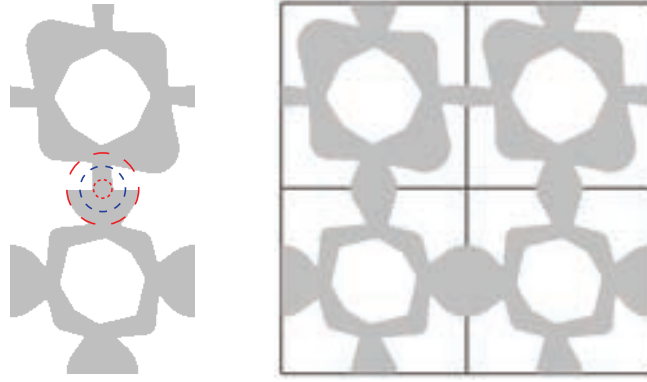


Figure 4.32: Error due to merging cell structures. Left: radii of contact nodes of two structures with same material properties (in red) and resulting radius after merge (in blue). Right: base homogenization cell to verify the properties of merged structure (which are significantly different than original).

Interpolation accuracy. Finally, we show that our spline approximation is quite accurate. Figure 4.33 shows the absolute differences $|H_V(\mathcal{P}(E_j, v_j)) - v_j|$ and $|H_E(\mathcal{P}(E_j, v_j)) - E_j|$, at points (E_j, v_j) not present in the originally sampled set: these are differences between actual homogenized material properties of the geometric structures obtained using our map \mathcal{P} , and the values at which \mathcal{P} was evaluated. Smoothness of the map \mathcal{P} is essential for the interpolation of the relatively sparse set of values to be accurate.

Examples of manufactured structures. Finally, we applied our complete pipeline for a set of 2D examples similar to [Panetta et al. 2015a], with a crucial difference being that we use irregular, boundary-conforming meshes optimized for rhombic shape of cells. The per-cell material property

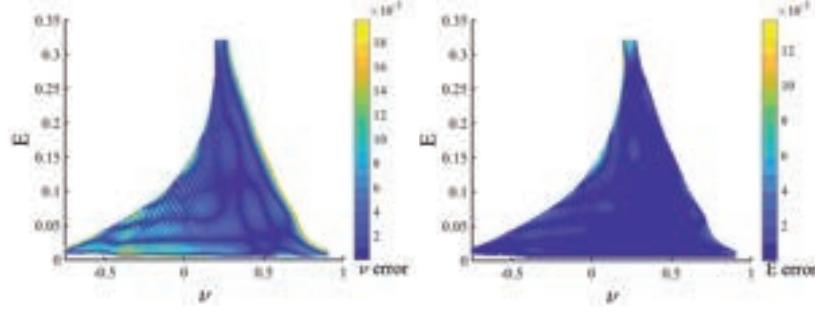


Figure 4.33: Accuracy of \mathcal{P} away from sample points, measured as the difference between directly computed and interpolated E and ν .

distribution is obtained using the optimization described in [Panetta et al. 2015a]: we prescribe fixed Dirichlet boundary conditions on a part of the boundary, and target positions of another (target) part, optimizing the material distribution to obtain the desired deformation.

As we worked with 2D elasticity, we used 0.5" sheets of closed-cell foam with small (< 0.2 mm) foam cell size to fabricate our examples using a laser cutter. We also experimented with other materials, as 1/8" acrylic sheets, which allow for higher resolution when compared to foam (see pliers example in Figure 4.34). These examples have primarily illustrative purpose. We expect more practical applications to be accomplished by integration into new CAD design pipelines e.g, nTop Platform [nTopology 2020], which already supports a range of small-scale parametric structures.

4.9 CONCLUSIONS AND LIMITATIONS

We have demonstrated that it is possible to construct a family of geometric structures for rhombic shapes with a large range of angles, parameterized directly by material parameters of these shapes. The map from the material parameters to the geometric parameters is smooth and represented in a compact form as a set of 3D splines. Within explicitly defined bounds, any structure in the family is isotropic. While the material-to-geometry map depends on the base material properties,

















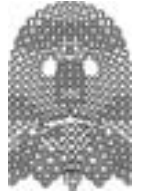



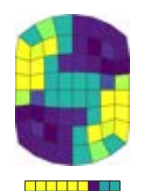



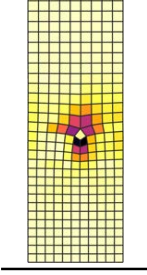
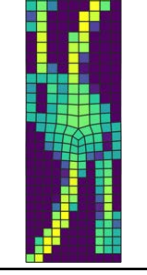
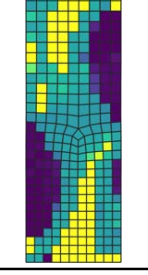
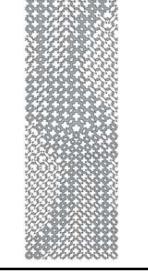


α	E	ν	pattern	simulated	fabricated (photograph)
					
					
					
					
					

Figure 4.34: Summary table of the results fabricated with our method. The plier model was laser-cut in acrylic, while the rest were laser-cut from foam. From left to right: angle per quad on the optimized mesh, optimized material distribution (Young's modulus and Poisson's ratio), final geometry at rest, deformed pattern (simulated) and photograph of deformed pattern (fabricated). Video clips of the deformation of the fabricated results are provided in the additional material. The minimum angle after optimization is 57.4 deg for the pliers, 64.5 deg for the ghost, 66.0 deg for the disk, and 67.2 deg for the bar. On these examples, our Young's moduli ranges from 0.005 to 0.25, and our Poisson's ratio ranges from -0.27 to 0.8.

the transformation between different materials can be achieved with a simple analytic formula; for this reason, the proposed family can be viewed as independent of the base material properties.

We demonstrate that using this family improves the accuracy with which target material properties can be approximated using cellular structures manufactured from a single material.

This work is a first step towards explicitly defined material-to-geometry maps of this type. We note that as an intermediate step in our construction we have obtained a map from arbitrary orthotropic, not just isotropic, properties to geometric parameters. Orthotropic materials may be useful in many contexts, and our work can be extended to this. Previous work [Ostanin et al. 2018; Milton et al. 2017] demonstrates that much broader, nearly optimal, range of parameters can be obtained by using more complex structure topology; expanding the range further is another possible future direction of work.

This work forms a foundation for developing similar parametric families in 3D, since, although requiring more parameters and computational time, we do not foresee any fundamental problems in extending the construction of cell geometry families to 3D structures. As a proof-of-concept, we produced a smooth family of microstructures for cubic cells in 3D, following exactly the process described in Section 4.6.2, with 2D homogenization replaced with 3D. We used a cell geometry with cubic symmetry, guaranteeing orthotropy with same Young's modulus in all directions, which means having a three dimensional material space ($m = 3$). Initially, 9 parameters defining the geometry were considered ($p = 9$). This number was reduced to 7 and then to 5 with parameter elimination step ($p' = 5$). Then, we used PCA for the final reduction from 5 to 3. Figure 4.35 shows the initial and also the final coverage with our 3-parameter space, from which a nonlinear map for isotropic structures can be extracted. We note that the second component of the method, decomposing shapes into hexahedral cells of suitable shape, e.g., close to rhombohedra, requires much greater adaptation, although hex meshing methods can be used as a starting point.

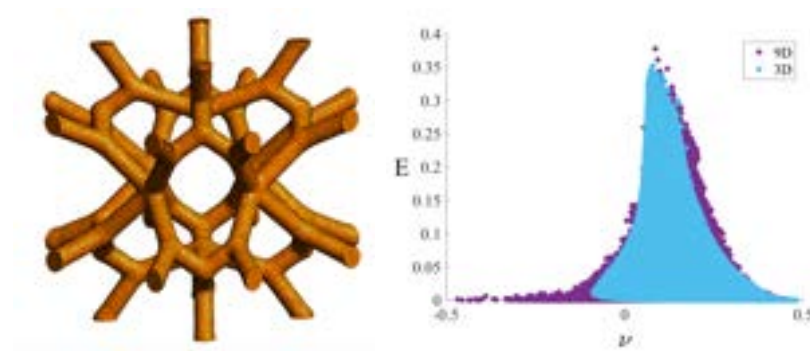


Figure 4.35: Left: example of geometry in our 3D family. Right: comparison of the material properties obtained with our initial 9 parameters and the reduced set of only 3.

4.10 CHAPTER NOTES

4.10.1 MICROSTRUCTURE GEOMETRY REPRESENTATION

In this appendix we summarize how the geometry of the microstructure is computed from the geometry parameters p_i . We use the approach of [Panetta et al. 2017a] in 2D.

Inflation Graph. Microstructure is described by a graph of edges, with positions and radii and blending ratios associated with vertices. The geometry itself is a level set of a signed distance function constructed by combining distance functions associated with a primitive assigned to each edge.

Let $G = (\mathcal{V}, \mathcal{E})$ be the graph, where $\mathcal{V} = \{v_i\}_{i=1}^n$ represents the vertex indices and $\mathcal{E} = \{e_j\}_{j=1}^m$ are the edges. Recall that $F : Q \rightarrow R$ is the map from a unit square to the rhombus of interest. The shape parameters are $\{\mathbf{q}_i, r_i, b_i\}_i$ where:

- $\mathbf{q}_i = F\tilde{\mathbf{q}}_i$ are the position of the vertices of G in R , and $\tilde{\mathbf{q}}$ are positions in Q ,
- r_i are per-vertex radii,
- b_i are per-vertex blending parameters.

In the family described in this paper, there are 12 nodes, but due to diagonal symmetries, only 4 have independent parameters, these can be chosen as in Figure 4.3. Moreover, symmetries also require nodes 3 and 4 to stay on diagonals (i.e., their coordinates have equal in magnitude values), and for node 1 to remain fixed. In our family we also do not vary the blending parameters keeping them at value 0.01.

Note that the positions $\{\tilde{\mathbf{q}}_i\}_i$ correspond to the reference unit square, while $\{\mathbf{q}_i\}_i$ correspond to the rhombus.

Edge Geometry. A disk of radius r_i is placed at each vertex; then two outer common tangents are constructed for the disks centered at vertices v_i and v_j at the endpoints of an edge.

We define a signed distance function for each primitive described above, associated with an edge, and combine them together for a final shape.

Joints Geometry. The next step in the construction of the SDF for the graph G , is to determine the smooth blended geometry at a joint. A joint is defined as a pair $(v_i, \{e_k | k = 1 \dots N_i\})$, with a central vertex v_i and the list of its incident edges e_k .

Given the SDF distance d_k from query point \mathbf{x} to each incident edge primitive (as defined above), the SDF to the blended joint is defined as a smooth minimum over the y_k with a blending parameter ρ_i :

$$KS(d_1, \dots, d_{N_i}, \rho_i) = -\frac{1}{\rho_i} \ln \left(\sum_k \exp(-\rho_i d_k) \right) \quad (4.17)$$

The blending parameter ρ_i is obtained as by multiplying the vertex blending parameter b_i with two spatially-varying modulation terms. One term depends on the convex hull of the joint, to avoid bulging. Another term depends on the minimum incident angle of the joint (between each

pair of incident edges). The idea of the latter is to smooth more joints with small angles. [Panetta et al. 2017a] explains how these quantities are computed.

Combining the Joints. The last element to consider when building the SDF for the graph as a whole, is to avoid sharp creases that can arise when combining joints together. Here, the idea is to blend the SDF of the two closest joints using the smooth-min function Equation (4.17), with a blending factor that depends on the difference between the “hard-union” distance and the “smooth” SDF to those joints. More details can be found in [Panetta et al. 2017a].

4.10.2 SWEEP PARAMETERS

Displacements. In the initial 8-dimensional sweep, Node 2 displacement parameters (p_1, p_2) vary in the range $(0.1, 0.5)$, as $p_1 + p_2 > 0.5$ would take the node out of the triangle. Node 3 and node 4 displacements p_3 , and p_4 vary in the range $(0.1, 0.9)$, to span most of the allowed range (we avoid placing nodes too close to the boundary as this is likely to result in difficult-to-manufacture structures). We use 5 samples for all 4 offset parameters.

Radii. All radii p_5, \dots, p_8 are allowed to vary in the $0.05 \dots 0.3$ range, and the number of samples was 6 for p_5, p_7, p_8 , and 4 for p_6 . The upper bound was chosen somewhat lower than the values needed to produce a solid cell, as we focused on the range of Young’s moduli $0.01 \dots 0.3$ relative to the base material, and using radii about 0.3 produces E close to 0.33.

5 | A PARAMETRIC CLASS OF COMPOSITES WITH A LARGE ACHIEVABLE RANGE OF EFFECTIVE ELASTIC PROPERTIES

In order to successfully produce objects using microstructures, it is important to have geometries that can reach a large range of elasticity responses. In this context, this final chapter, based on a published work in collaboration with Igor Ostanin, George Ovchinnikov, and Denis Zorin [[Ostanin et al. 2018](#)], explores how close one can get to the theoretical limits in 2D with physically realizable structures. We present two families of hexagonal microstructures that are able to achieve effective material properties covering most of the theoretical possible range for isotropic metamaterials. In this project, I was responsible for designing, running, and analyzing homogenization sweeps to identify the relationship between geometric parameters and material properties for different families of microstructures. I also performed shape optimization experiments targeting extreme material properties in order to perform a shape derivative study and show that our structures are close to local optima.

We study an instance of the *G-closure* problem for two-dimensional periodic metamaterials. Specifically, we consider composites with isotropic homogenized elasticity tensor, obtained as a

mixture of two isotropic materials. We focus on the case when one material has zero stiffness i.e., single-material structures with voids. This problem is important, in particular, in the context of designing small-scale structures for metamaterials that can be manufactured using additive fabrication. A range of effective metamaterial properties can be obtained this way using a single base material.

We demonstrate that two closely related simple parametric families based on the structure proposed by O. Sigmund in [Sigmund 2000] attain good coverage of the space of isotropic properties satisfying Hashin-Shtrikman bounds. In particular, for positive Poisson's ratio, we demonstrate that Hashin-Shtrikman bound can be approximated arbitrarily well, within limits imposed by numerical approximation: a strong evidence that these bounds are achievable in this case. For negative Poisson's ratios, we numerically obtain a bound which we hypothesize to be close to optimal, at least for metamaterials with rotational symmetries of a regular triangle tiling.

5.1 INTRODUCTION

The relationship between the geometric structure of a periodic composite material and its effective properties (elastic properties in particular) is a central question in mechanics of composites. The direct problem – finding the effective elastic properties from the known periodic structure of the composite material – is solved by means of homogenization theory. The inverse problem (finding a periodic structure yielding specific material properties) is more challenging: in general, it requires solving a non-convex non-linear PDE constrained optimization problem with highly nonunique solution (inverse homogenization). Most importantly, it is not known in which cases this problem has feasible solutions, i.e., for which target elasticity tensors a corresponding composite structure with a given base material exists. This problem is known as the G-closure problem [Allaire 2002b]. While for the heat conduction explicit solutions are known in some cases, for elasticity this is a long-standing problem. The outer bounds of the feasible region in the space of elasticity tensors

are given by the well-known Hashin-Shtrikman bounds [Hashin and Shtrikman 1963] on the bulk and shear moduli, as well as their refinement [Cherkaev and Gibiansky 1993]. For positive Poisson ratios, the answer is provided in the recent work [Milton et al. 2017]. The right Hashin-Shtrikman bounds are achieved by sequential laminates, *i.e.*, assuming infinite geometric complexity and separation of scales (*i.e.*, very large differences in periods of laminations in sequence). In general, these materials cannot be practically realized, although one can manufacture, with considerable difficulty, finite-resolution approximations.

We explore solutions to this problem numerically, using specific simple parametric families of structures inspired by several previous papers. We show that with modest geometric complexity, and no separation of scales, one can approach the boundaries closely. Thus, a large fraction of our structures can be, in principle, manufactured and can serve as a starting point for further simplification. Our experiments provide an estimate on the topological complexity required to achieve a particular percentage of coverage of the area in the space of elastic properties defined by Hashin-Shtrikman bounds.

Contributions. We explored two families of structures (depicted in Figure 5.1(A, B)), using a hexagonal base cell, and symmetric with respect to rotation of the cell by $n\pi/3$, which ensures that their effective elastic tensors are perfectly isotropic. Each structure has only four parameters. We observe that these structures have the following properties:

- For positive effective Poisson’s ratio and volume fractions approaching one, the bounds of the Hashin-Shtrikman region are approached arbitrarily closely, within limits of numerical accuracy.
- For negative effective Poisson’s ratio and volume fractions approaching one, chirality parameter allows us to cover a larger fraction of the region defined by the Hashin-Shtrikman bounds than all previously known.

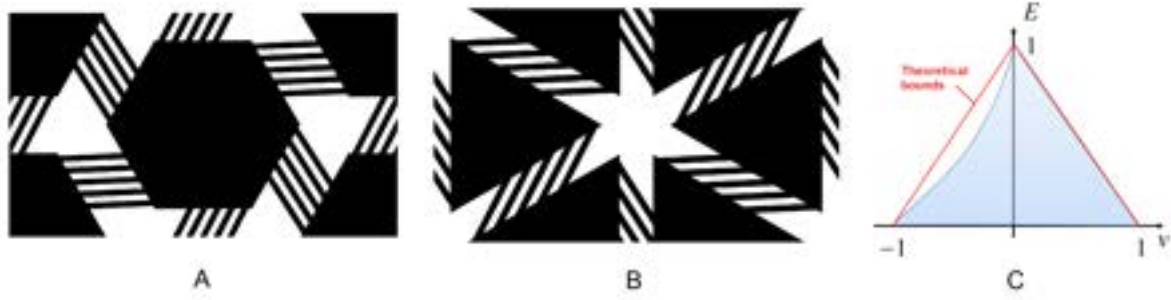


Figure 5.1: (A,B) parametric structure families, (C) Domains of achievable effective elastic properties and Hashin-Shtrikman theoretical bounds ($E_b = 1$, $\nu_b = 0$) in (E, ν) domain.

- The low number of parameters of the structure allows for simple mapping of material parameters to structure parameters, potentially avoiding inverse homogenization entirely.
- We show that for volume fractions different from one and positive effective Poisson's ratios, proposed structures are also close to the Hashin-Shtrikman bounds.
- For negative effective Poisson's ratios, we show evidence that our structures are close to locally optimal (i.e., increasing the number of structure parameters is not likely to lead to improvements)
- Our family of structures in 2D covers a larger fraction of the Hashin-Shtrikman domain than all previously known families.

Figure 5.1(C) shows the set of Young's moduli and Poisson's ratios that could be achieved by tuning the parameters of structures (A) and (B), as related to the known theoretical bounds on these moduli (see the discussion in the next section).

Remark. To illustrate material property coverage, in most of our figures we use Young's modulus – Poisson's ratio coordinates, vs. bulk-shear modulus more commonly found in the literature: the distance in this space captures the differences in material behavior in a more intuitive way. We

discuss the relationship between these two parameterizations of material properties below.

5.2 BACKGROUND AND RELATED WORK

Bounds on elastic properties of isotropic materials. The elastic tensor characterizing properties of a periodic composite material is obtained by homogenization over a base cell of the composite structure (Section 5.4). In this work, we consider composite structures that are isotropic due to spatial symmetries of the base cell. Such composites can be described by two independent elastic moduli (bulk and shear moduli or Young's modulus and Poisson's ratio). The best known bounds for composite properties are Hashin-Shtrikman bounds [Hashin and Shtrikman 1963]. In case of well ordered strong (κ_s, μ_s) and weak (κ_w, μ_w) phases $((\kappa_s^{HS} - \kappa_w^{HS})(\mu_s^{HS} - \mu_w^{HS}) > 0)$, the HS bounds are:

$$\begin{aligned} \kappa_l^{HS} &< \kappa < \kappa_u^{HS}, & \mu_l^{HS} &< \mu < \mu_u^{HS}, \\ \kappa_u^{HS} &= \kappa_s + \frac{1-\phi}{\frac{1}{\kappa_w - \kappa_s} + \frac{\phi}{\kappa_s + \mu_s}}, & \kappa_l^{HS} &= \kappa_w + \frac{\phi}{\frac{1}{\kappa_s - \kappa_w} + \frac{(1-\phi)}{\kappa_w + \mu_w}}, \\ \mu_u^{HS} &= \mu_s + \frac{1-\phi}{\frac{1}{\mu_w - \mu_s} + \frac{\phi(\kappa_s + 2\mu_s)}{2\mu_s(\kappa_s + \mu_s)}}, & \mu_l^{HS} &= \mu_w + \frac{\phi}{\frac{1}{\mu_s - \mu_w} + \frac{(1-\phi)(\kappa_w + 2\mu_w)}{2\mu_w(\kappa_w + \mu_w)}}. \end{aligned}$$

Figure 5.2 shows Hashin-Shtrikman bounds for void-material (A,B) and bimaterial (C,D) composites, in terms of bulk and shear moduli (A,C) and Young's modulus and Poisson's ratio (B,D) for arbitrary volume fractions $0 \dots 1$ and for several fixed volume fractions. These bounds are known not to be optimal the general case: Cherkaev-Gibiansky bounds are substantially tighter [Cherkaev and Gibiansky 1993]. However, in the case when the weak phase has zero elastic tensor, these bounds exactly coincide with the Hashin-Shtrikman bounds. In terms of volume fraction ϕ , and bulk and shear moduli of the base material equal to κ_b and μ_b respectively, the bounds on effective bulk

moduli κ and μ in this case reduce to:

$$0 < \kappa < \kappa_u^{HS}, \quad 0 < \mu < \mu_u^{HS},$$

$$\kappa_u^{HS} = \kappa_b + \frac{1 - \phi}{-\frac{1}{\kappa_b} + \frac{\phi}{\kappa_b + \mu_b}}, \quad \mu_u^{HS} = \mu_b + \frac{1 - \phi}{-\frac{1}{\mu_b} + \frac{\phi(\kappa_b + 2\mu_b)}{2\mu_b(\kappa_b + \mu_b)}}. \quad (5.1)$$

Expressing bulk and shear moduli as $\kappa = \frac{E}{2(1-\nu)}$, $\mu = \frac{E}{2(1+\nu)}$, one obtains the isotropic bounds in terms of Young's modulus and Poisson's ratio, forming a triangular region:

$$0 < E(\nu) < -2C_1\nu + 2C_1, \quad 0 < E(\nu) < 2C_2\nu + 2C_2,$$

$$C_1 = \frac{E_b}{2(1-\nu_b)} + \frac{E_b(1-\phi)}{\phi(1-\nu_b^2) - 2(1-\nu_b)},$$

$$C_2 = \frac{E_b}{2(1+\nu_b)} + \frac{2E_b(1-\phi)}{\phi(3-\nu_b)(1+\nu_b) - 4(1+\nu_b)}. \quad (5.2)$$

where E_b and ν_b are the Young's modulus and Poisson's ratio of the base material. For the case of $\phi = 1$, expressions (5.1,5.2) are reduced to

$$0 < \kappa < \kappa_b, \quad 0 < \mu < \mu_b,$$

$$E(\nu) < -\frac{E_b}{1-\nu_b}\nu + \frac{E_b}{1-\nu_b}, \quad E(\nu) < \frac{E_b}{1+\nu_b}\nu + \frac{E_b}{1+\nu_b}. \quad (5.3)$$

Note that the rational transformation between (κ, μ) and (E, ν) coordinates is not globally one-to-one: in the extreme case of vanishing Young's modulus, both κ and μ go to zero, so the lower side of the triangle in (E, ν) domain collapses to a single point $\kappa = 0, \mu = 0$, except points $\nu = \pm 1$. These two points in (E, ν) coordinates correspond to the lines $\kappa = 0$ and $\mu = 0$ in (κ, μ) coordinates. This in part explains why we view (E, ν) coordinates as more intuitive: for low E , there is a substantial

difference in measurable behavior between, *e.g.*, materials with $\nu = -1$, and $\nu = 1$, while both κ and μ are close to zero.

G-closure. The problem of G-closure was extensively studied. It is known to have a solution for thermal properties; however the problem remains unsolved for elasticity. Related theory is covered in several books [Allaire 2002b; Torquato 2002b; Milton 2002a; Cherkaev 2012]. The first theoretical example of a composite attaining maximum bulk modulus, a random assemblage of coated spheres (Figure 5.3(A)) – was identified by Hashin [Hashin 1962]. Another type of structures attaining extremal bulk modulus without separation of scales was found by Vigdergauz for anisotropic [Vigdergauz 1989] and later isotropic (Figure 5.3(B)) case [Vigdergauz 1999]. Several authors [Lurie and Cherkaev 1985; Norris 1985; Francfort and Murat 1986; Milton 1986] demonstrated that sequential laminates can achieve extreme bulk and shear modulus simultaneously (Figure 5.3(C)). Milton has shown [Milton 1992] that sequential laminates can demonstrate negative Poisson’s ratio close to -1 (Figure 5.3(D)). Milton and Cherkaev [Milton and Cherkaev 1995] have demonstrated the attainability of G-closure for infinitely rigid and void phases. They describe an approach for constructing composites with any given tensor using elementary structures as building blocks (an example of such structure is given in Figure 5.3(E)). For the same purpose Sigmund adopted laminated regions, and suggested [Sigmund 2000] earlier unknown class of extreme isotropic composites (Figure 5.3(G)), that display high bulk modulus while maintaining low shear modulus. Our approach is based on this work.

An important related direction is the design of materials with negative Poisson’s ratio (also known as *auxetic*, or *dilational*). Auxetic materials were first described in the work of Lakes [Lakes 1987], and developed in many papers (see a recent review [Lakes 2017]). Many different types of auxetic materials were identified: beyond already named sequential laminates [Milton 1992] and unimode structures [Milton and Cherkaev 1995] presented in Figures 5.3(D,E), there are structures based on rigid rotating units, connected with hinges (Figures 5.3(I,F)) [Grima and Evans 2000; Grima

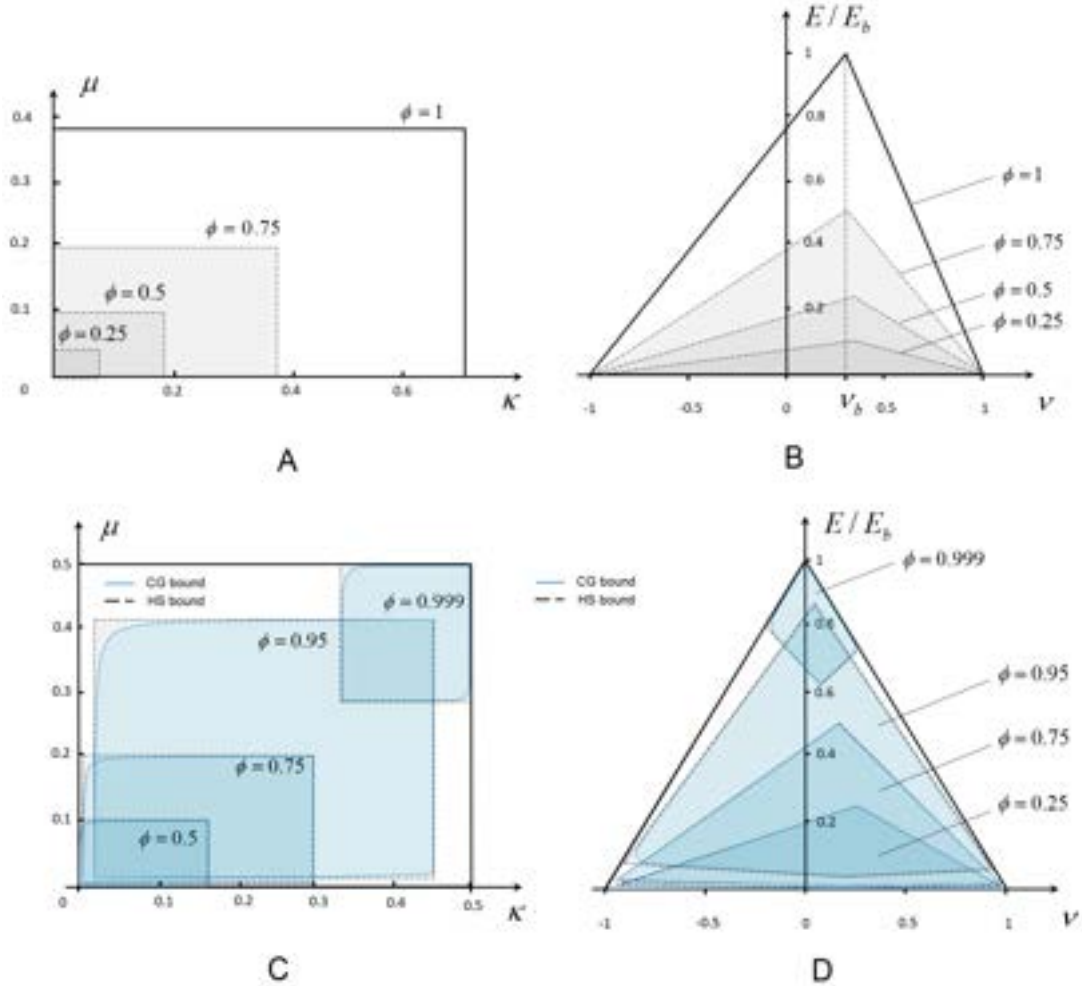


Figure 5.2: (A, B) Hashin-Shtrikman bounds for the elastic moduli of isotropic void-material composites: (A) in bulk and shear modulus coordinates and (B) the same bounds transformed to Poisson's ratio and Young's modulus coordinates. (C,D) Hashin-Shtrikman (grey) and Cherkhaev-Gibiansky (blue) bounds for elastic moduli of isotropic bimaterial composite ($\nu_s = \nu_w = 0$, $E_w = 0.001E_s$) (C) in terms of bulk and shear moduli and (D) in terms of Poisson's ratio and Young's modulus. For brevity, here and below Hashin-Shtrikman bounds are labeled on images as HS bounds and Cherkhaev-Gibiansky bounds – as CG bounds.

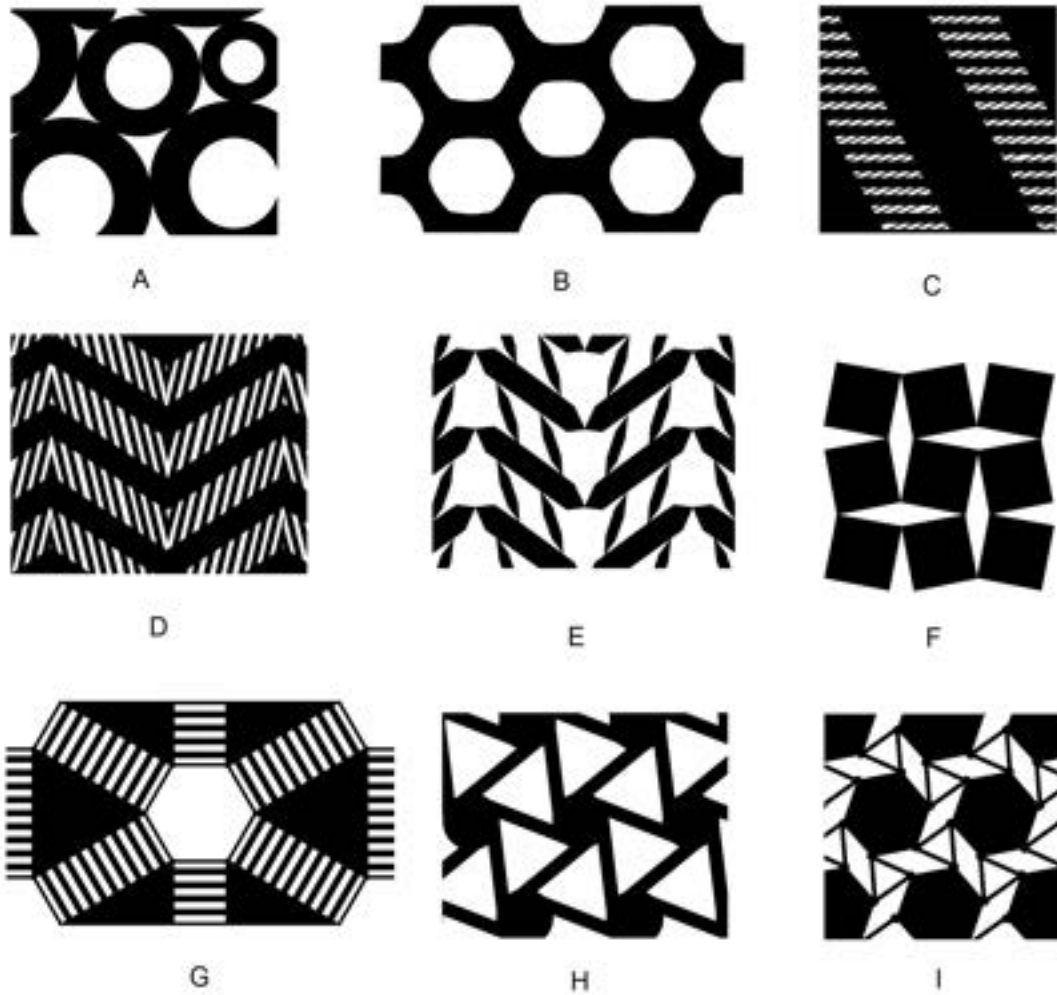


Figure 5.3: Known extremal composites (A) Coated spheres assemblage, exhibiting extremal bulk modulus, (B) Vigdergauz structure, achieving maximum bulk modulus (C) rank 3 sequential laminate, exhibiting maximum bulk and shear modulus (D) “Herringbone” laminate structure providing Poisson’s ratio of -1 and (E) and its unimode analog. (F) Auxetic material made of rotating squares [Grima et al. 2005] (G) Sigmund extremal structure, providing maximum bulk and minimum shear modulus (H) Chiral isotropic auxetic material [Prall and Lakes 1996] and (I) similar structure, suggested in [H et al. 2011]

et al. 2005; H et al. 2011; Milton 2015], and honeycomb chiral auxetic structures(Figures 5.3(H)) [Prall and Lakes 1996]. We use the latter idea in our current work.

In their recent work [Milton et al. 2017], Milton *et al.* identify optimal bounds and corresponding microgeometries for both 2D and 3D anisotropic composites. This work demonstrates that the *right* Hashin-Shtrikmann bound for any volume fraction is achieved by a finite-rank laminate composite. In our work, we demonstrate that the right bound for isotropic 2D composites can be approached arbitrarily closely for any volume fraction by a family of realizable structures, approaching a two-scale composite in the limit, a subclass of which has been earlier found and studied by Sigmund [Sigmund 2000].

Topology optimization was used for the purpose of extremal material design (see [Xia and Breitkopf 2015] for an overview). New periodic structure designs were obtained in [Sigmund 1994; Neves et al. 2000; X. Huang 2011]. However, the problem of finding a structure with specific properties is highly nonconvex; as a result, the optimization often fails to reach the target properties. Use of filtering techniques for suppression of checkerboard effects makes it difficult to obtain complex-topology designs which appear to be necessary for target parameters close to the theoretical bounds. Shape optimization is often more successful at achieving specific target material properties, but topology preservation means that initial design topology needs to be obtained by other means.

Wide use of additive fabrication lead to renewed interest in using small-scale structure to achieve specific material behaviors: on the one hand, additive technology makes fabrication of these complex structures possible; on the other hand, these structures allow to manufacture strong parts with lower weight, or objects with continuously variable material properties, *e.g.*, for manufacturing prosthetic devices, or for “soft” robotics. In this context, a number of additional practical issues become of importance – printability of the structure, the absence of extreme stress concentrations that lead to structure damage, its stability towards unwanted nonlinear

behavior. Panetta and co-authors [Panetta et al. 2015b, 2017b] presented a framework for the structural design based on ground state search for topology with subsequent low-parametric shape optimization to achieve the desirable elasticity tensor while satisfying a set of additional constraints (*e.g.* constraint on maximum von Mises stress).

In this work, rather than using topology or shape optimization to solve the problem of G-closure numerically, we opt for an intermediate approach: we integrate features of several proposed structures into two parametric families with a small number of parameters. As a result, the material property space coverage of these structures can be explored in brute-force way by parameter sweeps. This achieves two goals: first, this allows us to find an inner bound on the G-closure domain (with Hashin-Shtrikman bounds providing the outer domain). Second, we can straightforwardly tabulate the necessary structures parameters for any desirable effective elastic properties.

5.3 PERIODIC STRUCTURE FAMILIES

Our families are based on synthesis of hexagonal/triangular isotropic structures invented by Sigmund [Sigmund 2000] (Figure 5.3(C)), and the chiral structures proposed in [Prall and Lakes 1996; H et al. 2011; Milton 2015] (Figure 5.3(H,I)) for negative Poisson’s ratio.

Figure 5.4(A,B) shows the proposed new periodic structures. The starting point is the Triangle and Hexagon structures presented in [Sigmund 2000]. These structures consist of triangular or hexagonal solid areas, connected by rectangles partitioned into separate beams (we refer to these rectangles as laminate areas). We extend these structures in a very simple way, by allowing the beams of laminate region to have arbitrary orientation – this considerably extends the range of materials represented by the structure, providing auxetic behavior via the mechanism described in [Prall and Lakes 1996]. Structures of this type are described by four parameters shown in the

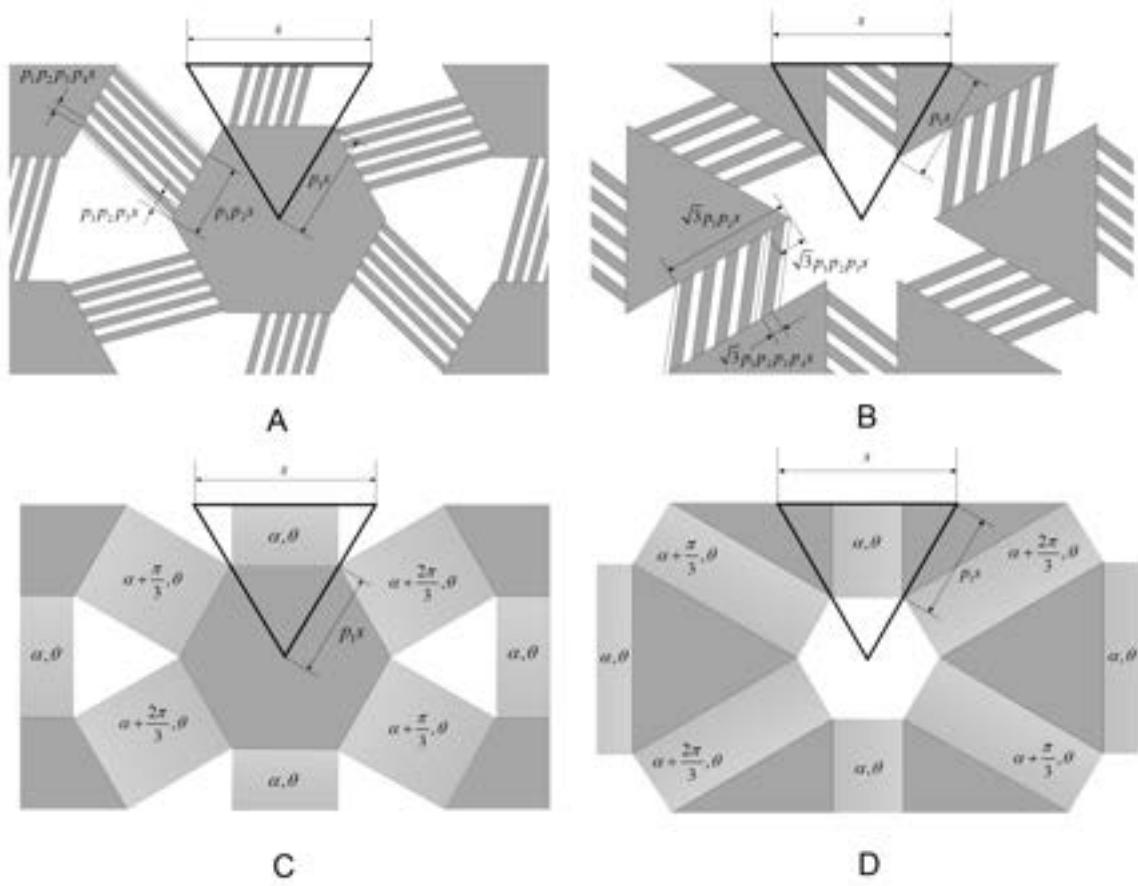


Figure 5.4: (A,B) Structure families: (A) Triangle, (B) Hexagon. (C,D) Limit structures: (C) Triangle, (D) Hexagon

figure (size of the solid region p_1 , width of the laminate area p_2 , period of the beams in the laminate area p_3 , and width of a beam p_4 , with each dimension measured relative to the previous (*e.g.*, p_2 is a fraction of p_1 , in the range $0 \dots 1$). It is important to note that all parameters are *continuous*, including the one determining the number of beams p_3 ; if it is not of the form $1/n$, where n is an integer, the resulting structure has thinner “remainder” beams, which we place on the sides of the laminate area. In this way, all parameters can be changed continuously, and optimized more easily.

We also consider a limit case of these structures, to provide a validation of the hypothesis that with the number of beams in the laminate areas going to infinity (*i.e.*, the resulting structure becomes two-scale), the material properties approach the boundary of the reachable domain (Figure 5.4(C)); these structures are parametrized in a similar way, but have only three parameters, as the beam period approaches zero. Parameter p_1 is defined in the same way as for finite structures, and the two remaining parameters θ and α (shown in Figure 5.5(B) below), define the volume fraction and the orientation angle of the laminated areas. Parameters α and θ are used to set the elastic tensor in the laminated region.

5.4 METHODS

In order to perform the parametric studies of the new periodic structures, we use a FEM-based, regular-grid homogenization code, extending [Xia and Breitkopf 2015]. Our algorithm of periodic homogenization follows the method of mutual energies, as described in [Xia and Breitkopf 2015]. The homogenized tensor of elasticity of the rectangular cell $Y \in \mathbb{R}^2$ is found as:

$$C_{ijkl}^H = \frac{1}{|Y|} \int_Y C_{pqrs}(y) \varepsilon_{pq}^{Aij}(y) \varepsilon_{rs}^{Akl}(y) dY. \quad (5.4)$$

Here $|Y|$ is the area of the cell, $C_{pqrs}(y)$ is the tensor of elasticity of the material at the point y inside the cell domain, and $\varepsilon_{pq}^{A_{ij}}(y)$ are the strain fields induced by the imposition of three constant macroscopic unit test strains A_{ij} . These strains are found by solving three ($pq = 11, 22, 12$) cell problems:

$$\begin{aligned} \left(C_{ijkl}(y) \varepsilon_{kl}^{A_{pq}}(y) \right)_{,j} &= 0, \\ \varepsilon_{kl}^{A_{pq}}(y) &= w_{kl}(y) + A_{kl}, \\ w_{kl}(y) &\text{is Y-periodic.} \end{aligned} \quad (5.5)$$

Constant unit test strains A_{pq} are imposed on the domain boundary in the form of periodic boundary conditions on displacements u_i^{k+}, u_i^{k-} in the elements nodes (Figure 5.5(A)):

$$u_i^{k+} - u_i^{k-} = A_{ij}(y_j^{k+} - y_j^{k-}) = A_{ij} \Delta y_j^k. \quad (5.6)$$

Here $k = 1, 2$ stands for the domain boundaries perpendicular to the k -th coordinate axis. The homogenized elasticity tensor is found as follows:

$$C_{ijkl}^H = \frac{1}{|Y|} \sum_{e=1}^N \mathbf{u}_e^{A_{ij}} \mathbf{k}_e(\mathbf{u}_e^{A_{kl}}), \quad (5.7)$$

where index e stands for the element, \mathbf{k}_e is the element stiffness matrix, and $\mathbf{u}_e^{A_{ij}}$ is the vector of displacements at the nodes of e -th element upon imposition of A_{ij} -th test strain.

The computational domain is the rectangle comprising $N = N_1 \times N_2$ 4-node isoparametric finite elements (Figure 5.5(A)). For every element, we specify its density ρ and the type and parameters of its stiffness matrix \mathbf{k}_e . Finite element stiffness matrix $\mathbf{k}_e(D)$ is computed based on the material

elasticity tensor in matrix notation D , which is the elasticity tensor.

For finite structures, we use isotropic elasticity tensor.

$$D(E, \nu) = \frac{\rho}{1 - \nu^2} \begin{pmatrix} E & \nu E & 0 \\ \nu E & E & 0 \\ 0 & 0 & \frac{E}{2}(1 - \nu) \end{pmatrix}, \quad (5.8)$$

where E and ν are the Young's modulus of either the strong or weak material (the void is approximated with a very weak material).

To explore the limit case as the genus of the structure goes to infinity, as we increase the number of parallel beams in a particular region, we use the anisotropic rank-one laminate tensor [Allaire 2002b].

$$D(\alpha, \theta, E_m, E) = \rho T(\alpha) L(\theta, E_m, E) T^T(\alpha), \quad \text{where}$$

$$L(\theta, E_m, E) = \begin{pmatrix} (1 - \theta)E_m + \theta E & 0 & 0 \\ 0 & \frac{1}{\frac{\theta}{E_m} + \frac{1-\theta}{E}} & 0 \\ 0 & 0 & \frac{1}{2(\frac{\theta}{E_m} + \frac{1-\theta}{E})} \end{pmatrix},$$

$$T(\alpha) = \begin{pmatrix} \cos^2 \alpha & \sin^2 \alpha & -2 \sin \alpha \cos \alpha \\ \sin^2 \alpha & \cos^2 \alpha & 2 \sin \alpha \cos \alpha \\ \sin \alpha \cos \alpha & -\sin \alpha \cos \alpha & \cos^2 \alpha - \sin^2 \alpha \end{pmatrix}. \quad (5.9)$$

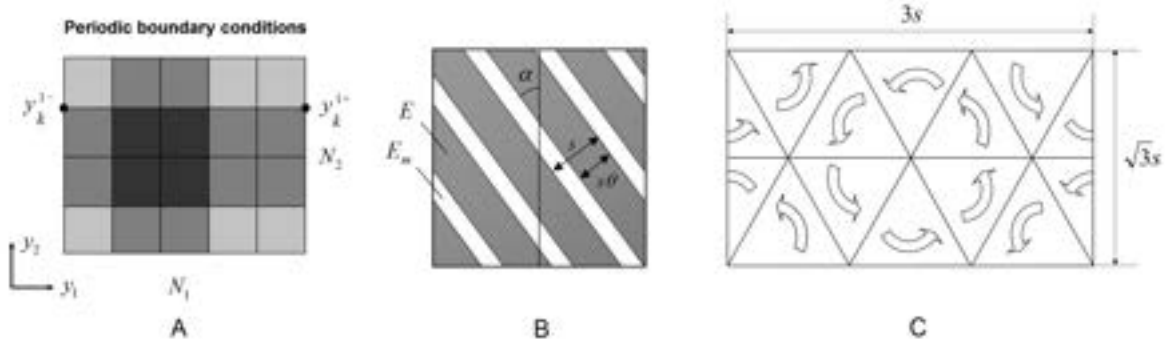


Figure 5.5: Diagrams of (A) a cell computational domain (B) rank-1 laminate (C) prescribed rotational symmetries of a computational domain.

In this equation, α is a lamination direction, E_m is the Young's modulus of the laminate's soft material (in void-material simulations $E_m = 10^{-6} \cdot E$), θ is the volume fraction of the laminate's strong phase(Figure 5.5(B)).

We ensure that the effective elasticity tensor obtained by homogenization is isotropic, by using hexagonal/triangular cells, and imposing symmetries of structures with respect to $\pi/3$ rotations, as any elasticity tensor invariant with respect to these symmetries is isotropic [Love 1944]. It is important to note that reflectional symmetries are *not* required for isotropy. This makes it possible for us to use *chiral* structures to create isotropic materials.

We use a rectangular computational domain with the ratio of the sides $3 : \sqrt{3}$ (Figure 5.5(C)). This rectangular base cell has the area of two hexagonal base cells.

Each square element overlapping the structure is assigned elasticity tensor of the base material (or laminate, if this part is marked as laminate area). Our drawing system uses geometric primitives (polygons) to define the structure that are rendered with anti-aliasing to obtain gray-scale density values in 0 to 1 range on a regular grid, that are used to scale the elasticity tensor.

The net effect of replacing the precisely defined structure with its anti-aliased rasterization is

hard to estimate precisely, although it can be expected to improve accuracy vs. using a binary density. Experimentally, we have observed that this discretized family of structures has Poisson ratio range for a given Young modulus, to the family of obtained by conforming triangulation of the precise structure, as discussed in more detail in Appendix A. The regular grid approximation is significantly faster to evaluate.

For the verification purposes, our computations were also checked with an alternative homogenization tool [Panetta et al. 2015b], which is based on representing the shape as an implicit function, and meshing the domains with an unstructured triangular mesh. These two approaches yield close results – see Appendix A for details.

Our homogenization and structure generation codes are available at https://bitbucket.org/iostanin/pco_toolbox_matlab.

5.5 NUMERICAL STUDIES

We explore the coverage of the families of structures described above using homogenization under several scenarios, as well as provide evidence, using shape optimization, that the boundary we obtain is likely to be close to optimal. For simplicity and ease of understanding, we use a base material with Poisson’s ratio $\nu_b = 0$. However, our results can be straightforwardly generalized to any Poisson’s ratio, thanks to the CLM theorem [Cherkaev et al. 1992]. The specific value of E_b simply sets the scale, as we use linear elasticity, so in all plots we use the ratio E/E_b for the effective Young’s modulus E .

We define the *coverage* of a family of structures as the set of points in effective material properties space that can be obtained by varying structure parameters. In order to quantify coverage, we introduce coverage coefficients Δ_R, Δ_L , defined as ratios of the highest (lowest) achievable Poisson’s ratio at $E/E_b = 0.5$, and the corresponding theoretical limit value of Poisson’s ratio at

$E/E_b = 0.5, \phi = 1$ (in case of $\nu_b = 0$, these limit values are $\nu_R = 0.5$, and $\nu_L = -0.5$).

Figure 5.6(A, B) shows the coverage we were able to obtain with two structure families (Triangle and Hexagon). Figure 5.6(A) provides the areas covered by two structures, whereas Figure 5.6(B) shows raw datasets, as well as the analytical conjectured bounds based on these datasets (see the discussion below). For comparison, we also sketch the approximate positions of several known extremal composites, shown in Figure 5.3. Figure 5.6(C) gives an idea how the coverage depends on the number of beams $n = 1/p_3$ for single scale structures.

Relatively few papers attempted to design families of structures with a large coverage area. The best coverage known to us was obtained in [Panetta et al. 2015b, 2017b] in 3D. For comparison, we applied the same approach to two-dimensional structures with square symmetries, to estimate coverage that can be compared to the families of structures considered in this paper.

More specifically, a ground structure was obtained by subdividing a square into smaller squares and triangulating each of these. These ground structures are parametrized by node sizes and displacements. Initially, a combinatorial search is performed by removing edges from the ground structure (while maintaining the square symmetries) and generating a set of initial structures that are estimated to yield the largest range of material properties. Then the parameters of each identified structure are optimized multiple times, to minimize the deviation of the homogenized material properties from a set of target pairs (E_i, ν_i) in the admissible region. This optimization is performed by solving the adjoint equations of elasticity at every step to compute the shape derivative. The topology of each structure remains fixed during the optimization.

Figure 5.6(D) gives the comparison of the performance of our new structures with the ones found previously by ground-state combinatorial search with subsequent shape optimization which provided the widest coverage known so far. We use coverage data for two structures, shown in the same figure, that cover essentially the whole domain covered by any structure obtained by edge

removal. We can see that our new Triangle and Hexagon structures ($n = 30$) significantly improve the coverage of possible elastic properties of composites for the same base material's elastic moduli. Fig 6(D) also provides the coverage of effective elastic properties achievable with the isotropic laminates described in [Milton 1992] (yellow line). As one can see, in case of vanishing weak phase and negative Poisson's ratio, these laminates appear to be much more compliant than our microgeometries.

Below we take a closer look at the obtained numerical results.

Optimality of Hashin-Shtrikman bounds for positive Poisson's ratios.

Our results provide compelling numerical evidence that the right Hashin-Shtrikman bound for non-fixed volume fraction can be approached with good precision with Hexagon structures as the number of beams n approaches infinity ($n = 1/p_3$). This is confirmed using simulation with limit laminate structures, which approach the boundary very closely. Figure 5.7(A) demonstrates the increase of coverage with increase of n , figure 5.7(B) demonstrates the limit case of the structures with $\phi \rightarrow 1, n \rightarrow \infty$, achieving right Hashin-Shtrikman bound. Figure 5.7(C) indicates that limit laminate structures also remain very close to Hashin-Shtrikman bound for volume fractions smaller than 1 (see also Figure 5.10). These results are consistent with the ones obtained previously in [Sigmund 2000] and recent work [Milton et al. 2017], establishing the attainability of the right Hashin-Shtrikman bound for volume fractions smaller than 1.

We observe that chirality (captured by the parameter p_2 , or α for a limit two-scale structure) does not play a major role in this case, *i.e.* the right Hashin-Shtrikman bound is closely approximated by structures with no chirality.

The stiffness of such structures (for the considered case $\nu_b = 0$ – the position (E, ν) on $E + \nu = 1$ straight line) is defined solely by the stripe's relative thickness (p_4 for type one structure and θ for

the limit two-scale structure).

Best achievable bound for negative Poisson's ratios.

Introduction of nonzero chirality leads to a decrease of the effective Poisson's ratio due to the rotation of solid segments of the structure (Figure 5.8). Extremal values that closely approach the left theoretical bound (Figure 5.9(A)) are achieved by the Triangle structure, which is, in a sense, a geometric dual of the Hexagon structure. Similarly to the right bound, increasing the number of beams in the laminate area leads to structures closer to the boundary (Figure 5.6(C)). However, in this case, the limit structure does *not* follow the boundary. Figure 5.9(B) demonstrates this coverage coefficient as a function of the volume fraction ϕ . The coverage curve grows rapidly when approaching volume fraction 1. However, reasonable extrapolation indicate that the coverage does not exceed 0.7. The angle of chirality α of the extreme structures is always close to $\pi/3$, however, it depends slightly on the Young's modulus (Figure 5.9(C)).

The right bound achieved in our numerical tests is reasonably well approximated by a fourth order polynomial (Figure 5.6(B)):

$$E(\nu) = 1 + (1 + c_1 - c_2 + c_3)\nu + c_1\nu^2 + c_2\nu^3 + c_3\nu^4 \quad (5.10)$$

with the constants $c_1 = 3.477$, $c_2 = 3.098$, $c_3 = 0.988$.

Intermediate volume fractions.

Above we could see that our Triangle and Hexagon structures exhibit wide coverage of elastic properties for volume fractions approaching 1. Next, we examine the performance of these structures for intermediate volume fractions.

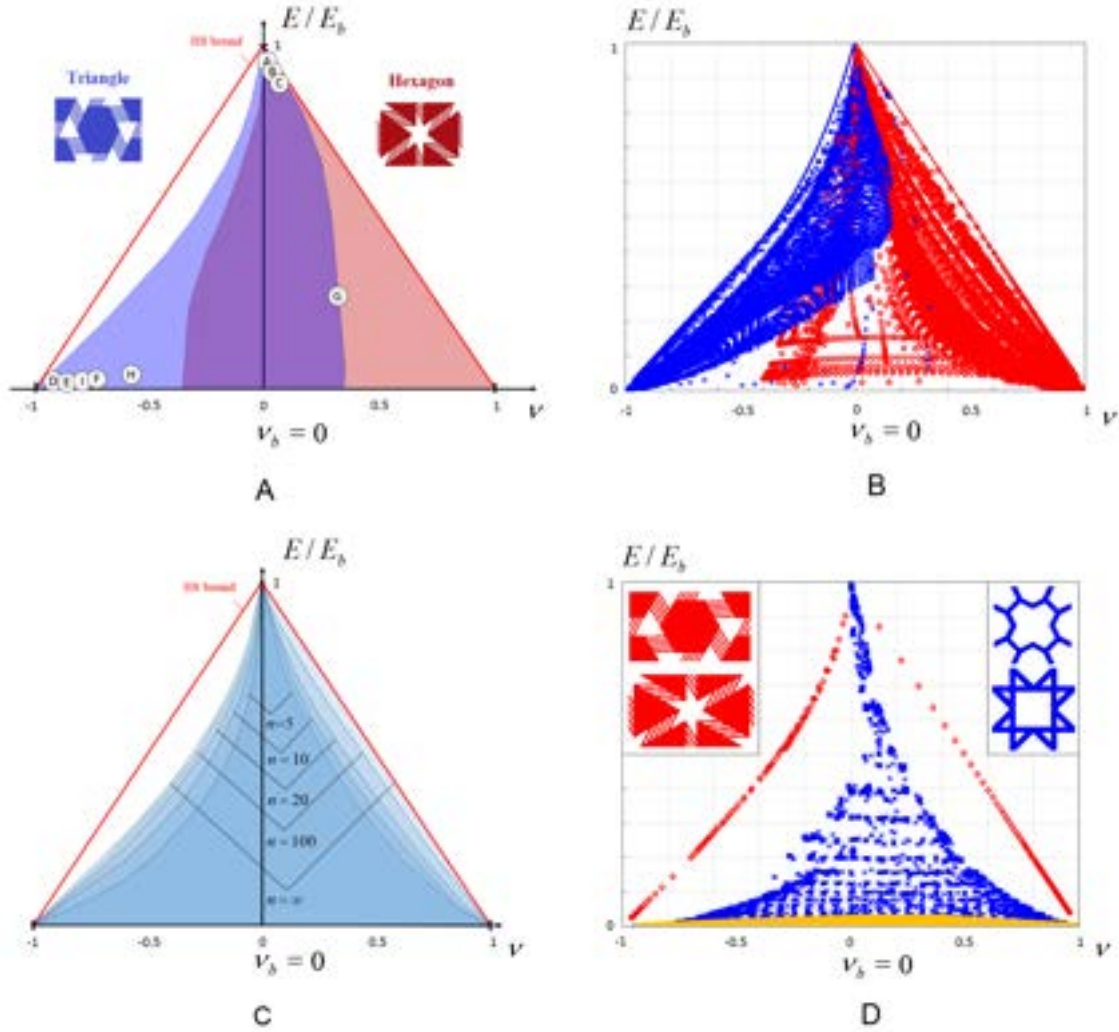


Figure 5.6: (A, B) Coverage of Hashin-Shtrikman bounds with the new structures of all types. Letters denote tentative positions of the extremal composites presented in Figure 5.3. (A) - covered sets, (B) - raw data samples and the conjectured analytical bounds. (C) Expansion of the coverage as the number of beams $n = 1/p_3$ increases. (D) Comparison of the coverage obtained in this work with Triangle and Hexagon structures with $(n = 30, v_b = 0)$, only extreme structures are shown, red points), in comparison with the widest coverages obtained based on previous numerical studies [Panetta et al. 2015b, 2017b] (blue points), and analytically computed coverage achieved with isotropic laminates proposed in [Milton 1992] (yellow region, parameters of lamination are swept maintaining isotropy, the ratio between stiffnesses of weak and strong phases is 10^{-6}).

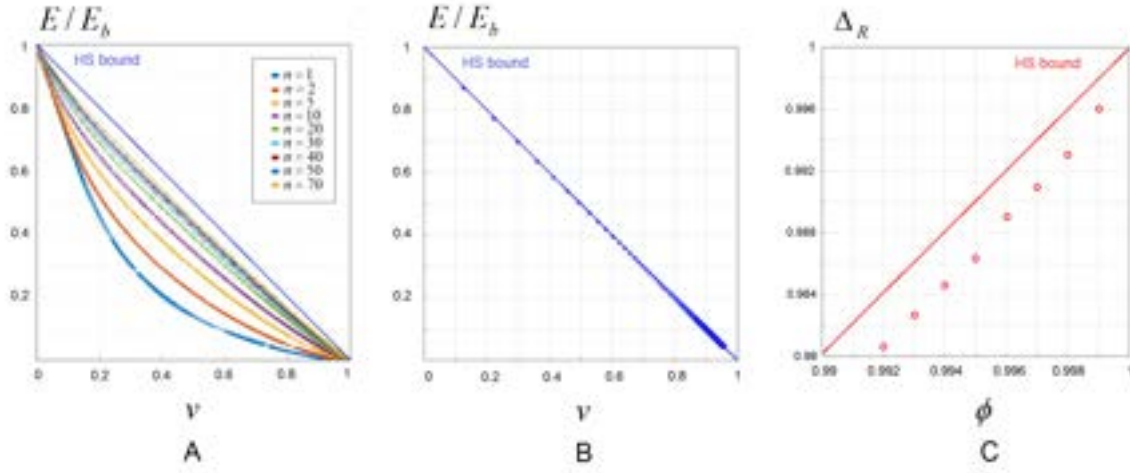


Figure 5.7: (A) Coverage of elastic properties for positive Poisson's ratios as a function of the number of beams in the joint (B) Coverage achieved in the limit case of two-scale structure (C) coverage measure Δ_R as a function of volume fraction ϕ

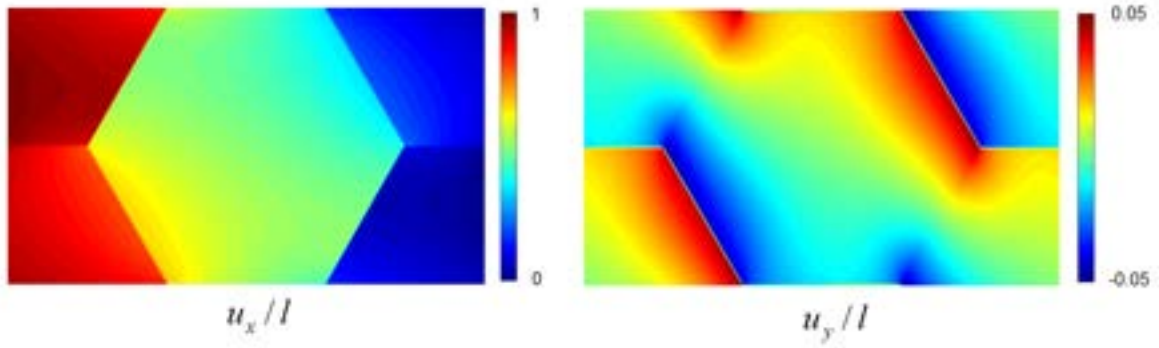


Figure 5.8: Displacement fields in the horizontal tension ($pq = 11$, Triangle limit structure) cell problem. Rotation of solid parts of the structure, leads to auxetic behavior.

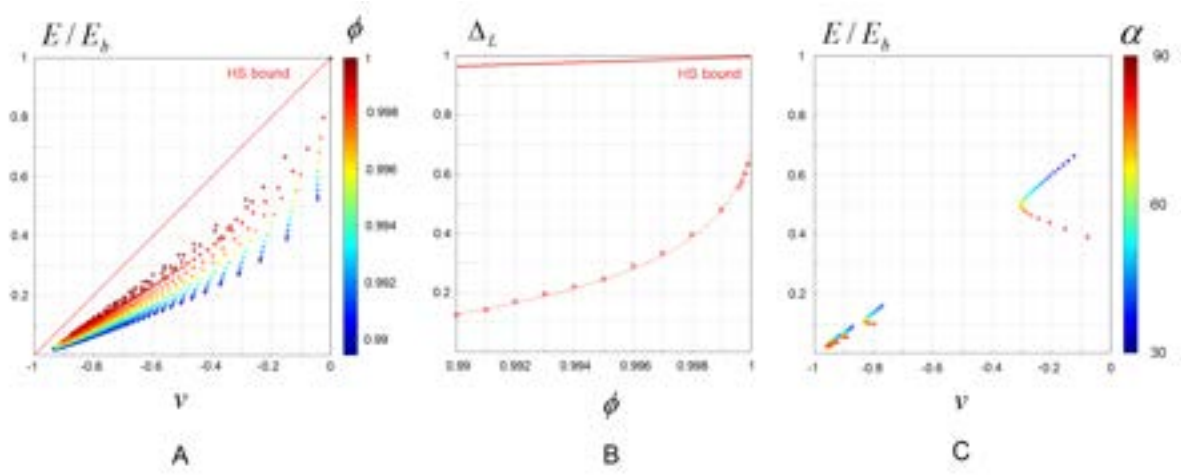


Figure 5.9: Limit structures approaching left HS bound. (A,B) Left-bound extreme elastic moduli, achieved with the Hexagon limit structure; chart colored by values of p_1 (A) and p_2 (B). (C) Dependence of the achievable negative Poisson's ratio as the function of chirality angle. Chirality angle p_3 sweeps are performed for several values of p_2 .

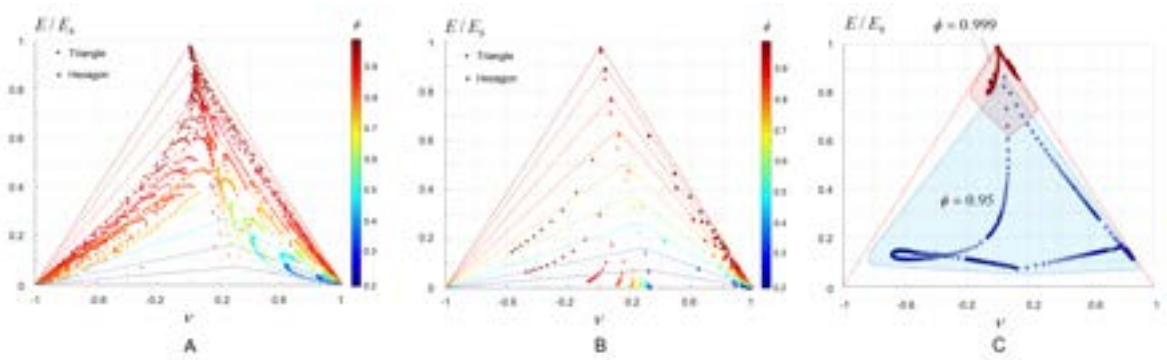


Figure 5.10: Coverage of Hashin-Shtrikman bounds with triangular and hexagonal structures for a set of intermediate volume fractions. (A) Structures of the first type, (B) structures of the second type. (C) Coverage of Cherkaev-Gibiansky bounds with bimaterial composite for two volume fractions.

For both Triangle and Hexagon structures the volume fraction is given by:

$$\phi = 2p_4p_1(1 - p_1) + p_1^2, \quad (5.11)$$

or, for the limit two-scale structure,

$$\phi = 2\theta p_1(1 - p_1) + p_1^2. \quad (5.12)$$

Considering that for the extremal composites $p_4(\theta)$ has to stay very close to 1, the volume fraction can be viewed primarily as the function of a single argument p_1 .

Figure 5.10 illustrates the coverage of Hashin-Shtrikman bounds for a set of intermediate volume fractions. Figure 5.10 (A) shows the parametric sweeps for structures with fixed number of beams, Figure 5.10 (B) demonstrates the behavior of their two-scale counterparts.

For both single scale and two-scale structures, we explored the parameter p_1 in the range $0.1 \dots 0.9$. We can see that in both cases the right Hashin-Shtrikman bound is easily approached with Hexagon structures with zero chirality angle, however, the coverage on the left side deteriorates dramatically with decrease of the base material volume fraction. This effect has trivial explanation. Consider the case of two-scale limit structure. Clearly, if $\theta < 1$ and the lamination angle is steeper than the diagonal of the rectangular joint region, the stiffness of the joint drops to zero. Therefore, for the values of p_1 smaller than the critical value defined by

$$\sin \alpha = \frac{p_1}{\sqrt{3(1 - p_1)^2 + p_1^2}}, \quad (5.13)$$

the stiffness of the Hexagon structure will drop to zero. For $\alpha = \pi/6$, the critical value of $p_1 = 0.5$, and the corresponding volume fraction $\phi = 0.75$. Figure 5.10(B) shows that all structures with this

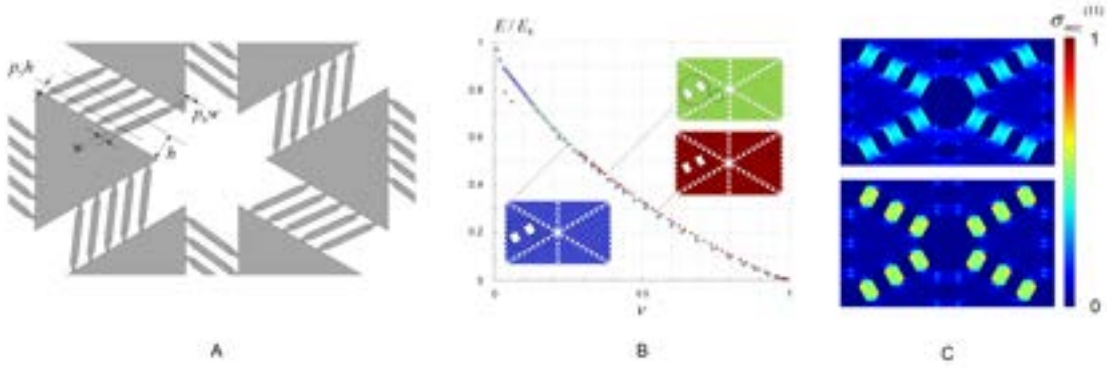
volume fraction and $p_2 < 1$ have nearly zero stiffness, with the remaining response is determined by the nonzero density of the ersatz material.

Non-void weak material. In case of a bimaterial mixture, the range of the elastic properties that could be achieved with our structures shrinks significantly compared to the corresponding theoretical bounds. Figure 5.10(C) gives the coverage achieved for the composite material with the weak phase 1000 times softer than the strong phase. For these cases we performed same parameter sweeps that provided us the most extreme properties in the case of void-material composite. We can see that though the structures without chirality remain close to right bound, the chiral bimaterial structures are very far from the extreme properties.

Expanding the parameter space.

Potentially, expanding the parameter space may improve the performance of our structures. One obvious way to improve the performance is to modify the configuration of joints, that are supposed to be very strong in axial compression, but very compliant in shear. The performance of the joints can be improved by introducing "indentation" regions at the tips of the beams (Figure 5.11(A)). In our numerical experiments the structures with indented beams does not improve the coverage of the elastic moduli at large volume fractions, since the features of sharpened stripes simply can not be resolved. However, they can be useful for the case of intermediate volume fractions – low values of shear modulus can be reached at a lower complexity of the structure (*i.e.* smaller number of stripes), which can be important in practical situations. Figure 5.11(B) illustrates the improvement of the coverage achieved with the structures with intended beams compared to the original ones. However, this relatively insignificant improvement comes at a cost – higher stress concentrations at the tips of the stripes, that make failures more likely (Figure 5.11(C)).

Optimality of the negative Poisson's ratio bounds.



(A) Expanding parameter space: a parameterization of the structure with indented beams.

Figure 5.11: (B) Change in coverage with addition of indentations. (C) Von Mises stress concentrations caused by the introduction of beam indentations.

It is difficult to establish with certainty that no points closer to the Hashin-Shtrikman bounds can be found. However, it is highly likely that a very different structure would be needed for this. To validate this conjecture numerically, we have performed the following numerical experiments. Starting from a structure with effective elastic parameters far away from the achievable boundary (but of sufficiently large number of beams), we vary the parameters of the structure to optimize the goal functional. We consider the functional $J = \frac{1}{2} \|S^H - S^*\|_F^2$, the Frobenius norm of the difference between the effective elasticity tensor of our structure, and a target tensor S^* , which we take to be a point on the Hashin-Shtrikman boundary $(-0.5, 0.5)$. We optimize this functional until the point reaches a minimum which is on the achievable boundary.

At this point, the gradient of the functional, with respect to the structure parameters, is zero (or, rather, its projection on the hyperplane orthogonal to the constraint gradients). This of course, does not imply optimality, as we are considering only a small number of parameters. However, we can measure the norm of the shape derivatives vector with respect to *positions of all points on the shape boundary*.

The shape derivative measures how perturbing the shape affects the objective function (given by

$J = \frac{1}{2} \|S^H - S^*\|_F^2$). To compute it, we use the following formulation [Panetta et al. 2015b]:

$$\begin{aligned} dJ[v] &= [S^H - S^*] : dS^H[v] \\ dS^H[v] &= -S^H : dC^H[v] : S^H \\ dC^H[v] &= \frac{1}{|Y|} \int_{\partial w} [(A_{ij} + \epsilon(w_{ij})) : C^{\text{base}} : (A_{kl} + \epsilon(w_{kl}))] (v \cdot \hat{n}) dA(y) \end{aligned}$$

Here, v is a vector field corresponding to the velocity of the shape. Also, as described before, A_{kl} represents the unit test strain, while w_{kl} is the microscopic fluctuation in base cell Y .

In the discrete setting, the shape derivative can also be written as a sum of inner products: $\sum_m sd_m \cdot v_m$, where v_m is the velocity on vertex m of the input mesh and sd_m represents the shape derivative at this same node. This enable us to compute the norm of the derivative.

The values of shape derivative are tabulated in Table 1, as a function of the iteration number. We see that close to the boundary the derivative declines by more than an order of magnitude. Examining the distribution of nonzero values, we see that these are almost entirely close to sharp corners of the structure, and are close to zero elsewhere. This suggests that adding degrees of freedom to optimization is not likely to improve how close one can get to the Hashin-Shtrikman bound for negative Poisson's ratios. The bound we observe appears to be very close to a local optimum for the topology we consider. This does not imply global optimality of our structures; however, a large change in geometry or topology is likely to be needed for further improvement.

5.6 DISCUSSION, CONCLUSIONS AND FUTURE WORK

In our work we have demonstrated nearly complete G-closure for positive Poisson's ratio – effectively, attainability of Hashin-Shtrikman bounds for the elastic properties. While for negative Poisson's ratio we have not reached the bound, and it remains unclear whether Hashin-Shtrikman

Table 5.1: Norm of the shape derivatives vector as a function of iteration number

N	$\ \nabla\ _2$	N	$\ \nabla\ _2$
1	172.64	11	19.24
2	170.82	12	18.65
3	169.76	13	13.73
4	168.05	14	13.67
5	165.05	15	10.15
6	117.15	16	10.37
7	90.77	17	8.78
8	90.54	18	9.00
9	51.56	19	9.03
10	31.42	20	8.67

bounds are optimal in this case, presented family of structure has broader coverage than any previously considered family.

The region that is left not covered by our structure is the neighborhood of the left HS bound with values of Young's modulus close to 0.5. A number of alternative approaches (*e.g.* topology optimization) did not allow us to reach the extremal composite that would attain an isotropic elastic tensor in that region. The questions of the existence of such microstructures, as well as the optimality of Hashin-Shtrikman bounds in that neighborhood remains unanswered.

The range of achievable elastic properties available with our two microstructures covers all known two-dimensional isotropic void-material composites (some of them are shown in Figures 3 and 6(A,D)). We are not aware of the regular isotropic structures, random assemblages or general topology optimization solutions that provide elastic moduli that fall out of the coverage presented in our work. Therefore, our work suggests the most robust solution of the two-dimensional problem of inverse homogenization of void material isotropic composites available today.

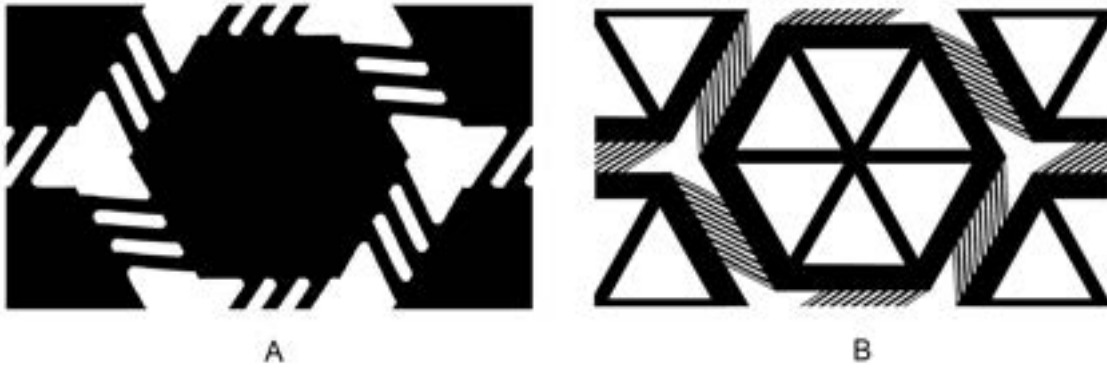


Figure 5.12: Possible generalizations of the extremal structures. (A) Structure with an improved strength due to elimination of stress concentrations (B) Structure with wide coverage of elastic moduli at intermediate volume fractions.

The theoretical structures suggested in our work can be adapted for practical usage. For example, sharp corners causing stress singularities can be smoothened out (Figure 5.12(A)). On the other hand, a straightforward modification of the structure (Figure 5.12(B)) can dramatically improve the performance of the microstructures with intermediate volume fractions.

Although the microstructures suggested in this work do not allow straightforward generalization of the three-dimensional case, similar ideas (laminate-like regions and chirality) can be used to improve the performance of the earlier developed 3D patterns with programmable elastic properties [Panetta et al. 2015b, 2017b].

5.7 CHAPTER NOTES - APPROXIMATION ACCURACY

The discretization used in our study introduces errors in the behavior of homogenized elastic constants, we have compared our approach to a more precise, but significantly slower approach based on unstructured conforming meshing of structure boundaries.

The counteracting sources of this discrepancy is inadequate stiffness of inclined beam elements

Table 5.2: Matching regular grid computations and unstructured mesh computations. Beam thicknesses parameter for the regular grid is chosen to match the values of the Young's modulus found on an unstructured grid

Type	Grid	E	ν	p_1	p_2	p_3	p_4	ε
Hexagon	Reg.	0.3553	0.4917	0.75	0.1	1.0	0.77	0.0073
	Unstr.	0.3554	0.4873	0.75	0.1	1.0	0.74	
	Reg.	0.3121	0.4595	0.74	0.2	1.0	0.546	0.0173
	Unstr.	0.3123	0.4500	0.74	0.2	1.0	0.526	
	Reg.	0.6846	0.1734	0.88	0.1	1.0	0.657	0.0111
	Unstr.	0.6845	0.1656	0.88	0.1	1.0	0.642	
Triangle	Reg.	0.1408	-0.5154	0.73	0.2	0.7	0.805	0.0098
	Unstr.	0.1409	-0.5205	0.73	0.2	0.7	0.787	
	Reg.	0.0760	-0.7337	0.84	0.2	0.65	0.794	0.0051
	Unstr.	0.0759	-0.7375	0.84	0.2	0.65	0.769	
	Reg.	0.7685	0.0112	0.93	0.2	0.9	0.846	0.0143
	Unstr.	0.7681	0.0010	0.93	0.2	0.9	0.812	

that are represented on a regular grid and replacement of very low stiffness soft phase with a stiffer material. To compare effective properties achieved by the two discretization, we match the effective Young modulus for two structures, by varying the beam thickness (p_4) while keeping other parameters the same and compare the effective Poisson ratio. One achieves the accuracy better than one percent (Table 5.2) on several samples for both structures, which indicates that the coverage obtained by using the alternative slow method is very close.

This level of accuracy applies only for the situations when the grid refinement is sufficient to resolve fine features, beam elements and notches between them.

BIBLIOGRAPHY

- Aage, N., Andreassen, E., Lazarov, B. S., and Sigmund, O. (2017). Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84–86.
- Aigerman, N. and Lipman, Y. (2013). Injective and bounded distortion mappings in 3d. *ACM Trans. Graph.*, 32(4):106:1–106:14.
- Aigerman, N., Poranne, R., and Lipman, Y. (2014). Lifted bijections for low distortion surface mappings. *ACM Trans. Graph.*, 33(4):69:1–69:12.
- Alappat, C., Basermann, A., Bishop, A. R., Fehske, H., Hager, G., Schenk, O., Thies, J., and Wellein, G. (2020). A recursive algebraic coloring technique for hardware-efficient symmetric sparse matrix-vector multiplication. *ACM Trans. Parallel Comput.*, 7(3).
- Alart, P. and Curnier, A. (1991). A mixed formulation for frictional contact problems prone to Newton like solution methods. *CMAME*, 92(3).
- Allaire, G. (2002a). *Shape optimization by the homogenization method*, volume 146. Springer.
- Allaire, G. (2002b). *Shape optimization by the homogenization method*. Springer.
- Allaire, G. and Jouve, F. (2008). Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 32(11):909 – 918. Shape and Topological Sensitivity Analysis: Theory and Applications.

- Allaire, G., Jouve, F., and Maillot, H. (2004). Topology optimization for minimum stress design with the homogenization method. *Structural and Multidisciplinary Optimization*, 28(2-3):87–98.
- Andreassen, E., Lazarov, B. S., and Sigmund, O. (2014). Design of manufacturable 3D extremal elastic microstructure. *Mechanics of Materials*, 69(1):1–10.
- Bächer, M., Knoop, E., and Schumacher, C. (2021). Design and control of soft robots using differentiable simulation. *Current Robotics Reports*, pages 1–11.
- Baque, P., Remelli, E., Fleuret, F., and Fua, P. (2018). Geodesic convolutional shape optimization. In *International Conference on Machine Learning*, pages 472–481. PMLR.
- Belytschko, T., Liu, W., and Moran, B. (2000). *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, Ltd.
- Bendsøe, M. P. (1989). Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202.
- Bendsøe, M. P. and Sigmund, O. (2003). *Topology optimization: theory, methods and applications*. Springer.
- Beremlijski, P., Haslinger, J., Outrata, J., and Pathó, R. (2014). Shape Optimization in Contact Problems with Coulomb Friction and a Solution-Dependent Friction Coefficient. *SIAM Journal on Control and Optimization*, 52(5):3371–3400.
- Bern, J., Banzet, P., Poranne, R., and Coros, S. (2019). Trajectory optimization for cable-driven soft robot locomotion. In *Robotics: Science and Systems XV*, volume 1. Robotics: Science and Systems Foundation.
- Bern, J. M., Schnider, Y., Banzet, P., Kumar, N., and Coros, S. (2020). Soft robot control with a learned differentiable model. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 417–423. IEEE.

- Bickel, B., Bäcker, M., Otaduy, M. A., Lee, H. R., Pfister, H., Gross, M., and Matusik, W. (2010). Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10.
- Bollhöfer, M., Eftekhari, A., Scheidegger, S., and Schenk, O. (2019). Large-scale sparse inverse covariance matrix estimation. *SIAM Journal on Scientific Computing*, 41(1):A380–A401.
- Bollhöfer, M., Schenk, O., Janalik, R., Hamm, S., and Gullapalli, K. (2020). State-of-the-art sparse direct solvers. pages 3–33.
- Bommes, D., Campen, M., Ebke, H.-C., Alliez, P., and Kobbelt, L. (2013). Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.*, 32(4):98:1–98:12.
- Bonnetier, É. and Dapogny, C. (2020). An introduction to shape and topology optimization.
- Bouaziz, S., Deuss, M., Schwartzburg, Y., Weise, T., and Pauly, M. (2012). Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum*, 31(5):1657–1667.
- Bouaziz, S., Martin, S., Liu, T., Kavan, L., and Pauly, M. (2014). Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4):154:1–154:11.
- Brewer, M., Diachin, L., Knupp, P., Leurent, T., and Melander, D. (2003). The mesquite mesh quality improvement toolkit. In *Proc. of the 12th International Meshing Roundtable*, pages 239–250.
- Bridson, R., Fedkiw, R., and Anderson, J. (2002). Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. on Graph.*, 21.
- Brogliato, B. (1999). *Nonsmooth Mechanics*. Springer-Verlag.
- Brown, G. E., Overby, M., Forootaninia, Z., and Narain, R. (2018). Accurate dissipative forces in optimization integrators. *ACM Trans. Graph.*, 37(6).

- Bückmann, T., Stenger, N., Kadic, M., Kaschke, J., Frölich, A., Kennerknecht, T., Eberl, C., Thiel, M., and Wegener, M. (2012). Tailored 3d mechanical metamaterials made by dip-in direct-laser-writing optical lithography. *Advanced Materials*, 24(20):2710–2714.
- Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. (2016). A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*.
- Chen, B., Wang, N., Zhang, X., and Chen, W. (2020). Design of dielectric elastomer actuators using topology optimization on electrodes. *Smart Mater. Struct.*, 29(7):075029.
- Chen, D., Skouras, M., Zhu, B., and Matusik, W. (2018). Computational discovery of extremal microstructure families. *Science advances*, 4(1):eaao7005.
- Cherkaev, A. (2000). *Variational methods for structural optimization*, volume 140. Springer.
- Cherkaev, A. (2012). *Variational Methods for Structural Optimization*. Springer Science and Business Media.
- Cherkaev, A. and Gibiansky, L. (1993). Coupled estimates for the bulk and shear moduli of a two-dimensional isotropic elastic composite. *Journal of the Mechanics and Physics of Solids*, 41(5):937–980.
- Cherkaev, A. V., Lurie, K. A., and Milton, G. W. (1992). Invariant properties of the stress in plane elasticity and equivalence classes of composites. *Proceedings of the Royal Society A*, 428:519–529.
- Cioranescu, D. and Donato, P. (1999). *An introduction to homogenization*. Oxford University Press.
- Coevoet, E., Escande, A., and Duriez, C. (2019). Soft robots locomotion and manipulation control using fem simulation and quadratic programming. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 739–745. IEEE.

- Coevoet, E., Morales-Bieze, T., Largilliere, F., Zhang, Z., Thieffry, M., Sanz-Lopez, M., Carrez, B., Marchal, D., Goury, O., Dequidt, J., and Duriez, C. (2017). Software toolkit for modeling, simulation and control of soft robots. *Advanced Robotics*, 31:1208–1224.
- Coumans, E. et al. (2013). Bullet physics library. *Open source: bulletphysics.org*, 15(49):5.
- Daviet, G., Bertails-Descoubes, F., and Boissieux, L. (2011). A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Trans. on Graph.*, 30.
- de Vaucorbeil, A., Nguyen, V. P., Sinaie, S., and Wu, J. Y. (2019). Material point method after 25 years: theory, implementation and applications. *Submitted to Advances in Applied Mechanics*, page 1.
- Degener, P., Meseth, J., and Klein, R. (2003). An adaptable surface parameterization method. In *Proceedings of the 12th International Meshing Roundtable*, pages 201–213.
- Desmorat, B. (2007). Structural rigidity optimization with frictionless unilateral contact. *International Journal of Solids and Structures*, 44(3):1132–1144.
- Deuss, M., Deleuran, A. H., Bouaziz, S., Deng, B., Piker, D., and Pauly, M. (2015). *ShapeOp—A Robust and Extensible Geometric Modelling Paradigm*, pages 505–515. Springer International Publishing, Cham.
- Ding, O. and Schroeder, C. (2020). Penalty force for coupling materials with coulomb friction. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2443–2455.
- Du, T., Wu, K., Ma, P., Wah, S., Spielberg, A., Rus, D., and Matusik, W. (2021). Diffpd: Differentiable projective dynamics. *ACM Trans. Graph.*, 41(2).
- Eck, C., Jarusek, J., Krbec, M., Jarusek, J., and Krbec, M. (2005). *Unilateral Contact Problems : Variational Methods and Existence Theorems*. CRC Press.

- Faure, F., Duriez, C., Delingette, H., Allard, J., Gilles, B., Marchesseau, S., Talbot, H., Courtecuisse, H., Bousquet, G., Peterlik, I., and Cotin, S. (2012). SOFA: A Multi-Model Framework for Interactive Physical Simulation. In Payan, Y., editor, *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pages 283–321. Springer.
- Ferguson, Z. et al. (2020). IPC Toolkit.
- Ferguson, Z., Li, M., Schneider, T., Gil-Ureta, F., Langlois, T., Jiang, C., Zorin, D., Kaufman, D. M., and Panozzo, D. (2021). Intersection-free rigid body dynamics. *ACM Transactions on Graphics (SIGGRAPH)*, 40(4).
- Francfort, G. and Murat, F. (1986). Homogenization and optimal bounds in linear elasticity. *Archives of Rational Mechanics Analysis*, 94:307–334.
- Fu, X.-M. and Liu, Y. (2016). Computing inversion-free mappings by simplex assembly. *ACM Trans. Graph.*, 35(6):216:1–216:12.
- Fu, X.-M., Liu, Y., and Guo, B. (2015). Computing locally injective mappings by advanced mips. *ACM Trans. Graph.*, 34(4):71:1–71:12.
- Garner, E., Kolken, H. M. A., Wang, C. C. L., Zadpoor, A. A., and Wu, J. (2019). Compatibility in microstructural optimization for additive manufacturing. *Additive Manufacturing*, 26:65–75.
- Gavriil, K., Guseinov, R., Pérez, J., Pellis, D., Henderson, P., Rist, F., Pottmann, H., and Bickel, B. (2020). Computational design of cold bent glass façades. *ACM Trans. Graph.*, 39(6).
- Gaynor, A. T. and Guest, J. K. (2016). Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5):1157–1172.

- Geilinger, M., Hahn, D., Zehnder, J., Bächer, M., Thomaszewski, B., and Coros, S. (2020a). Add: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Trans. Graph.*, 39(6).
- Geilinger, M., Hahn, D., Zehnder, J., Bächer, M., Thomaszewski, B., and Coros, S. (2020b). Add: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6):1–15.
- Geuzaine, C. and Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331.
- Gil-Ureta, F., Pietroni, N., and Zorin, D. (2019). Structurally optimized shells. *arXiv preprint arXiv:1904.12240*.
- Gjoka, A., Huang, Z., Tozoni, D. C., Ferguson, Z., Schneider, T., Panozzo, D., and Zorin, D. (2022). Differentiable solver for time-dependent deformation problems with contact.
- Goury, O. and Duriez, C. (2018). Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Trans. Robot.*, 34(6):1565–1576.
- Greaves, G. N., Greer, A. L., Lakes, R. S., and Rouxel, T. (2011). Poisson’s ratio and modern materials. *Nature Materials*, 10(11):823–837.
- Grima, J., Alderson, A., and Evans, K. (2005). Auxetic behavior from rotating rigid units. *Physica Status Solidi B*, 242:561–575.
- Grima, J. and Evans, K. (2000). Auxetic behavior from rotating squares. *Journal of Materials Science Letters*, 19:1563–1565.

- Groen, J. P. and Sigmund, O. (2017). Homogenization-based topology optimization for high-resolution manufacturable microstructures. *International Journal for Numerical Methods in Engineering*, 113(8):1148–1163.
- Groen, J. P., Wu, J., and Sigmund, O. (2019). Homogenization-based stiffness optimization and projection of 2d coated structures with orthotropic infill. *Computer Methods in Applied Mechanics and Engineering*, 349:722–742.
- H, M., J, S., F, S., M, S., C, K., RF, S., V, R., K, M., and GE, S.-T. (2011). Finding auxetic frameworks in periodic tessellations. *Advanced Materials*, 23(22-23):2669–2674.
- Hafner, C., Schumacher, C., Knoop, E., Auzinger, T., Bickel, B., and Bächer, M. (2019). X-cad: Optimizing cad models with extended finite elements. *ACM Trans. Graph.*, 38(6):1–15.
- Hahn, D., Banzet, P., Bern, J. M., and Coros, S. (2019). Real2sim: Visco-elastic parameter estimation from dynamic motion. *ACM Trans. Graph.*, 38(6):1–13.
- Harmon, D., Vouga, E., Smith, B., Tamstorf, R., and Grinspun, E. (2009). Asynchronous contact mnumpageeconomics. In *ACM Trans. on Graph. (TOG)*, volume 28. ACM.
- Harmon, D., Vouga, E., Tamstorf, R., and Grinspun, E. (2008). Robust treatment of simultaneous collisions. *SIGGRAPH (ACM Trans. on Graph.)*, 27(3).
- Hashin, Z. (1962). The elastic moduli of heterogeneous materials. *Journal of Applied Mechanics*, 29(1):142–150.
- Hashin, Z. and Shtrikman, S. (1963). A variational approach to the theory of elastic behavior of multiphase materials. *Journal of the Mechanics and Physics of Solids*, 11:127–140.
- Haslinger, J., Neittaanmaki, P., and Tiihonen, T. (1986). Shape optimization in contact problems based on penalization of the state inequality. *Aplikace matematiky*, 31(1):54–77.

- Heiden, E., Millard, D., Coumans, E., Sheng, Y., and Sukhatme, G. S. (2020). Neuralsim: Augmenting differentiable simulators with neural networks. *arXiv preprint arXiv:2011.04217*.
- Herskovits, J., Leontiev, A., Dias, G., and Santos, G. (2000). Contact shape optimization: a bilevel programming approach. *Structural and Multidisciplinary Optimization*, 20(3):214–221.
- Hiller, J. and Lipson, H. (2009). Design and analysis of digital materials for physical 3d voxel printing. *Rapid Prototyping Journal*, 15(2):137–149.
- Hollister, S. J. (2005). Porous scaffold design for tissue engineering. *Nature Materials*, 4(7):518–524.
- Hormann, K. and Greiner, G. (2000). MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, pages 153–162.
- Hoshyari, S., Xu, H., Knoop, E., Coros, S., and Bächer, M. (2019). Vibration-minimizing motion retargeting for robotic characters. *ACM Trans. Graph.*, 38(4):1–14.
- Hu, Y., Anderson, L., Li, T.-M., Sun, Q., Carr, N., Ragan-Kelley, J., and Durand, F. (2019a). DiffTaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations*.
- Hu, Y., Liu, J., Spielberg, A., Tenenbaum, J. B., Freeman, W. T., Wu, J., Rus, D., and Matusik, W. (2019b). Chainqueen: A real-time differentiable physical simulator for soft robotics. In *2019 International conference on robotics and automation (ICRA)*, pages 6265–6271. IEEE.
- Hu, Y., Schneider, T., Wang, B., Zorin, D., and Panozzo, D. (2020a). Fast tetrahedral meshing in the wild. *ACM Trans. Graph.*, 39(4).
- Hu, Y., Schneider, T., Wang, B., Zorin, D., and Panozzo, D. (2020b). Fast tetrahedral meshing in the wild. *ACM Trans. Graph.*, 39(4).

- Huang, X. and Xie, Y.-M. (2010). A further review of eso type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 41(5):671–683.
- Ion, A., Frohnhofen, J., Wall, L., Kovacs, R., Alistar, M., Lindsay, J., Lopes, P., Chen, H.-T., and Baudisch, P. (2016). Metamaterial mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 529–539, New York, NY, USA. ACM.
- Ion, A., Lindlbauer, D., Herholz, P., Alexa, M., and Baudisch, P. (2019). Understanding metamaterial mechanisms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 647:1–647:14, New York, NY, USA. ACM.
- Jean, M. and Moreau, J. J. (1992). Unilaterality and dry friction in the dynamics of rigid body collections. In *Proc. of Contact Mech. Int. Symp.*, volume 1.
- Jiang, C., Tang, C., Vaxman, A., Wonka, P., and Pottmann, H. (2015). Polyhedral patterns. *ACM Trans. Graphics*, 34(6). Proc. SIGGRAPH Asia.
- Jiang, Z., Schneider, T., Zorin, D., and Panozzo, D. (2020). Bijective projection in a shell. *ACM Trans. Graph.*, 39(6).
- Kang, H. S. (2010). *Hierarchical design and simulation of tissue engineering scaffold mechanical, mass transport, and degradation properties*. PhD thesis, The University of Michigan.
- Kikuchi, N. and Oden, J. T. (1988). *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, volume 8 of *SIAM Studies in App. and Numer. Math.* Society for Industrial and Applied Mathematics.
- Knupp, P. M. (2001). Algebraic mesh quality metrics. *SIAM Journal on Scientific Computing*, 23(1):193–218.
- Konaković-Luković, M., Panetta, J., Crane, K., and Pauly, M. (2018). Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics (TOG)*, 37(4):106.

- Kovalsky, S. Z., Aigerman, N., Basri, R., and Lipman, Y. (2015). Large-scale bounded distortion mappings. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH Asia)*, 34(6).
- Kovalsky, S. Z., Galun, M., and Lipman, Y. (2016). Accelerated quadratic proxy for geometric optimization. *ACM Trans. Graph.*, 35(4):134:1–134:11.
- Lakes, R. (1987). Foam structures with a negative poisson’s ratio. *Science*, 235(4792):1038–1040.
- Lakes, R. (2017). Negative Poisson’s ratio materials: Auxetic solids. *Annual Review of Materials Research*, 47:63–81.
- Langelaar, M. (2016). An additive manufacturing filter for topology optimization of print-ready designs. *Structural and Multidisciplinary Optimization*, pages 1–13.
- Li, M., Ferguson, Z., Schneider, T., Jiang, C., Zorin, D., Panozzo, D., and Kaufman, D. M. (2022). Convergent incremental potential contact. in preparation.
- Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., and Kaufman, D. M. (2020a). Incremental potential contact: Intersection- and inversion-free large deformation dynamics. *ACM Transactions on Graphics*, 39(4).
- Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., and Kaufman, D. M. (2020b). Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics (TOG)*, 39(4):49–1.
- Lian, H., Christiansen, A. N., Tortorelli, D. A., Sigmund, O., and Aage, N. (2017). Combined shape and topology optimization for minimization of maximal von mises stress. *Structural and Multidisciplinary Optimization*, 55(5):1541–1557.
- Liang, J., Lin, M., and Koltun, V. (2019). Differentiable cloth simulation for inverse problems. *Neural Information Processing Systems*.

- Lin, C.-Y., Hsiao, C.-C., Chen, P.-Q., and Hollister, S. J. (2004a). Interbody fusion cage design using integrated global layout and local microstructure topology optimization. *Spine*, 29(16):1747–1754. PMID: 15303018.
- Lin, C. Y., Kikuchi, N., and Hollister, S. J. (2004b). A novel method for biomaterial scaffold internal architecture design to match bone elastic properties with desired porosity. *Journal of Biomechanics*, 37(5):623–636.
- Lipman, Y. (2012). Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph.*, 31(4):108:1–108:13.
- Liu, H., Hu, Y., Zhu, B., Matusik, W., and Sifakis, E. (2018). Narrow-band topology optimization on a sparsely populated grid. *ACM Trans. Graph.*, 37(6):251:1–251:14.
- Livesu, M., Sheffer, A., Vining, N., and Tarini, M. (2015). Practical hex-mesh optimization via edge-cone rectification. *Transactions on Graphics (Proc. SIGGRAPH 2015)*, 34(4).
- Love, A. (1944). *A treatise on the mathematical theory of elasticity*. Dover publications New York.
- Lurie, K. and Cherkaev, A. (1985). The problem of formation of an optimal isotropic multicomponent composite. *Journal of Optimization Theory and Applications*, 46:571–589.
- Macklin, M., Erleben, K., Müller, M., Chentanez, N., Jeschke, S., and Makoviychuk, V. (2019). Non-smooth newton methods for deformable multi-body dynamics.
- Maloisel, G., Knoop, E., Schumacher, C., and Bacher, M. (2021). Automated routing of muscle fibers for soft robots. *IEEE Trans. Robot.*, 37(3):996–1008.
- Malomo, L., Pérez, J., Iarussi, E., Pietroni, N., Miguel, E., Cignoni, P., and Bickel, B. (2018). Flexmaps: Computational design of flat flexible shells for shaping 3D objects. *ACM Trans. Graph.*, 37(6):241:1–241:14.

- Martínez, J., Dumas, J., and Lefebvre, S. (2016). Procedural Voronoi foams for additive manufacturing. *ACM Transactions on Graphics (TOG)*, 35(4):44.
- Martínez, J., Skouras, M., Schumacher, C., Hornus, S., Lefebvre, S., and Thomaszewski, B. (2019). Star-shaped metrics for mechanical metamaterial design. *ACM Transactions on Graphics*, 38(4):13. Special issue, SIGGRAPH 2019.
- Martínez, J., Song, H., Dumas, J., and Lefebvre, S. (2017). Orthotropic k-nearest foams for additive manufacturing. *ACM Transactions on Graphics*, 36(4):1–12.
- Maury, A., Allaire, G., and Jouve, F. (2017). Shape optimisation with the level set method for contact problems in linearised elasticity.
- Milton, G. (1986). Vol. 1 of IMA. In *Modelling the Properties of Composites by Laminates.*, pages 150–174. Springer-Verlag.
- Milton, G. (1992). Composite materials with Poisson’s ratios close to -1. *Journal of the Mechanics and Physics of Solids*, 40(5):1105–1137.
- Milton, G. (2002a). *Theory of Composites*. Cambridge University Press.
- Milton, G. (2015). New examples of three-dimensional dilational materials. *Physica Status Solidi B*, 252(7):1426–1430.
- Milton, G. and Cherkaev, A. (1995). Which elasticity tensors are realizable? *Journal of Engineering Materials and Technology, Transactions of ASME*, 117(4):483–493.
- Milton, G. W. (2002b). *The theory of composites*. Cambridge University Press.
- Milton, G. W., Briane, M., and Harutyunyan, D. (2017). On the possible effective elasticity tensors of 2-dimensional and 3-dimensional printed materials. *Mathematics and Mechanics of Complex Systems*, 5(1):381–391.

- Mitusch, S. K., Funke, S. W., and Dokken, J. S. (2019). dolfin-adjoint 2018.1: automated adjoints for fenics and firedrake. *Journal of Open Source Software*, 4(38):1292.
- Moreau, J. J. (1973). On unilateral constraints, friction and plasticity. *New Variational Tech. in Math. Phys.*
- Morzadec, T., Marcha, D., and Duriez, C. (2019). Toward shape optimization of soft robots. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 521–526. IEEE.
- Nakasone, P. and Silva, E. (2010). Dynamic design of piezoelectric laminated sensors and actuators using topology optimization. *Journal of Intelligent Material Systems and Structures*, 21(16):1627–1652.
- Navarro, S. E., Nagels, S., Alagi, H., Faller, L.-M., Goury, O., Morales-Bieze, T., Zangl, H., Hein, B., Ramakers, R., Deferme, W., Zheng, G., and Duriez, C. (2020). A model-based sensor fusion approach for force and shape estimation in soft robotics. *IEEE Robot. Autom. Lett.*, 5(4):5621–5628.
- Neves, M., Rodrigues, H., and Guedes, J. (2000). Optimal design of periodic linear elastic microstructures. *Computers and Structures*, 76(1):421–429.
- Norris, A. N. (1985). A differential scheme for the effective moduli of composites. *Mechanics of Materials*, 4:1–16.
- nTopology (2020). ntop platform. Engineering software built for digital manufacturing.
- NVIDIA (2022). Physx library. <https://developer.nvidia.com/gameworks-physx-overview>.
- Ostanin, I., Ovchinnikov, G., Tozoni, D. C., and Zorin, D. (2018). A parametric class of composites with a large achievable range of effective elastic properties. *Journal of the Mechanics and Physics of Solids*, 118:204–217.

- Otaduy, M., Tamstorf, R., Steinemann, D., and Gross, M. (2009). Implicit contact handling for deformable objects. *Comp. Graph. Forum*, 28.
- Panetta, J., Rahimian, A., and Zorin, D. (2017a). Worst-case stress relief for microstructures. *ACM Transactions on Graphics*, 36(4):1–16.
- Panetta, J., Rahimian, A., and Zorin, D. (2017b). Worst-case stress relief for microstructures. *ACM Transactions on Graphics*, 36(4):122:1–14.
- Panetta, J., Rahimian, A., and Zorin, D. (2017c). Worst-case stress relief for microstructures. *ACM Transactions on Graphics*, 36(4).
- Panetta, J., Zhou, Q., Malomo, L., Pietroni, N., Cignoni, P., and Zorin, D. (2015a). Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)*, 34(4):135.
- Panetta, J., Zhou, Q., Malomo, L., Pietroni, N., Cignoni, P., and Zorin, D. (2015b). Elastic textures for additive fabrication. *ACM Transactions on Graphics*, 34(4):135:1–12.
- Panozzo, D., Block, P., and Sorkine-Hornung, O. (2013). Designing unreinforced masonry models. *ACM Trans. Graph.*, 32(4).
- Polajnar, M., Kosel, F., and Drazumeric, R. (2017). Structural optimization using global stress-deviation objective function via the level-set method. *Structural and Multidisciplinary Optimization*, 55(1):91–104.
- Poranne, R. and Lipman, Y. (2014). Provably good planar mappings. *ACM Trans. Graph.*, 33(4):76:1–76:11.
- Poranne, R., Ovreiu, E., and Gotsman, C. (2013). Interactive Planarization and Optimization of 3D Meshes. *Computer Graphics Forum*.

- Prall, D. and Lakes, R. (1996). Properties of a chiral honeycomb with a Poisson’s ratio -1 . *International Journal of Mechanical Sciences*, 39:305–314.
- Qian, X. (2016). Undercut and overhang angle control in topology optimization: a density gradient based integral approach. *International Journal for Numerical Methods in Engineering*.
- Qiao, Y.-L., Liang, J., Koltun, V., and Lin, M. (2020). Scalable differentiable physics for learning and control. In *International Conference on Machine Learning*, pages 7847–7856. PMLR.
- Rabinovich, M., Poranne, R., Panozzo, D., and Sorkine-Hornung, O. (2017a). Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4):1.
- Rabinovich, M., Poranne, R., Panozzo, D., and Sorkine-Hornung, O. (2017b). Scalable locally injective mappings. *ACM Trans. Graph.*, 36(2):16:1–16:16.
- Rojas, J., Sifakis, E., and Kavan, L. (2021). Differentiable implicit soft-body physics. *arXiv preprint arXiv:2102.05791*.
- Sander, P. V., Snyder, J., Gortler, S. J., and Hoppe, H. (2001). Texture mapping progressive meshes. In *ACM SIGGRAPH*, pages 409–416.
- Schneider, T., Dumas, J., Gao, X., Zorin, D., and Panozzo, D. (2019). PolyFEM. <https://polyfem.github.io/>.
- Schüller, C., Kavan, L., Panozzo, D., and Sorkine-Hornung, O. (2013). Locally injective mappings. In *Symposium on Geometry Processing*, pages 125–135.
- Schumacher, C., Bickel, B., Rys, J., Marschner, S., Daraio, C., and Gross, M. (2015). Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):136.
- Schumacher, C., Knoop, E., and Bacher, M. (2020). Simulation-ready characterization of soft robotic materials. *IEEE Robot. Autom. Lett.*, 5(3):3775–3782.

- Schumacher, C., Marschner, S., Cross, M., and Thomaszewski, B. (2018a). Mechanical characterization of structured sheet materials. *ACM Trans. Graph.*, 37(4):148:1–148:15.
- Schumacher, C., Zehnder, J., and Bäcker, M. (2018b). Set-in-stone: Worst-case optimization of structures weak in tension. *ACM Trans. Graph.*, 37(6).
- Schwerdtfeger, J., Wein, F., Leugering, G., Singer, R. F., Körner, C., Stingl, M., and Schury, F. (2011). Design of auxetic structures via mathematical optimization. *Advanced Materials*, 23(22):2650–2654.
- Sharma, A. and Maute, K. (2018). Stress-based topology optimization using spatial gradient stabilized xfem. *Structural and Multidisciplinary Optimization*, 57(1):17–38.
- Shewchuk, J. R. (1996a). Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In Lin, M. C. and Manocha, D., editors, *Applied Computational Geometry Towards Geometric Engineering*, pages 203–222, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Shewchuk, J. R. (1996b). Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Workshop on Applied Computational Geometry*, pages 203–222. Springer.
- Shtengel, A., Poranne, R., Sorkine-Hornung, O., Kovalsky, S. Z., and Lipman, Y. (2017). Geometric optimization via composite majorization. *ACM Trans. Graph.*, 36(4):38:1–38:11.
- Sigmund, O. (1994). Materials with prescribed constitutive parameters: an inverse homogenization problem. *Computers and Structures*, 31(17):2313–2329.
- Sigmund, O. (2000). A new class of extremal composites. *Journal of the Mechanics and Physics of Solids*, 48:397–428.
- Skouras, M., Thomaszewski, B., Coros, S., Bickel, B., and Gross, M. (2013a). Computational design of actuated deformable characters. *ACM Trans. Graph.*, 32(4).

- Skouras, M., Thomaszewski, B., Coros, S., Bickel, B., and Gross, M. (2013b). Computational design of actuated deformable characters. *ACM Transactions on Graphics (TOG)*, 32(4):82.
- Smith, J. and Schaefer, S. (2015). Bijective parameterization with free boundaries. *ACM Trans. Graph.*, 34(4):70:1–70:9.
- Sonmez, F. (2009). Optimal shape design of shoulder fillets for flat and round bars under various loadings. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(8):1741–1754.
- Sorkine, O. and Alexa, M. (2007). As-rigid-as-possible surface modeling. In *Proceedings of EURO-GRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116.
- Sorkine, O., Cohen-Or, D., Goldenthal, R., and Lischinski, D. (2002). Bounded-distortion piecewise mesh parameterization. In *Proceedings of the Conference on Visualization*, pages 355–362.
- Stava, O., Vanek, J., Benes, B., Carr, N., and Muech, R. (2012). Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11.
- Stewart, D. E. (2001). Finite-dimensional contact mechanics. *Phil. Trans. R. Soc. Lond. A*, 359.
- Stupkiewicz, S., Lengiewicz, J., and Korelc, J. (2010). Sensitivity analysis for frictional contact problems in the augmented Lagrangian formulation. *Computer Methods in Applied Mechanics and Engineering*, 199(33):2165–2176.
- Sun, T. and Zheng, C. (2015). Computational design of twisty joints and puzzles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015)*, 34(4).
- Takayama, K., Panozzo, D., and Sorkine-Hornung, O. (2014). Pattern-based quadrangulation for n-sided patches. In *Proceedings of the Symposium on Geometry Processing, SGP '14*, pages 177–184, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

- Tang, C., Sun, X., Gomes, A., Wallner, J., and Pottmann, H. (2014). Form-finding with polyhedral meshes made simple. *ACM Trans. Graphics*, 33(4). Proc. SIGGRAPH.
- Tapia, J., Knoop, E., Mutný, M., Otaduy, M. A., and Bäcker, M. (2020). Makesense: Automated sensor design for proprioceptive soft robots. *Soft Rob.*, 7(3):332–345.
- Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.
- Torquato, S. (2002a). *Random heterogeneous materials: microstructure and macroscopic properties*, volume 16. Springer.
- Torquato, S. (2002b). *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer.
- Tournois, J., Faraj, N., Thiery, J.-M., and Boubekur, T. (2021). Tetrahedral remeshing. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.3 edition.
- Tozoni, D. C., Dumas, J., Jiang, Z., Panetta, J., Panozzo, D., and Zorin, D. (2020). A low-parametric rhombic microstructure family for irregular lattices. *ACM Trans. Graph.*, 39(4).
- Tozoni, D. C., Zhou, Y., and Zorin, D. (2021). Optimizing contact-based assemblies. *ACM Trans. Graph.*, 40(6).
- Ureta, F. G., Tymms, C., and Zorin, D. (2016). Interactive modeling of mechanical objects. *Computer Graphics Forum*, 35(5):145–155.
- Van Miegroet, L. and Duysinx, P. (2007). Stress concentration minimization of 2d filets using x-fem and level set description. *Structural and Multidisciplinary Optimization*, 33(4-5):425–438.
- Verschoor, M. and Jalba, A. C. (2019). Efficient and accurate collision response for elastically deformable models. *ACM Trans. on Graph. (TOG)*, 38(2).

- Vigdergauz, S. (1989). Regular structures with extremal elastic properties. *Mechanics of Solids*, 24(3):57–63.
- Vigdergauz, S. (1999). Energy-minimizing inclusions in a planar elastic structure with macroisotropy. *Structural Optimization*, 17(203):104–112.
- Vouga, E., Höbinger, M., Wallner, J., and Pottmann, H. (2012). Design of self-supporting surfaces. *ACM Trans. Graph.*, 31(4).
- Wang, Z., Song, P., Isvoranu, F., and Pauly, M. (2019). Design and structural optimization of topological interlocking assemblies. *ACM Trans. Graph.*, 38(6).
- Whalen, E., Beyene, A., and Mueller, C. (2021). SimJEB: Simulated Jet Engine Bracket Dataset. *Computer Graphics Forum*.
- Wieschollek, P. (2016). Cppoptimizationlibrary. <https://github.com/PatWie/CppNumericalSolvers>.
- Wriggers, P. (1995). Finite element algorithms for contact problems. *Archives of Comp. Meth. in Eng.*, 2.
- Wu, J., Aage, N., Westermann, R., and Sigmund, O. (2018). Infill optimization for additive manufacturing—approaching bone-like porous structures. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1127–1140.
- Wu, J., Dick, C., and Westermann, R. (2016). A system for high-resolution topology optimization. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1195–1208.
- X. Huang, A. Radman, Y. X. (2011). Topological design of microstructures of cellular materials for maximum bulk or shear modulus. *Computational Materials Science*, 50:1861–1870.

- Xia, L. and Breitkopf, P. (2015). Design of materials using topology optimization and energy-based homogenization approach in matlab. *Structural and Multidisciplinary Optimization*, 52(6):1229–1241.
- Xia, Q., Shi, T., Liu, S., and Wang, M. Y. (2012). A level set solution to the stress-based structural shape and topology optimization. *Computers & Structures*, 90:55–64.
- Yao, J., Kaufman, D. M., Gingold, Y., and Agrawala, M. (2017). Interactive design and stability analysis of decorative joinery for furniture. *ACM Transactions on Graphics (TOG)*, 36(2):20:1–20:16.
- Zhang, X., Le, X., Wu, Z., Whiting, E., and Wang, C. C. (2016). Data-driven bending elasticity design by shell thickness. *Computer Graphics Forum (Proceedings of Symposium on Geometry Processing)*, 35(5):157–166.
- Zhao, H., Xu, W., Zhou, K., Yang, Y., Jin, X., and Wu, H. (2016). Stress-constrained thickness optimization for shell object fabrication. In *Computer Graphics Forum*. Wiley Online Library.
- Zhou, Y., Kalogerakis, E., Wang, R., and Grosse, I. R. (2016). Direct shape optimization for strengthening 3d printable objects. In *Computer Graphics Forum*, volume 35, pages 333–342. Wiley Online Library.
- Zhu, B., Skouras, M., Chen, D., and Matusik, W. (2017). Two-scale topology optimization with microstructures. *ACM Trans. Graph.*, 36(4).
- Zhu, Y., Bridson, R., and Kaufman, D. M. (2017). Blended Cured Quasi-Newton for Geometric Optimization. *ArXiv e-prints*.