U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Extension of an industry-focused analytical tool for optimisation of composite crash structures (in collaboration with McLaren Racing)

Francisco Guimarães Vasconcelos Moniz Arriscado

Thesis submitted to Universidade do Porto – Faculdade de Engenharia
in partial fulfilment of the requirements for the degree of
Master in Mechanical Engineering

Supervisors:
Professor Dr. Albertino Arteiro
Dr. Denis Dalli
Dr. Rui Coelho

Mestrado em Engenharia Mecânica (M.EM)
Departamento de Engenharia Mecânica (DEMec)
Faculdade de Engenharia (FEUP)
Universidade do Porto (UP)

© Porto, June 2026

“Speed has never killed anyone. Suddenly becoming stationary, that’s what gets you.”

- Jeremy Clarkson

Ao meu avô,
à Targa,

Extension of an industry-focused analytical tool for optimisation of composite crash structures (in collaboration with McLaren Racing)

Abstract

Composite crash structures in high-performance applications such as Formula 1 are designed against a mandated energy absorption requirement, while every gram of structural mass carries a direct performance penalty. The crush force developed by a progressively crushing composite section depends strongly on the local curvature of its walls, which makes the cross-sectional shape a powerful design variable. This thesis develops a gradient-based optimisation framework that finds the lightest cross-sections able to carry a required crush force, using the perimeter as a direct proxy for mass, and building on the analytical crush prediction tool *CrushAnalytica* developed in previous FEUP work with McLaren Racing.

The optimisation module inherited from that previous work could only move each point of the boundary along a fixed direction from the centre of the section, so the section could shrink but never fold. Theory, however, predicts that the best section is one with alternating signed curvature, with walls held at the tightest radius that manufacturing allows, and such a shape lies outside what that description can represent. The problem was therefore reformulated with the curvature of the wall itself as the design variable. With this choice the manufacturing limit is respected by construction, and the optimiser is free to fold the section wherever that buys crush force at a lower perimeter. The crush force, closure and smoothness requirements are handled inside a fully differentiable implementation of the analytical model, so a gradient-based optimiser can reshape a section in minutes on a normal desktop computer.

The new formulation was first validated on single sections, where it converges to the wavy optimum from many different starting shapes, settling closely to the theoretical minimum perimeter. It was then extended to the complete Side Impact Structure, with all thirty-five cross-sections optimised together and additional mechanisms that keep neighbouring sections in the same shape family, so that the stacked sections form a smooth, manufacturable body rather than a collection of unrelated optima.

The final assembly meets every section crush force target with 14.1% less outer wall perimeter, and 24.7% less load-bearing wall overall. This translates into a mass saving of 0.205 kg, from 0.479 kg to 0.274 kg, and a 25.0% increase in specific energy absorption, while keeping the shape transitions between sections almost as smooth as those of the original production part. The optimised geometry is exported back as a finite element model and re-analysed by *CrushAnalytica*, which reads back the designed forces to within 1.4%, closing the loop between optimisation and verification.

Keywords: Composite materials, Crashworthiness, Energy absorption, Crash structures, Shape optimisation, Curvature parameterisation, Gradient-based optimisation, Automatic differentiation, PyTorch, Adam optimiser, Motorsport safety, Cross-section design, Progressive crushing

Extension of an industry-focused analytical tool for optimisation of composite crash structures (in collaboration with McLaren Racing)

Resumo

As estruturas de absorção de impacto em compósito utilizadas em aplicações de alto desempenho, como a Fórmula 1, são projetadas para cumprir um requisito regulamentar de absorção de energia, ao mesmo tempo que cada grama de massa estrutural penaliza diretamente o desempenho. A força de esmagamento desenvolvida por uma secção compósita em esmagamento progressivo depende fortemente da curvatura local das suas paredes, o que torna a forma da secção transversal uma variável de projeto poderosa. Esta tese desenvolve uma metodologia de otimização baseada em gradientes que encontra as secções mais leves capazes de suportar uma força de esmagamento exigida. A previsão de esmagamento é feita pela ferramenta analítica *CrushAnalytica*, desenvolvida em trabalhos anteriores da FEUP com a McLaren Racing.

O módulo de otimização herdado do trabalho anterior apenas conseguia mover cada ponto da fronteira ao longo de uma direção fixa. A teoria, porém, prevê que a melhor secção é uma secção com curvatura com sinal alternado, com as paredes mantidas no menor raio fabricável, e essa forma está fora do que aquela descrição consegue representar. O problema foi por isso reformulado tomando a própria curvatura da parede como variável de projeto. Os requisitos de força e a imposição de secções fechadas e com transições suaves são tratados dentro de uma implementação totalmente diferenciável do modelo analítico, pelo que um otimizador baseado em gradientes consegue redesenhar uma secção em minutos num computador de secretária comum.

A nova formulação foi primeiro validada em secções isoladas, onde converge para o ótimo com sinal de curvatura alternada a partir de muitas formas iniciais diferentes, aproximando-se do perímetro mínimo teórico. Foi depois alargada à Estrutura de Impacto Lateral completa, com as suas trinta e cinco secções otimizadas em conjunto e com mecanismos adicionais que mantêm as secções vizinhas na mesma família de formas, de modo a que as secções empilhadas formem um corpo suave e fabricável.

A montagem final cumpre o alvo de força de esmagamento de todas as camadas com menos 14,1% de perímetro na parede exterior e menos 24,7% de parede resistente no total. Isto traduz-se numa redução de massa de 0,205 kg, de 0,479 kg para 0,274 kg, e num aumento de 25,0% na absorção de energia específica, mantendo as transições de forma entre camadas quase tão suaves como as da peça original. A geometria otimizada é reexportada como modelo de elementos finitos e reanalisada pelo *CrushAnalytica*, que recupera as forças de projeto com um desvio de 1,4%, fechando o ciclo entre otimização e verificação.

Palavras-chave: Materiais compósitos, Resistência ao impacto, Absorção de energia, Estruturas de colisão, Otimização geométrica, Parametrização por curvatura, Otimização baseada em gradientes, Diferenciação automática, PyTorch, Otimizador Adam, Segurança no desporto automóvel, Projeto de secção transversal, Esmagamento progressivo

Acknowledgements

To begin with, I would like to thank Professor Dr. Albertino Arteiro for allowing me to undertake this interesting and challenging research project. I am also grateful for the time he set aside from his busy schedule for our meetings, and for the guidance he provided throughout this thesis.

Likewise, I would like to express my gratitude to Dr. Denis Dalli, the first person to introduce me to this project. He helped me get it off the ground, cleared up many of my doubts, and was very patient throughout. He helped me steer the work in the right direction and shared valuable insight into this type of application.

I would also like to thank Dr. Rui Coelho for lending me his expertise on the optimisation side of the work. His ideas and suggestions helped focus the project and proved an important contribution to the final result.

I would like to extend my thanks to all the professors who, throughout my time at FEUP, set an example through their dedication and knowledge of what a true professional should aim to be.

I would also like to thank my parents, Lurdes and Filipe, my brother Tomás, and the rest of my family. Thank you all for standing by me through these long years, and for all the support you have given me.

Last but not least, a special thank you to my late grandfather, Dr. Damião Arriscado, and my godmother, Alexandra Vangelder, who supported me the most throughout the entire process of earning my degrees. Without them I surely would not be where I am today, and for that I am deeply thankful.

To all of you, a sincere thank you.

Sustainable Development Goals

This work is framed within the 2030 Agenda for Sustainable Development of the United Nations and its seventeen Sustainable Development Goals (SDGs), to which the University of Porto is institutionally committed.

The thesis contributes most directly to SDG 3 (Good Health and Well-Being): crashworthy composite structures exist to protect human life in high-energy impacts, and the design tools developed here support target 3.6, which calls for a substantial reduction in deaths and injuries from road traffic accidents, since safety technology matured in motorsport is routinely transferred to road vehicles. It also aligns with SDG 9 (Industry, Innovation and Infrastructure), by advancing an industry-focused analytical and optimisation methodology that shortens the design cycle of advanced composite structures, and with SDG 12 (Responsible Consumption and Production), since replacing part of the experimental and finite element campaign with fast analytical prediction, and tailoring each cross-section to its load, reduces the material, energy and computational resources consumed during development.

Contents

List of Figures xvii

List of Tables xxi

List of Code Listings xxiii

Nomenclature xxv

1 Introduction 1

1.1 Context . . . 1
1.2 Motivation . . . 2
1.3 Objectives . . . 3
1.4 Thesis outline . . . 4

2 Literature Review 5

2.1 Crashworthiness of composite structures . . . 5
2.2 Parameters and performance metrics . . . 6
2.2.1 Crush stress . . . 7
2.2.2 Total energy absorption . . . 7
2.2.3 Specific energy absorption . . . 7
2.2.4 Peak and average force . . . 8
2.2.5 Crush efficiency . . . 9
2.3 Composite fracture and crushing modes . . . 9
2.4 Crush stress dependencies . . . 12
2.4.1 Curvature . . . 12
2.4.2 Inclination . . . 13
2.4.3 Laminate thickness . . . 14
2.4.4 Strain rate . . . 14

2.5 Optimisation of crashworthy structures . . . 14
2.5.1 Surrogate-based optimisation . . . 15
2.5.2 Topology optimisation . . . 16
2.5.3 Configuration design and emerging methods . . . 16
2.6 Structural shape optimisation . . . 17
2.6.1 Boundary parameterisations . . . 17
2.6.2 Curvature as a design variable . . . 18
2.7 Gradient-based optimisation and automatic differentiation . . . 19
2.7.1 The Adam optimiser . . . 19
2.7.2 Penalty formulations of the constraints . . . 20
2.7.3 Non-convexity and basins of attraction . . . 20
2.8 Concluding remarks . . . 21

3 Analytical Framework and Problem Definition 23
3.1 The CrushAnalytica force model . . . 23
3.1.1 Crush stress law . . . 24
3.1.2 Force integration . . . 25
3.1.3 Curvature reconstruction in CrushAnalytica . . . 26
3.2 Reference structure: the Side Impact Structure . . . 27
3.3 Problem statement . . . 31
3.4 Theoretical limits and the expected optimal shape . . . 32
3.5 Correction of a partitioning cross-section artefact . . . 33
3.6 The inherited radial optimiser . . . 37
3.6.1 Geometric representation . . . 37
3.6.2 Differentiable force evaluation . . . 38
3.6.3 Objective and optimisation . . . 39
3.7 Refinements to the inherited optimiser . . . 39
3.7.1 Summary of changes . . . 39
3.7.2 Results . . . 40
3.7.3 Remaining limitations and the need for a new optimiser . . . 42

4 Direct Curvature Formulation 45
4.1 Design variables . . . 46
4.2 Geometry reconstruction . . . 48

4.3 Constraints and loss function . . . 50
4.4 Optimisation strategy . . . 50
4.5 Characterisation of the formulation . . . 51
4.5.1 The saturated optimum . . . 51
4.5.2 Comparison with the radial optimiser . . . 53
4.5.3 Curve discretisation . . . 54
4.5.4 Iteration budget . . . 55
4.5.5 Force capacity . . . 57
4.5.6 Convergence to a global optimum . . . 58
4.6 Application to the SIS cross-sections . . . 59

5 Optimisation of the Full Structure **65**
5.1 Simultaneous optimisation of the multiple cross-sections . . . 65
5.2 Basins of attraction and consistency between sections . . . 67
5.3 Single-family warm-starting . . . 68
5.4 Aligning the stack to the reference structure . . . 70
5.5 Requirements for validation . . . 72
5.5.1 The closed design loop . . . 72
5.5.2 Building the finite element model . . . 72
5.6 Concluding remarks . . . 75

6 Results and Validation **77**
6.1 Final optimised assembly . . . 77
6.2 Verification through CrushAnalytica . . . 80
6.3 Generated finite element models . . . 82
6.4 Computational performance . . . 86
6.5 Discussion . . . 86
6.5.1 Distance to the theoretical bound . . . 86
6.5.2 Limitations . . . 87

7 Conclusions and Future Work **89**
7.1 Conclusions . . . 89
7.2 Future work . . . 91

References **93**

A Optimiser Command-Line Reference **101**

A.1 Command-line options 101

A.2 Reproduction commands 103

B Seed Shape Library **105**

B.1 The seed families 105

B.2 Per-seed convergence runs 109

C Data Interfaces of the Design Loop **113**

C.1 Partition data: `BigArray.mat` 113

C.2 Optimised assembly: the optimiser `.mat` 114

D Per-Section Result Tables **115**

List of Figures

2.1 Example force-displacement curve of an axial crush test. 6

2.2 Fracture mechanisms at the crush front of a progressively crushing composite wall. 10

2.3 The two primary macroscopic crushing modes of a fibre-reinforced composite wall. 11

2.4 CFRP tubular coupons of increasing radius of curvature, crushed quasi-statically. 13

2.5 Influence of inclination angle on failure modes. 14

3.1 CrushAnalytica pipeline, from a meshed structure to the force-displacement response of the crushed component. 24

3.2 Crush stress as a function of wall radius of curvature for a representative woven CFRP laminate. 25

3.3 Force integration over a discretised cross-section. 26

3.4 Side-view sketch of the Side Impact Structure. 27

3.5 Upper and lower SIS inside the side-pod of an F1 car. 27

3.6 Two views of the Side Impact Structure partitioned sections assembled in 3D. . 28

3.7 Cross-section of the SIS, showing the branched topology. The stadium-shaped outer wall (4.2 mm, 14 plies) is divided by an internal rib (1.2 mm, 4 plies). . 29

3.8 Force-displacement curve obtained from CrushAnalytica, crushing direction from outbound to inbound end. 29

3.9 Original SIS design envelope. 30

3.10 Loft kink-angle procedure. (a) A node index i is tracked across three consecutive partitions, giving $\boldsymbol{P}_{k-1,i}$, $\boldsymbol{P}_{k,i}$ and $\boldsymbol{P}_{k+1,i}$. (b) The kink angle $\theta_{k,i}$. The shape change is exaggerated for clarity. 31

3.11 Perimeter as a function of target crush force for different ply counts at $R_{\min}$ = 15 mm. 33

3.12 An optimised elliptical section, shown against its smooth initial shape. 34

3.13 Partitioned SIS cross-section at $z = -132.4$ mm. 34

3.14 Partitioned SIS cross-section at $z = 197.6\,\text{mm}$. 35
3.15 How the partitioning algorithm plants a false node. 36
3.16 Radial boundary parameterisation. 38
3.17 Minimum-perimeter results of the refined radial optimiser for the square and stadium seeds of Table 3.1. 40
3.18 Minimum perimeter optimisation of the same elliptical seed before and after the refinements. 41
3.19 Radius of curvature profile around the optimised Stadium section (log scale). 42
3.20 Representational limit of a radial description. 42

4.1 From design variables to crush force in the radial formulation (left, Section 3.6.2) and in the direct curvature formulation of this chapter (right). 45
4.2 Constructing the design vector from a starting shape, worked on a circle of radius 30 mm cut into $N = 8$ segments. 47
4.3 Reconstructing the boundary from the design vector, the inverse of Figure 4.2, on the same circle of radius 30 mm with $N = 8$ segments. 49
4.4 Optimised single section on the stadium seed, with initial and final geometry, and curvature profiles. 52
4.5 Loss landscape of the stadium run, a 2D contour with the descent path. 52
4.6 Loss landscape of the stadium run, 3D surface. 53
4.7 Final perimeter against the number of segments N on the stadium seed, with the continuous lower bound $L_{lb} = 215.20\,\text{mm}$. 55
4.8 Run time against the number of segments N on the stadium seed. 55
4.9 Final perimeter against the number of Adam steps on the stadium seed ($N = 128$), with the continuous lower bound $L_{lb} = 215.20\,\text{mm}$. 56
4.10 Run time against the number of Adam steps on the stadium seed ($N = 128$). . 56
4.11 Force target sweep, stadium seed ($N = 128$). 57
4.12 Multi-basin landscape and basin-hopping. 58
4.13 Final perimeter by seed ($N = 128$, $F_t = 75\,\text{kN}$, 100 000 steps). 59
4.14 Seed extraction, stage 1: the original FE partition of a ribbed section at $z = -132.396\,\text{mm}$. 60
4.15 Seed extraction, stage 2: the rib (18 nodes) and the near-colinear short-edge nodes (2) flagged for removal. 60
4.16 Seed extraction, stage 3: the cleaned loop smoothed, resampled to N segments of equal arc length and centred, giving the optimiser seed. 61

4.17 Section 22 and section 23 optimised shapes, showcasing the difference between two acceptable solutions. 63
4.18 Isometric view of the stacked sections. 63

5.1 Multi-section seed-extraction and optimisation pipeline. 66
5.2 Distinct curvature lobe families in an uncoupled run. 67
5.3 Curvature sign of the whole stack: rows - sections along the crush axis; columns - arc-length fraction. 70
5.4 Aligning an optimised section to the reference structure. 71
5.5 Alignment of the optimised SIS structure: (a) centre alignment; (b) side alignment, with the left face on the reference surface. 71
5.6 The closed design loop around the analytical tool: from the original finite element model to a verifiable optimised one. 72
5.7 Circumferential mesh refinement. Intermediate nodes are placed on the true arc implied by the stored curvature (b), not by subdividing the straight chord between the nodes of the optimiser itself (a). 73
5.8 Abaqus section manager resulting from running the script, alongside the properties of one section. 74
5.9 Abaqus section assignment example from running the script. 75

6.1 Original (a) and final optimised (b) Side Impact Structure cross-sections, coloured by wall thickness, view from outbound end. 78
6.2 Original (a) and final optimised (b) Side Impact Structure cross-sections, coloured by wall thickness, view from inbound end. 79
6.3 CrushAnalytica output for the same section in the different assemblies: (a) original SIS structure; (b) final assembly. 79
6.4 Force-displacement response read by CrushAnalytica for the original structure (a) and for the final optimised assembly (b). 81
6.5 Finite element shell model of the final optimised assembly in Abaqus/CAE: (a) isometric view; (b) XY view; (c) XZ view; (d) YZ view. 84
6.6 Progression of the structure across the optimisation campaign: (a) original structure; (b) independently optimised stack; (c) final delivered assembly. . . 85

B.1 Seed shape library, each family shown with its boundary (left) and its initial signed curvature profile $\kappa(s)$ (right). The dashed lines on the curvature panels mark the manufacturing bound $\pm 1/R_{\min}$. 106

B.2 Per-seed convergence runs: optimised cross-section against the initial outline (left) and optimised signed curvature profile $\kappa(s)$ (right). The dashed lines on the curvature panels mark the manufacturing bound $\pm 1/R_{\min}$. 109

D.1 Per-section optimisation results, sections 1 to 12: extracted seed outline (black) and optimised boundary (dark red); the dashed rectangle marks the seed bounding box. 117

D.2 Per-section curvature profiles, sections 1 to 12: initial seed (black) and optimised section (dark red), as the normalised curvature $\kappa R_{\min}$ against segment index; the dashed lines mark the manufacturing bound ± 1. 120

List of Tables

3.1 Minimum perimeter results of the refined radial optimiser for three seed shapes. 41

4.1 Adam optimiser settings tuned for the curvature formulation. The remaining hyperparameters (β_1, β_2, ϵ and the weight decay) are kept at their library defaults. 51

4.2 Radial versus curvature optimiser on the stadium seed at the matched target ($F_t \approx 70$ kN, $t = 2.4$ mm, $R_{min} = 15$ mm). The continuous lower bound on the perimeter is $L_{lb} = 200.85$ mm. 54

4.3 Seed convergence study results ($N = 128$, $F_t = 75$ kN, 100 000 steps). 59

4.4 Results of the independent single-section optimisation of the SIS partitions. Sections are ordered from the outbound end (section 1) to the inbound end (section 35). 62

5.1 Maximum kink between adjacent sections of the assembled stack (before) and of the final delivered stack (after). Sections run from the outbound end (section 1) to the inbound end (section 35); the kink is undefined at the two end sections. 69

5.2 Mesh convergence of the CrushAnalytica force on a representative optimised section, exported at circumferential refinement factors $r = 1$ to 5. Deviations are measured against the finest mesh ($r = 5$). 73

6.1 The full structure result across the optimisation campaign. Perimeter reductions are quoted against the total load-bearing wall length (8430.6 mm). 78

6.2 Closed loop force check: the re-analysis by CrushAnalytica of the generated model against the force accounting of the optimiser itself. 80

6.3 CrushAnalytica readings of the original SIS and of the final optimised assembly. Crushed mass, absorbed energy and specific energy absorption are the totals of the tool itself for each run. 80

6.4 Mesh properties of the generated finite element model of the final assembly. . 82

6.5 Mass properties of the finite element meshes, extracted with the Abaqus mass properties query. 83

6.6 Wall-clock cost of the definitive optimisation runs (desktop CPU, eight threads). 86

C.1 Fields of `BigArray.mat` read by the seed extraction. n_k is the number of partition nodes of entry k. 113

C.2 Fields of the optimiser output `.mat`. N is the number of boundary segments per section (here $N = 128$). 114

D.1 Section results of the final optimised assembly. 116

List of Code Listings

A.1 Principal options of `rocOptimizer_v8.py` (`-help` output). 101
A.2 Final full-structure run (joint + family warm start). 103
A.3 Uncoupled sequential run (Chapter 6 comparison baseline). 103
A.4 Finite element model generation from the optimised `.mat`. 103
A.5 Scripted import into Abaqus/CAE. 103

Nomenclature

Abbreviations

AD	Automatic Differentiation
CA	CrushAnalytica (analytical crush-prediction tool)
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CFRP	Carbon Fibre Reinforced Polymer
CPU	Central Processing Unit
EA	Energy Absorption
F1	Formula 1
FE	Finite Element
FEM	Finite Element Model
FEUP	Faculdade de Engenharia da Universidade do Porto
FIA	Fédération Internationale de l'Automobile
GPU	Graphics Processing Unit
RMS	Root Mean Square
RoC	Radius of Curvature
S4R	four-node reduced-integration shell element (Abaqus)
SEA	Specific Energy Absorption
SIS	Side Impact Structure
VEA	Volumetric Energy Absorption

Symbols (Latin)

A	cross-sectional area of a wall segment
a	donor blend fraction of the family warm start
c_x, c_y	closure residuals of the reconstructed boundary
c_θ	turning residual of the reconstructed boundary
$\mathrm{d}s$, Δs	arc-length element / segment length
F	crush force
F_{avg}	average (steady-state) crush force
$F_{\max}$	peak crush force

F_0	crush force of the seed shape
F_t	target crush force
g	smoothed knot values of the smoothing spline
h_i	knot interval length of the smoothing spline
L	total arc length (perimeter) of the section boundary
L_{lb}	continuous lower bound on the perimeter
ℓ_i	nodal tributary length
ℓ_med	median edge length of a partition branch
M_i	second derivatives at the spline knots
m_c	crushed mass
N	number of arc segments in the discretised boundary
p	smoothing parameter of the Reinsch spline
P	perimeter
P_0	seed (initial) perimeter
$\boldsymbol{P}_{k,i}$	position of node i on partition k
$P_l^\star$	converged single-section perimeter of section l
P_min	minimum feasible perimeter at the force target
Q, R	Reinsch matrices of the smoothing spline
r	radial coordinate of the polar parameterisation
R	radius of curvature
R_flat	transition radius beyond which the wall is treated as flat
R_min	minimum manufacturable radius of curvature
s	arc-length coordinate along the boundary
t	wall thickness
u_i	unconstrained curvature design variables
$\boldsymbol{v}_{k,i}$	loft vector joining node i of partitions k and $k+1$
V_c	crushed volume
$\boldsymbol{x}_i$	position of boundary node i
z_k	coordinate of partition k along the crush direction

Symbols (Greek)

δ	crush displacement
δ_1, δ_2	start and end displacement of the progressive crush stage
δ_max	maximum crush displacement
η	learning rate of the Adam optimiser
η_c	crush efficiency
θ	segment heading angle
θ_0	initial heading of the first segment
$\theta_{k,i}$	loft kink angle at node i of partition k
θ_max	maximum loft kink angle (manufacturability allowable)

κ	signed curvature
λ	penalty weight (subscripts denote the penalised term)
ξ	dummy arc-length integration variable
σ_c	crush stress
σ_0	constant (baseline) crush stress
ϕ	polar angle
$\partial\Omega$	closed boundary curve of the cross-section

Chapter 1

Introduction

1.1 Context

Carbon-fibre-reinforced polymers (CFRP) are the primary structural material of modern high-performance vehicles. Their high specific stiffness and strength, that is, the stiffness and strength per unit of mass, let engineers meet structural requirements with far less weight when compared to metals. In Formula 1, a sport where every gram matters, this property of composite materials is extensively explored. This results in the chassis, the bodywork and most load-bearing components being manufactured in carbon-fibre composite materials. Composites also have a second property that matters for safety. Under crushing they do not fold and hinge like a ductile metal. Instead they fail progressively at a crush front, where the plies fragment, splay and delaminate while the rest of the structure stays intact, and in doing so, they dissipate a large amount of energy per unit of crushed mass [1, 2]. This combination of low mass and high energy absorption is what makes CFRP the natural choice for the crash protection of a racing car.

Crash safety in Formula 1 is built around the survival cell. This is a stiff carbon-fibre monocoque that surrounds the driver and must stay essentially intact in an accident. The survival cell does not absorb the impact itself. That task falls to dedicated energy-absorbing structures attached to it: the front impact structure ahead of the driver's feet, the side impact structures along the flanks of the car, and the rear impact structure behind the gearbox. Each of these is a thin-walled composite body designed to crush in a controlled way, converting the kinetic energy of an impact into fracture energy and decelerating the car within limits a human can survive [3]. The reference component used throughout this thesis is the Side Impact Structure (SIS), which protects the driver against lateral intrusion.

The performance of such a structure is described by the force it develops as it crushes. A well-designed crash structure sustains a roughly constant crush force over a long stroke, so that the force-displacement curve is a long plateau rather than a single sharp peak. The area under that curve is the energy absorbed. Two quantities summarise the behaviour. The mean

crush force is the sustained load over the stable stroke. The specific energy absorption (SEA) is the energy absorbed per unit mass. A high SEA at a controlled crush force is the aim of every crash-structure design.

These structures are not designed to a free target. They are designed to pass regulation. The Fédération Internationale de l'Automobile (FIA) homologates each impact structure through mandatory crash tests, in which a structure of prescribed geometry is loaded at a prescribed energy and must absorb it while keeping the peak force and the average deceleration below set limits [4]. A structure that fails the test cannot race. For the designer this turns crashworthiness into a hard constraint rather than a soft objective: the structure must reach a specified crush force over its stroke, no less.

At the same time, mass is the currency of performance. A lighter car accelerates, brakes and corners better, and although a minimum weight is itself set by regulation, every gram of structure built above what the rules demand is a gram that cannot be placed where it helps the car most. A crash structure therefore must aim to balance two things. It must carry enough material to develop the regulatory crush force, and it must carry as little material as possible to avoid the mass penalty. The ideal crash structure is the lightest one that still passes the imposed crush force regulations.

This defines the engineering problem addressed in this thesis. For a given composite material, properly defined, the crush force a section develops is set by its cross-sectional shape, and in particular by the local curvature of its walls. The shape of the cross-section is thus not a fixed input but a design variable. The rest of this chapter explains why a fast analytical crush model makes it possible to optimise that shape directly, and sets out the objectives of the work.

1.2 Motivation

This work builds on analytical crush modelling previously developed at FEUP with McLaren Racing, developed by Cunha and later extended by Cardoso into the tool *CrushAnalytica* [5, 6]. It predicts the crush force of a composite structure directly from its partitioned cross-sections, returning the force-displacement curve of a crush test in far less time than an equivalent finite element crash simulation: seconds to a few minutes for the analytical tool, against several hours for the FE model.

The tool takes into account several quantities in its material characterisation, namely the curvature, inclination, ply count and crushing velocity at each node, following the work of Cunha [5]. Among these, curvature has the greatest effect on the crush stress. The crush stress of a wall rises sharply as its local radius of curvature decreases. It would be highest at the tightest possible radius, but such a radius cannot be manufactured, so a lower limit is imposed on the radius of curvature. A section that holds its walls at this lower limit develops the most crush force per unit of perimeter, and so reaches the required force with the least material.

So far the analytical model has been used to obtain a fast approximation of the force-displacement curve in the early design stages. This points to an opportunity. The same model,

accurate enough for early design, can also drive an optimisation of the structure on the same timescale as the force prediction, identifying from the start sections of a design that can be improved.

The loop between fast prediction and automatic design has, however, remained open. This thesis closes it. The force model is re-implemented in a differentiable form, so that the crush force can be differentiated with respect to the section shape. A gradient-based optimiser uses these gradients to reshape the section towards minimum perimeter and, consequently, less mass while preserving the required force. The approach scales to the dozens of stacked sections that make up a full structure, turning the analytical model from a calculator into a design tool of direct value to a Formula 1 team.

1.3 Objectives

This work is guided by the following research question: given a fast analytical crush model, can the cross-section of a composite crash structure be reshaped, by gradient descent, into the lightest closed section that still develops its required crush force, and can this be done for the complete multisection structure on the same timescale the analytical tool operates on?

The primary objective of this thesis is to develop a gradient-based cross-section optimisation framework for composite crash structures that finds minimum-perimeter (minimum-mass) closed sections subject to a target crush force, using the CrushAnalytica force model as the physics reference.

Primary objectives:

- Formulate the minimum-perimeter problem with the crush-force requirement, manufacturing curvature bound, and geometric closure constraints, and solve it with a differentiable PyTorch implementation of the analytical force model;
- Identify a boundary parameterisation capable of representing the true optimum (shown to require corrugated, alternating-curvature walls) and demonstrate convergence to it from independent seed shapes;
- Scale the single-section optimiser to the complete Side Impact Structure, optimising all partition sections simultaneously while enforcing inter cross-section geometric continuity for manufacturability.

Supplementary objectives:

- Verify the optimised geometry by closing the loop through CrushAnalytica and quantifying the reconstruction error of its curvature pipeline;
- Export optimised geometries to Abaqus finite element models for future numerical and experimental validation;

- Improve the robustness of the underlying analytical tool where the optimisation work exposed defects (cross-section partitioning artefacts).

1.4 Thesis outline

The present thesis is organised into seven chapters:

Chapter 2: Literature Review. Reviews the crashworthiness of composite structures with emphasis on the geometric dependencies of crush stress, analytical crush-prediction approaches, and the shape-optimisation and gradient-based optimisation methods on which this work builds.

Chapter 3: Analytical Framework and Problem Definition. Presents the CrushAnalytica force model and the reference Side Impact Structure, formally states the minimum-perimeter optimisation problem, derives theoretical bounds that shape the expected optimum, and documents the corrections made to the baseline tool. It also presents the inherited radial optimiser (Section 3.6): the baseline module from previous FEUP work, based on a polar boundary parameterisation and a differentiable smoothing-spline curvature pipeline, with its results and the structural limitation that motivated the curvature reformulation.

Chapter 4: Direct Curvature Formulation. Develops the curvature-based design parameterisation, the penalty treatment of the closure, turning, and force constraints, the optimisation strategy, and the single-section results including seed-independence studies.

Chapter 5: Optimisation of the Full Structure. Extends the formulation to the complete multi-section structure, describes the loft smoothness manufacturability criterion and the geometric post-processing, and closes the design loop with the automatic generation of Abaqus finite element models from the optimised geometry.

Chapter 6: Results and Validation. Presents the final optimised assembly, its verification through CrushAnalytica, the generated finite element models, and the computational performance of the framework.

Chapter 7: Conclusions and Future Work. Synthesises the findings, assesses the objectives, and identifies directions for future research.

Chapter 2

Literature Review

This chapter provides a comprehensive review of the state of the art in the use of composite materials for crashworthy structures, designed to absorb energy during crushing events. The chapter first reviews the physics of composite crushing and the metrics used to assess it: the performance parameters derived from the force-displacement curve, the fracture and energy dissipation mechanisms involved, and the macroscopic crushing modes. It then examines how crush stress depends on the geometry of the wall, in particular its radius of curvature, which is the dependency this thesis exploits. It then surveys the current state of the art in the optimisation of crashworthy structures, and how the field copes with the cost of the crush simulation. Finally it frames the design task as a structural shape optimisation problem and reviews the gradient-based methods used to solve it, closing with the gap in the literature that this work addresses.

2.1 Crashworthiness of composite structures

Crashworthiness is the ability of a structure to protect its occupants by absorbing kinetic energy in a controlled way during an impact [7]. These structures, in the specific case of automotive and aeronautic industries, have an extremely important function. By maintaining the integrity of the occupant compartment and ensuring tolerable deceleration levels for the passengers, they represent essential safeguards responsible for reducing the risk of injuries and deaths in accidents [8, 9].

The pursuit of lightweight and safe structures has been intensified in recent years by the regulations on sustainability and concerns on efficiency, as reductions in the weight of vehicles translate directly into improved fuel consumption [10]. This incentive has driven the evolution of materials employed in crashworthy structures, from traditional steels, to aluminium alloys that offer lower costs and higher specific energy absorption [11], and, more recently, to advanced composites such as carbon-fibre-reinforced polymers (CFRP). CFRPs, due to their exceptionally high strength-to-weight and stiffness-to-weight ratios, and distinct

fracture mechanisms, can achieve higher specific energy absorption than metals, enabling the development of lighter and safer components, although with a higher cost [12].

These great properties of fibre reinforced polymers have led to widespread use of thin walled composite structures in aeronautic, automotive and high-performance applications such as motorsport. In Formula 1 (F1), for instance, where weight optimisation is a critical aspect that highly influences lap-time performance, the Fédération Internationale de l'Automobile (FIA) enforces stringent safety regulations to protect the drivers. To this end, the monocoque survival cell is employed in F1 cars, which integrates multiple energy-absorbing structures, such as the front, side and rear impact structures, whose objective is to dissipate the large amounts of energy through composite fracture mechanisms [13, 14]. Regarding aeronautics, an important development of crashworthy structures was the introduction of the Military Standard MIL-STD-1290, issued by the United States Department of Defense in 1988 [15]. This standard defined the design requirements for the crash resistance of rotary-wing aircraft, aiming to maximise occupant survivability in crash scenarios. To achieve this, features such as energy-absorbing landing gear, crashworthy fuel systems to mitigate post-impact fire hazards, and structural layouts that maintain a survivable volume around the crew and passengers should be adopted [16]. Building upon these principles, modern crashworthiness regulations issued by the FAA [17–19] and EASA [20–22] have evolved from military origins into comprehensive civilian certification frameworks, establishing stringent requirements for both fixed-wing aircraft and rotorcraft.

2.2 Parameters and performance metrics

Crashworthiness is assessed experimentally through axial crushing tests, carried out either quasi-statically or dynamically. The force-displacement curve recorded during such a test, of which Figure 2.1 is a representative example, is the basis from which the performance parameters are derived. These parameters include the deceleration imposed on the structure and the level of energy absorbed. Both are needed to confirm that a component is correctly designed and able to protect the occupants during an impact.

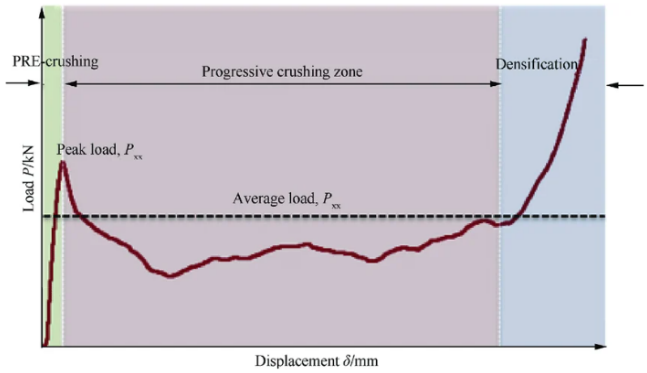


Figure 2.1: Example force-displacement curve of an axial crush test, showing the initial peak force and the pre-crush, progressive crush and densification stages (adapted from [23]).

2.2.1 Crush stress

Crush stress is a central parameter for judging how well a material absorbs energy under a crushing load. Other metrics, introduced below, are also used to compare materials, but crush stress is particularly informative because it reflects the capacity to dissipate energy through mechanisms such as damage, fracture and disintegration [1, 24]. It is not strictly a material property, since the same material develops different values under different conditions, yet it remains a useful indicator of performance. The factors behind this variability are reviewed in Section 2.4.

Because of its relevance, crush stress is often supplied as an input to numerical and analytical crush models. It follows directly from the force-displacement curve as the ratio of the force at a given displacement, $F(\delta)$, to the cross-sectional area under compression, $A(\delta)$, as written in Equation (2.1).

$$\sigma_{\text{crush}}(\delta) = \frac{F(\delta)}{A(\delta)}. \tag{2.1}$$

2.2.2 Total energy absorption

The total energy absorption (EA) is the area under the force-displacement curve (Figure 2.1). It quantifies the total energy dissipated during a crush test [24]. It is given by Equation (2.2),

$$EA = \int_0^{\delta_{\max}} F(\delta)\,\mathrm{d}\delta, \tag{2.2}$$

where δ is the displacement of the crushing device, $F(\delta)$ is the force measured at that displacement and $\delta_{\max}$ is the maximum displacement reached in the test.

The energy-absorption process can be divided into distinct stages, also indicated in Figure 2.1. The area under the first part of the curve corresponds to the energy absorbed during the pre-crush phase. This phase is governed by elastic material behaviour and ends with the initial peak force. Energy is then absorbed during the post-crush phase, also called the progressive crush stage, in which crushing propagates gradually. This stage is the most relevant for extracting the key metrics, excluding the peak force, and for analysing the underlying failure mechanisms. The final stage is densification, in which the force rises sharply and the crushing process ends.

2.2.3 Specific energy absorption

Specific energy absorption (SEA) is one of the most widely used metrics for assessing and comparing the crashworthiness of composite structures. It normalises the total absorbed energy by the mass of the structure, giving a measure of material or layup efficiency that suits the early design of lightweight crash structures [24, 25]. Physically it is the energy absorbed

per unit mass of the structure, and it is obtained from Equation (2.3),

$$SEA = \frac{1}{m_c}\int_0^{\delta_{\max}} F(\delta)\,\mathrm{d}\delta = \frac{EA}{m_c}, \tag{2.3}$$

where m_c is the crushed mass of the structure under analysis.

A related quantity is the volumetric energy absorption (VEA), the energy absorbed per unit volume. It is useful when the limiting constraint is packaging space rather than mass [24]. It is defined in Equation (2.4),

$$VEA = \frac{1}{V_c}\int_0^{\delta_{\max}} F(\delta)\,\mathrm{d}\delta = \frac{EA}{V_c}, \tag{2.4}$$

where V_c is the crushed volume of the structure.

2.2.4 Peak and average force

The peak force $F_{\max}$ is the highest force reached during crushing. It usually occurs at the onset of impact, before the structure settles into progressive crushing. This peak marks the end of the elastic response. In a vehicle it corresponds to the instant of greatest deceleration transmitted to the occupants. An excessively high peak force points to an overly stiff structure or to a geometric trigger that fails to initiate stable crushing, which can lead to unstable collapse [2]. Both material and geometry govern its magnitude. Without an effective triggering mechanism the peak force can rise sharply, increasing the deceleration and the risk of injury to the occupants. It is defined in Equation (2.5).

$$F_{\max} = \max[F(\delta)]\,. \tag{2.5}$$

The average force F_{avg}, also called the steady-state force, is the mean load carried by the structure during the progressive crushing stage. This stage follows the initial peak and shows a roughly constant force, although the response is often jagged because energy is released through successive micro-fracture events [2]. The average force reflects the ability of the material to sustain resistance throughout progressive crushing. It is found by integrating the force over displacement across the progressive crush stage, as in Equation (2.6),

$$F_{\text{avg}} = \frac{\displaystyle\int_{\delta_1}^{\delta_2} F(\delta)\,\mathrm{d}\delta}{\delta_2 - \delta_1}, \tag{2.6}$$

where δ_1 is the displacement at the start of the post-crush stage and δ_2 is the displacement at its end.

2.2.5 Crush efficiency

Crush efficiency η_c is a dimensionless metric, defined as the ratio of the average crush force F_{avg} to the peak force F_{max}, as written in Equation (2.7). It is a useful measure of how stable and progressive the crushing process is. A value close to unity implies a stable, progressive crush response, whereas a low value indicates a high initial peak force followed by a large drop in force, which is typical of unstable failure modes [3, 26]. Several factors, such as geometry, ply orientation and strain rate, can strongly influence this metric. To raise crush efficiency and limit the effect of high peak forces, crashworthy structures are designed with a gradually increasing cross-sectional area where the design allows.

$$\eta_c\,(\%) \;=\; \frac{F_{\text{avg}}}{F_{\text{max}}}. \tag{2.7}$$

2.3 Composite fracture and crushing modes

Metals absorb crush energy mainly through plastic deformation and progressive buckling [27]. Thin-walled fibre-reinforced polymers behave differently. They dissipate energy through fracture and damage so the crush response depends strongly on the material, the geometry and the test conditions [1, 2]. Efficient absorption requires a stable, progressive crush. Geometric instabilities such as buckling lead to premature collapse and to much lower energy dissipation [2].

Progressive crushing proceeds through several fracture mechanisms acting together at the crush front. These include interlaminar delamination, in both Mode I opening and Mode II shear, matrix cracking, fibre fracture, the bending of the separated plies into fronds, and frictional sliding between the fronds and the debris wedge [28] (Figure 2.2).

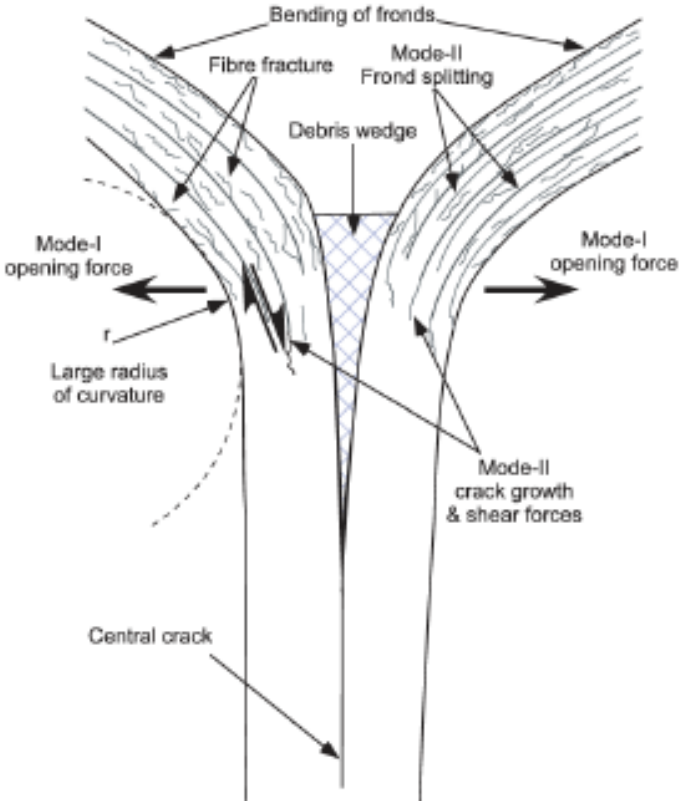
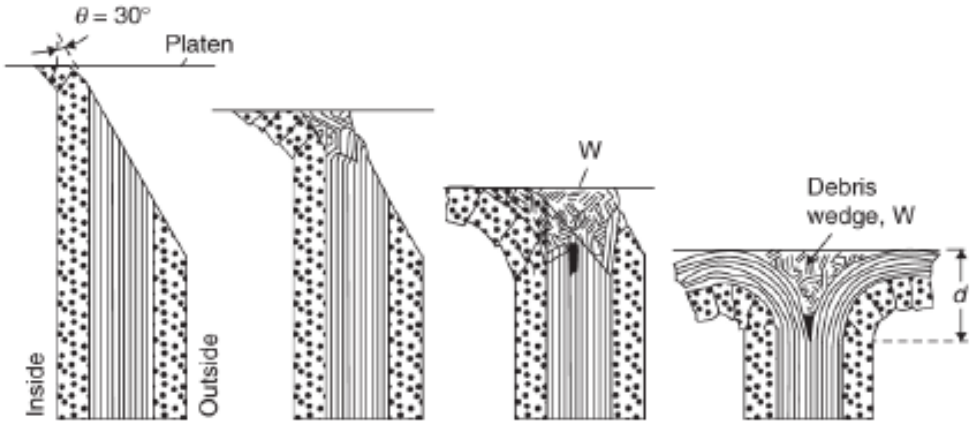


Figure 2.2: Fracture mechanisms acting together at the crush front of a progressively crushing composite wall: Mode I delamination, Mode II shear, matrix cracking, fibre fracture, frond bending and friction at the debris wedge (adapted from [28]).

The total energy absorbed is the sum of these individual contributions [24]. Their relative shares are not fixed. Reported splits differ widely between studies, because they depend on the material system, the geometry and the loading conditions, with no single distribution holding across all cases [29–31].

At the macroscopic scale this fracture activity organises into two primary crushing modes, identified in the foundational work of Hull [1]. The first is splaying, also called lamina bending. A central crack forms in the wall and propagates along the crush axis, delaminating the plies into fronds that bend inward and outward while the fibres fracture progressively. The second is fragmentation, also called transverse shearing. Interlaminar shear cracks break the wall into successively smaller fragments. Figure 2.3 illustrates the two modes. The two modes can act alone or together, and the balance between them is governed largely by the fibre architecture. Woven and high hoop-content laminates tend towards fragmentation, whereas laminates with fewer hoop fibres tend towards splaying [1, 25]. Only stable progressive crushing, in either mode, is useful for energy absorption, since unstable collapse dissipates far less [2].



(a) Splaying (lamina bending).

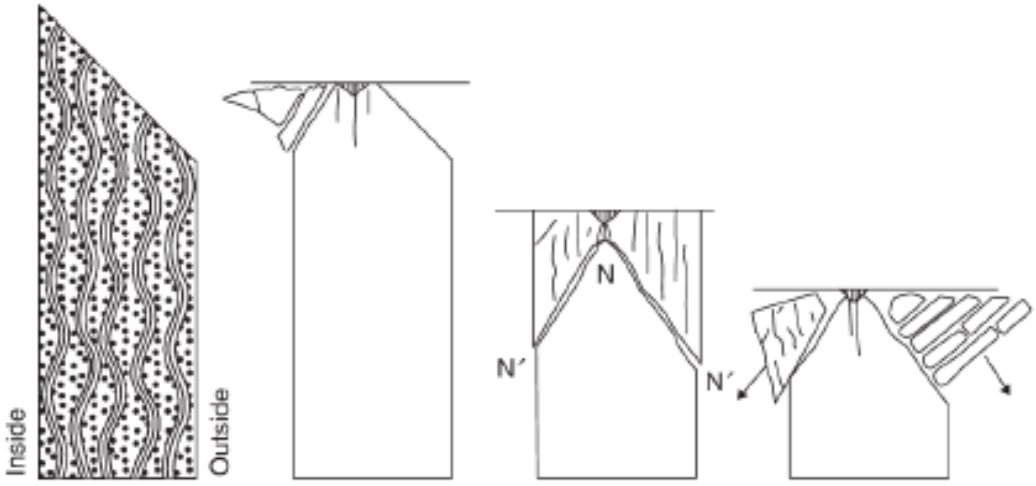


Figure 2.4: Illustration of fragmentation mode sequence (adapted from[39])

(b) Fragmentation (transverse shearing).

Figure 2.3: The two primary macroscopic crushing modes of a fibre-reinforced composite wall (adapted from [1]).

2.4 Crush stress dependencies

Crush stress is not an intrinsic material property. As noted in Section 2.2.1, the same laminate develops different sustained stresses depending on how it is crushed. The differences are set by the geometry of the wall and by the loading conditions, because these control the size and character of the fronds at the crush front, and therefore how the energy is split among delamination, bending, fibre fracture and friction. Numerous studies on both open and closed-section specimens confirm this [2, 6]. Understanding these dependencies is essential for any tool that predicts the response of an arbitrary section from its local geometry. This section reviews the four main dependencies, namely curvature, inclination, laminate thickness and strain rate, and then states the design implication that makes cross-sectional shape the main lever this thesis uses.

2.4.1 Curvature

Curvature is the main dependency explored in this thesis. The empirical result is that the sustained crush stress of a CFRP wall increases as its radius of curvature decreases. Two complementary lines of evidence support this.

The first comes from coupons designed to isolate curvature from the stabilising effect of a closed section. Feraboli developed self-stabilising corrugated specimens that characterise specific energy absorption without the anti-buckling fixtures that flat plates need [32]. When a carbon/epoxy square tube is cut into its corner and flat segments, the corner segments absorb much more energy than the flat segments, and the corner contribution grows as the corner radius is reduced [33]. This confirms that curvature itself, and not merely the presence of a closed section, raises crush stress.

The second line of evidence comes from tubular coupons of different radii. Lobão crushed five tubular CFRP geometries quasi-statically and found crush stress decreasing as the radius increased, tending towards the value of a flat coupon, which is the case of a tube with effectively infinite radius [34]. Moreira obtained consistent results across CFRP, flax-fibre and hybrid coupons of varying radius [35]. Figure 2.4 shows the recovered CFRP coupons across this range of radii. Both authors explained the trend from post-crushing observations. Smaller radii form narrower fronds, which require more fibre fracture per unit crushed length, while the closer internal fronds interact and add friction and shear, so more energy is absorbed [34, 35]. The effect depends on the material. It is strong for CFRP but weak or absent for some flax-fibre systems, where increasing the radius did not significantly change the peak load [9].

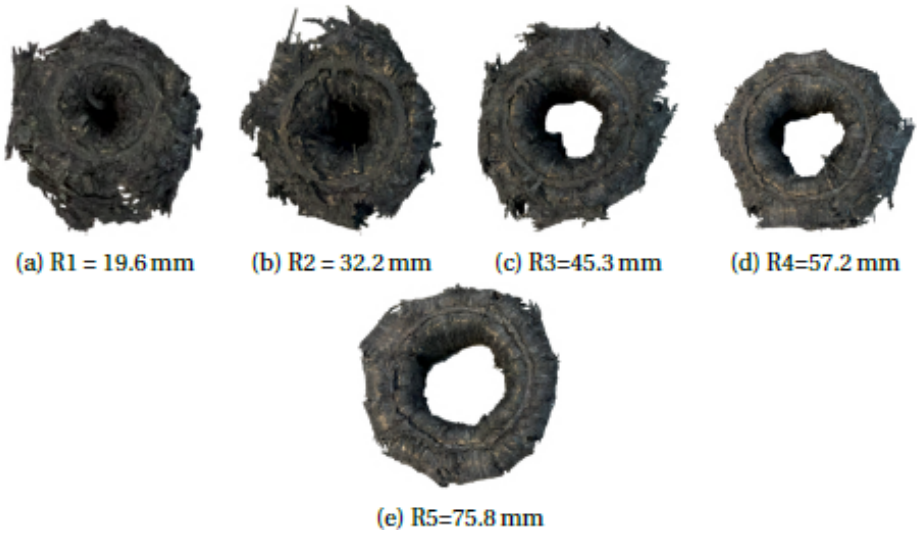


Figure 2.4: CFRP tubular coupons of increasing radius of curvature, crushed quasi-statically (adapted from [35]).

To model this behaviour an empirical crush stress law is used of the form

$$\sigma(R) \;=\; \sigma_0 \left(a\, R^b + c \right), \qquad b < 0, \tag{2.8}$$

where σ_0 is a reference flat-wall crush stress and R is the local wall radius of curvature.

With calibrated coefficients, a wall at the manufacturing curvature limit reaches about 1.6 times the flat-wall crush stress [6]. Hardware measurements support the law. Dalli reported that the low-radius regions of a Formula 1 side impact structure developed crush stresses of about 140 to 180 MPa, against 100 to 120 MPa in the flatter regions of the same component [3]. This is the law used throughout the present work, and its calibration is given in Chapter 3.

2.4.2 Inclination

Laminate inclination is the angle between the wall and the crushing axis. Its effect has been studied mainly with conical-shell coupons. Mamalis observed crush stress rising as the semi-apical angle increased from 5° to 15°, then falling at 20° [31]. The drop was attributed to a change of failure mode, from stable splaying to inward lamina bending with lower energy absorption (Figure 2.5). Moreira tested CFRP cones across a range of inclinations and confirmed a general reduction in crush stress with increasing inclination, with a pronounced decrease near 20° [35]. Inclination is therefore non-monotone and mode dependent.

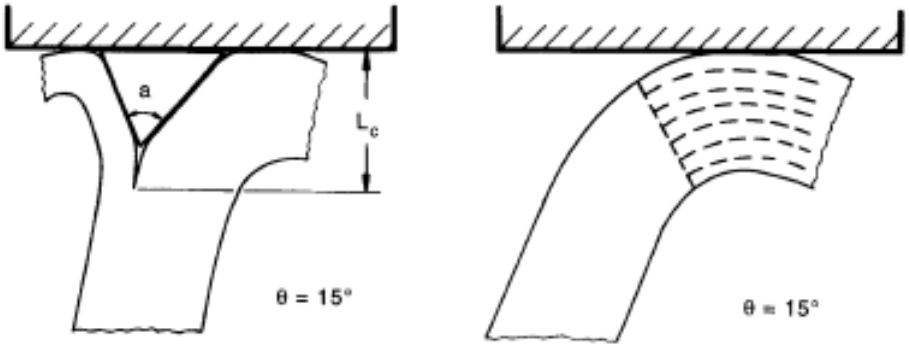


Figure 2.5: Influence of inclination angle on failure modes (adapted from [31]).

The analytical model used in this thesis can represent it, but the present optimisation treats the impact as axial and does not use the inclination dependence.

2.4.3 Laminate thickness

Wall thickness, set by the number of plies, is another strong dependency. Thicker walls resist the global and local buckling that triggers unstable collapse, favouring stable progressive crushing and raising the load carrying capacity [25, 36]. Tests on tubes and square coupons show that thickness itself, rather than geometric scaling, primarily governs the crush stress [36, 37], and that the effect is stronger for smaller-diameter specimens [9]. In the present framework thickness is held fixed within a cross-section. At fixed thickness the perimeter of the cross-section is a faithful proxy for mass, and it is the quantity minimised in this work.

2.4.4 Strain rate

Crush stress under dynamic loading usually differs from the quasi-static value, but literature is not consistent on the effects caused by strain rate. Some studies report higher specific energy absorption under dynamic conditions [31, 36], others a decrease [38], and others no significant change [39]. These discrepancies reflect the strain-rate sensitivity of the particular fibre-matrix system. Rate effects can also shift the dominant failure mode between splaying and fragmentation [36]. Yet, strain rate effects can be built onto the geometric law of Equation (2.8), and thus they do not bring any additional difficulty to the problem addressed here.

2.5 Optimisation of crashworthy structures

The dependencies reviewed above make clear that the crush response of a composite structure is not fixed by the material alone. It can be shaped by the geometry of the component. Turning this observation into a design tool is the subject of crashworthiness optimisation, a field that has

grown from experience and trial and error design into a formal discipline built on numerical simulation and mathematical optimisation [8]. This section reviews the state of the art. It first identifies the objectives and the obstacle common to all crashworthiness optimisation, then the three method families that the literature uses to overcome it, and finally the features these methods share, which motivate the different route taken in this thesis.

The design targets are the performance metrics of Section 2.2. A crashworthy component should absorb as much energy as possible, or reach a prescribed energy target, while keeping the peak force below the limit set by occupant tolerance and doing so at minimum mass. Crashworthiness optimisation is therefore usually posed as a multi-objective problem, in which specific energy absorption, peak crushing force and mass are traded against one another and the outcome is a Pareto front of compromise designs rather than a single optimum [8, 40].

The obstacle shared by every approach is the cost of evaluating a candidate design. The crush response is non-linear, transient and path dependent, so it is predicted with explicit dynamic finite element (FE) analysis. For composites the difficulty is increased because the constitutive model must reproduce the coupled intralaminar and interlaminar damage mechanisms that dictate progressive crushing. Building and calibrating such models is a research task in its own right [41], and it rests on extensive experimental characterisation of the material under crush loading [42]. A single high fidelity crush simulation can take hours, and its output is not readily differentiable with respect to the design variables.

2.5.1 Surrogate-based optimisation

The most common strategy is to replace the FE model with a cheaper approximation. A small set of designs is first generated using a design of experiments and then evaluated through simulation. The resulting input-output data is used to train a surrogate, or metamodel, such as a response surface, a Kriging model, or a radial basis function network. This surrogate can then predict the crush performance of new designs almost instantly. A population-based metaheuristic, most often a genetic algorithm, is then used to search the surrogate. In multi-objective problems, this is commonly done with the non-dominated sorting genetic algorithm (NSGA-II), which identifies the Pareto set [40]. This surrogate-plus-metaheuristic approach is widely used in crashworthiness studies. It has been applied to perforated CFRP tubes, where trigger geometry and layup are adjusted to balance specific energy absorption and peak force [40]; to the stacking sequence of an aircraft fuselage optimised for crashworthiness [43]; and to multilevel thin-walled absorbers, where a dedicated optimisation criterion is used to select between stage geometries [44]. Lian and co-workers optimised the lightweight and crashworthiness performance of an automotive front end with a stacked regression ensemble trained on crash simulations [45]. The main appeal of these methods is that they are gradient free and separate the search process from the expensive simulation. Their main limitation is that the surrogate is reliable only within the region covered by the sampled designs. As a result, the optimisation is usually restricted to the small number of scalar parameters chosen for the design of experiments.

2.5.2 Topology optimisation

A second group of methods gives the optimiser more freedom by allowing it to decide where material should be placed. Topology optimisation distributes material within a fixed design domain in order to improve a performance measure while satisfying a mass or volume constraint [46]. Its use in crashworthiness, however, is challenging. Density based and gradient driven methods work well for static stiffness problems because they rely on smooth sensitivities. Crush problems do not provide such smooth responses, since crushing is non-smooth and path dependent. For this reason, crashworthiness studies often use heuristic and evolutionary update schemes that do not require exact gradients. The hybrid cellular automaton method, for example, drives each element towards a uniform internal energy density and has become a standard approach for axial-impact problems [47]. Evolutionary structural optimisation has also been used to redesign an automotive crash box for frontal impact, by minimising compliance under a volume constraint and then validating the design in a full crash model [48]. The same heuristic approach is also found in recent extensions, including non gradient methods for dual material shell infill sections [49], multiscale optimisation of functionally graded lattice fills [50], and integrated topology optimisation of hybrid metal-composite tubes [51]. These methods greatly expand the design space. However, the material layouts they produce are not smooth, manufacturable wall boundaries, and the manufacturing limit on curvature is not included directly in the formulation.

2.5.3 Configuration design and emerging methods

A third, more empirical line of work improves crashworthiness by adding specific geometric features, rather than by carrying out a formal optimisation over continuous variables. Examples include crush initiators, functionally graded thickness or strength, multi-cell and bio-inspired cross-sections, and drilled triggering holes. These features are used to reduce the initial peak force and promote a more stable collapse. Ma and co-workers, for example, added triggering holes to composite/metal hybrid tubes and adjusted their size and position to reduce the peak crushing force while maintaining the absorbed energy [52]. Such studies usually combine a validated FE model with a parametric sweep or a small optimisation over the feature dimensions. Across all three families of methods, there is also a clear recent move towards machine learning and data-driven surrogates. This trend is motivated by the same need discussed throughout this section: to replace, or at least accelerate, the expensive crush simulation [45].

Two common features of this literature are important for the rest of the chapter. First, the design is usually described by a small set of scalar variables, such as wall thicknesses, cell counts, section dimensions, ply angles, or trigger positions. It is rarely described by the continuous shape of the load-bearing wall. Second, because the accurate forward model is expensive and non-differentiable, the optimisation is mainly gradient free and often relies on metaheuristics combined with surrogates. As a result, neither the continuous cross-sectional

boundary nor the constraint that is especially important in composite crush, namely the minimum manufacturable radius of curvature, is treated as a central part of the problem. This thesis follows a different route. It builds on a fast and differentiable analytical crush model, making gradient-based optimisation of the full cross-sectional boundary practical. The following sections introduce this framing.

2.6 Structural shape optimisation

The previous sections show that the crush force of a composite section depends not only on the material, but also on the shape of the wall. In particular, regions with small radius of curvature can develop higher crush stress than flatter regions. This makes the cross-sectional geometry an important design variable. If the same target force can be obtained with a smaller perimeter, then the structure can absorb the required energy with less material and therefore less mass.

Structural optimisation methods are commonly grouped by their design freedom. Topology optimisation redistributes material freely within a design domain [46] and has been applied to crashworthiness problems through heuristic update schemes [47]. The relevant class of the present problem is instead shape optimisation. Here the topology of the structure is fixed and only the shape of its boundary is changed. For the present work the section stays a closed, thin-walled profile, and the laminate thickness is held constant within each cross-section. The design freedom is therefore concentrated in the boundary of the cross-section. The task is to reshape this boundary so that the required crush force is kept while the perimeter, and hence the mass, is reduced.

A direct finite element approach would be too expensive for this task, because an optimiser must evaluate many candidate shapes. Instead, this thesis builds on the analytical crush-prediction work developed at FEUP. Cunha [5] introduced a method in which a three-dimensional impact structure is divided into cross-sections and the crush force of each section is estimated from its local geometry. Cardoso [6] later developed this approach into *CrushAnalytica*. In this tool, the local radius of curvature is used to assign a crush stress to each part of the wall, and these contributions are integrated around the section to obtain the cross-sectional force. For a Formula 1 side impact structure, the method matched the commercial CZone reference with an error of about 1%, while running around three orders of magnitude faster [6, 53]. This speed is important because it makes optimisation practical.

2.6.1 Boundary parameterisations

A cross-sectional boundary can be described in several ways. The simplest option is to move the Cartesian coordinates of the boundary nodes directly. This nodal approach is very general, because each point can move independently. However, it usually creates a large number of design variables. It can also produce jagged or irregular shapes unless additional smoothing is

introduced [54]. This is not ideal for a manufacturable composite component.

Another common option is a radial, or polar, description. In this case the boundary is written as a radius $r(\phi)$ measured from a reference point inside the section. This is close to the parameterisation used in the previous optimisation module inherited by this work. Each node was allowed to move mainly along a fixed direction from the centroid. The advantage is that the method is simple and naturally keeps the section closed. The disadvantage is that the shape is tied to the original angular distribution of the nodes. If the starting nodes are not well distributed, the optimiser has difficulty correcting the shape. More importantly, a radial description can only represent star-shaped sections, where each radial line intersects the boundary once. It cannot represent shapes that fold back in angle or contain strong corrugations.

Spline and CAD-based descriptions offer more flexibility. B-splines, Bézier curves and NURBS can describe smooth boundaries using a smaller number of control points [55]. Fourier descriptors provide another compact way of representing a closed curve [56]. These methods are useful in many shape optimisation problems, especially when smoothness and CAD compatibility are important. Their limitation in the present problem is that curvature remains an indirect result of the control points. A minimum manufacturing radius can be checked after the curve is reconstructed, but it is not naturally enforced by the parameterisation itself.

This distinction is central to the present thesis. The physical model reviewed in Section 2.4.1 makes curvature the main source of design leverage, but most conventional boundary descriptions treat curvature as a derived quantity. As a result, the most important constraint in the problem, the minimum radius of curvature, becomes difficult to impose directly. This motivates a different choice: using curvature itself as the design variable.

2.6.2 Curvature as a design variable

A planar curve can be reconstructed from its signed curvature distribution. If the curvature $\kappa(s)$ is known along the arc length s, the tangent angle is obtained by integration,

$$\theta(s) = \theta_0 + \int_0^s \kappa(\xi)\,\mathrm{d}\xi, \tag{2.9}$$

and the coordinates of the curve are then obtained by integrating the tangent direction,

$$\big(x(s), y(s)\big) = \big(x_0, y_0\big) + \int_0^s \big(\cos\theta(\xi), \sin\theta(\xi)\big)\,\mathrm{d}\xi. \tag{2.10}$$

This is the intrinsic, or natural, representation of a curve. It has a long history in curve design, where shapes are controlled through their curvature profile rather than through their coordinates [57–60].

The appeal of this representation for the present problem is direct. The manufacturing

limit is a limit on curvature. In intrinsic form it becomes a simple bound on the design variable,

$$|\kappa(s)| \leq \frac{1}{R_{\min}}, \tag{2.11}$$

where $R_{\min}$ is the smallest admissible manufacturing radius. This bound can be imposed directly. In the nodal, polar and spline descriptions, the radius of curvature is only known after the curve is reconstructed, so the same limit becomes an awkward nonlinear constraint.

The representation also has a cost. A curvature profile does not in general describe a closed curve. Closure must therefore be enforced separately, as a set of conditions on the curvature distribution. Conventional descriptions show the opposite balance: they keep the curve closed easily but make the curvature limit hard to impose. For a crush problem governed by a minimum radius, the intrinsic representation gives the more useful trade-off. The way this thesis enforces the curvature bound and the closure conditions is set out in Chapter 4.

2.7 Gradient-based optimisation and automatic differentiation

The analytical crush model is fast and is built from differentiable operations, such as spline fitting, the smooth crush-stress law and nodal integration. The curvature representation of Section 2.6.2 is also differentiable through its integral reconstruction. Both can therefore be differentiated automatically. This makes the design problem well suited to gradient-based optimisation. The gradients of the objective and the constraints can be computed exactly and cheaply, and a first-order optimiser can then drive the search. This section reviews the parts that make this possible: the optimiser (Section 2.7.1), the handling of constraints by penalties (Section 2.7.2), the management of non-convexity (Section 2.7.3), and the automatic differentiation that links them.

2.7.1 The Adam optimiser

Adam [61] is a first-order optimiser. It has become the default choice for high-dimensional, non-convex problems in machine learning, and is now common in differentiable engineering design. Adam tracks two running averages for each variable. The first is an average of the gradient, called the first moment. The second is an average of the squared gradient, called the second moment. At each step it divides the first moment by the square root of the second, with a correction for the early iterations.

The first moment acts as momentum and smooths the path through noisy or ill-conditioned regions. The second moment gives each variable its own learning rate, scaled to the size of its recent gradients. This last feature is what suits Adam to the present problem. The penalty formulation produces a loss in which the constraint penalties (closure, curvature and the force target) have much larger gradients than the perimeter objective. A single step size would either crawl on the objective or overshoot on the penalties. The per-variable scaling of Adam absorbs most of this difference by itself.

A constant step size still leaves some jitter as the solution is approached. The cosine annealing schedule of Loshchilov and Hutter [62] addresses this. It lowers the learning rate from its initial value to near zero along a half-cosine over the run. It can also use warm restarts, which reset the rate periodically to help the search escape poor basins. With this schedule the optimiser takes large, exploratory steps early and small, refining steps late. This fits the present problem well. The search must explore the corrugated curvature landscape broadly at the start, then settle the geometry carefully near the end.

2.7.2 Penalty formulations of the constraints

The design problem is constrained. The perimeter must be minimised subject to closure, the turning condition and a target crush force. The penalty method turns this into an unconstrained problem, which a first-order optimiser can handle. Its classical treatment is given by Nocedal and Wright [63].

Equality constraints, such as closure and the 2π turning condition, are handled with *quadratic* penalties. Each adds a term proportional to the square of the residual. The term is zero when the constraint is met and grows smoothly as it is violated, so its gradient pulls the solution back towards feasibility. Inequality constraints are handled differently. Here the main one is that the crush force must not fall below target. It is handled with a *one-sided* penalty (of ReLU type) that acts only when the constraint is violated and leaves the feasible region untouched. Both penalty types are differentiable almost everywhere, so they fit directly into the automatic-differentiation graph.

The penalty method has a well-known weakness. Exact satisfaction of the constraints requires the penalty weights to grow without bound, which ill-conditions the landscape. At finite weights the solution is instead slightly infeasible, and the residual trades off against the objective. The augmented Lagrangian method removes this weakness. It adds explicit Lagrange-multiplier estimates to the penalty and reaches feasibility at finite weight [63]. The cost is an outer loop that updates the multipliers, and the extra bookkeeping it brings.

For the present problem the plain penalty method is the better balance. The curvature bound, which would otherwise be the hardest inequality to handle, is removed from the penalty set altogether, because it is enforced by construction (Chapter 4). Only closure and the force target remain, and their small residuals at finite weight are acceptable. A known and important point about penalty methods is that the *relative* weighting of the penalties governs the result. If the penalties are too strong relative to the objective, the design collapses towards a feasible but heavy shape. The weights must therefore be balanced, and ideally scaled to the size of the objective, rather than fixed in absolute terms.

2.7.3 Non-convexity and basins of attraction

Shape-optimisation problems are usually non-convex and have many local minima. The curvature-bounded crush problem is strongly non-convex. Its corrugated optimum has alter-

nating-sign curvature saturated at the bound. Many different corrugation counts and phasings give separate basins of attraction of similar quality [54]. A gradient method converges to the basin of its starting point, so the result depends strongly on the initialisation.

The usual response is global optimisation. One option is to run many starts from different points (multi-start). Another is to alternate local minimisation with random perturbation. The basin-hopping method of Wales and Doye follows the second route. It maps each point to its local minimum, which turns the landscape into a set of flat steps, then accepts or rejects perturbed minima by a Metropolis criterion [64].

A complementary strategy for a structure of multiple cross-sections is *warm-starting*. One section is optimised first. Its neighbour is then initialised from the converged result. Adjacent sections settle into compatible basins, and the expensive search for a good basin is shared across the stack. The handling of basins is therefore not a minor detail. It is a main driver of solution quality, because the design space of this problem is inherently multi-modal by the physics of Section 2.4.1.

All of this is made practical by reverse-mode automatic differentiation (AD). AD computes exact derivatives of a numerical program at a cost similar to running the program once. It avoids both the truncation error of finite differences and the effort of deriving adjoints by hand [65]. Modern deep-learning frameworks provide AD over a wide set of differentiable operations. PyTorch is one such framework. It combines an eager (define-by-run) AD engine with GPU support and a large operator library, and it is the framework used to implement the crush model and the curvature representation of this thesis [66].

The wider pattern is *differentiable physics*. A forward physical model is written in an AD framework, so that design and inverse problems can be solved by gradient descent through the simulation. This is now an active theme in computational engineering. Hoyer and co-workers showed that solving a structural topology-optimisation problem through a differentiable physics model improves the optima found [67]. Differentiable finite element solvers such as JAX-FEM bring automatic inverse design and sensitivity analysis to three-dimensional solid mechanics [68]. GPU-accelerated differentiable finite element libraries for solid mechanics are now available as engineering tools [69]. The present work sits in this tradition. It uses a fast analytical crush model, made differentiable from end to end, and optimises it by gradient descent in a curvature-space representation.

2.8 Concluding remarks

This review has gathered the elements of the design problem this thesis addresses. Together they reveal a gap that none of the surveyed work fills.

Three facts are established. First, a fast and accurate analytical crush model already exists. The FEUP/McLaren tool predicts the crush force of a partitioned composite cross-section in seconds, within about 1% of the commercial CZone reference and some three orders of magnitude faster [5, 6, 53]. Second, wall curvature is a primary driver of crush stress. Coupon

and tube studies show crush stress rising steeply as the radius of curvature decreases towards the manufacturing limit, captured by the empirical law $\sigma(R) = \sigma_0(aR^b + c)$ [3, 32–35]. At fixed thickness, force per unit perimeter is therefore largest at minimum radius, so cross-sectional shape becomes the governing design variable. Third, the methods needed to exploit this are individually mature. They include intrinsic, curvature-space curve representations that make a hard curvature bound trivial, gradient-based optimisers with adaptive step sizes and annealed schedules, penalty handling of constraints, and end-to-end automatic differentiation through a differentiable physical model [57, 59, 61–63, 66, 67].

The gap lies at the intersection of these three facts. Across the surveyed work, the design is described by a small set of scalar parameters and searched with gradient-free methods, because the accurate forward model is too expensive and too irregular to differentiate. The continuous shape of the load-bearing wall is rarely a design variable, and the constraint that matters most in composite crush, the minimum manufacturable radius of curvature, is checked after the fact rather than built into the formulation. No surveyed study combines a fast differentiable crush model with a curvature-space description of the boundary, so the leverage that curvature offers over the crush stress remains unexploited. That combination, an analytical force model differentiable from end to end, a boundary written in the very variable the physics depends on, and a gradient-based optimiser driving both, is the route this thesis develops in the chapters that follow.

Chapter 3

Analytical Framework and Problem Definition

This chapter establishes the foundation on which the optimisation work is built: the CrushAnalytica force model that serves as physics reference (Section 3.1), the Side Impact Structure used as reference geometry throughout the thesis (Section 3.2), the formal statement of the minimum-perimeter design problem (Section 3.3), the theoretical bounds that characterise its solution (Section 3.4), the corrections made to the baseline tool during this work (Section 3.5), and the inherited radial optimiser that this work took as its starting point (Section 3.6).

3.1 The CrushAnalytica force model

CrushAnalytica takes a finite element model of a structure and generates a prediction of the crush force that said structure is able to withstand. The pipeline was set out in full by Cardoso [6], and it will be summarised here.

The tool takes a meshed structure and a crush direction as input and runs in five stages. First, the mesh is made uniform: each quadrilateral shell element is split into two triangles, so that every later operation works on triangles alone. Second, the structure is partitioned into cross-sections. A plane normal to the crush direction is swept along the structure in fixed offset steps, and at each step its intersection with the triangulated surface is found with Möller's triangle-triangle intersection test [70], returning a closed polyline of nodes and edges. Third, each node of a cross-section is assigned local properties, namely wall thickness, ply count, inclination and radius of curvature, inherited from the mesh elements on which it lies. Fourth, the crush stress at each node is evaluated from its radius of curvature and integrated around the section to give the cross-section crush force. Fifth, the cross-section forces are assembled along the crush direction into the force-displacement response of the structure. Figure 3.1 sets out the five stages. Throughout this thesis the terms *partition* and *cross-section* are used interchangeably: each denotes one closed planar section produced by this slicing stage.

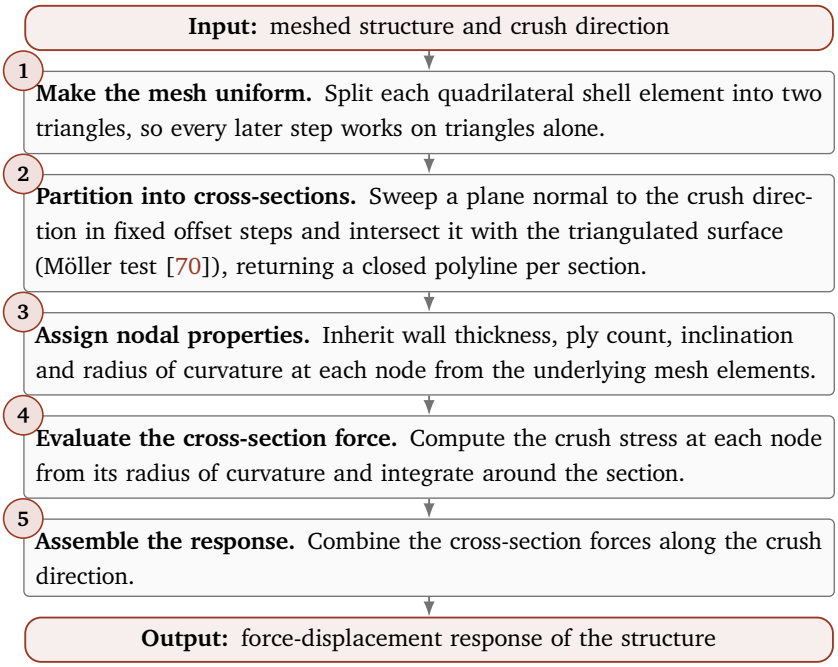


Figure 3.1: CrushAnalytica pipeline, from a meshed structure to the force-displacement response of the crushed component.

The optimisers of the later chapters reproduce the stress law of Section 3.1.1 and the nodal force integration of Section 3.1.2 exactly.

3.1.1 Crush stress law

The crush stress of a composite wall is modelled as a constant baseline stress scaled by an empirical curvature factor. For a local radius of curvature R, the curvature factor of the CFRP laminate used in this work is

$$\sigma_c(R) = \sigma_0 \left(7.95\, R^{-0.94} + 0.99\right), \tag{3.1}$$

where $\sigma_0 = 90\,\text{MPa}$ is the constant crush stress and R is expressed in millimetres.

The law grows without bound as $R \to 0$. The small-radius branch must be cut off, because woven CFRP cannot be laid up below a minimum manufacturing radius without fibre wrinkling and porosity. Draping a woven reinforcement over small radius tooling causes in-plane shear and out-of-plane wrinkling, which spoil geometric accuracy and weaken the laminate through fibre waviness and resin rich zones [71].

A lower bound has to be established to avoid a solution that is not physically manufacturable. The radius of curvature is clamped to the interval $[R_{\text{min}}, R_{\text{flat}}]$ with $R_{\text{min}} = 15\,\text{mm}$ the manufacturing lower bound and $R_{\text{flat}} = 1221.95\,\text{mm}$ the transition radius beyond which the wall is treated as flat.

The value $R_{\text{min}} = 15\,\text{mm}$ was chosen as a compromise between two competing factors. A smaller bound, such as $R_{\text{min}} = 5\,\text{mm}$, would let the optimiser reach lighter sections with a

lower perimeter. Such sections would have only theoretical value, because they could not be manufactured in practice. A larger bound, such as $R_{\min} = 25\,\text{mm}$, would give geometries that are easy to manufacture, but it would sacrifice too much of the performance the optimiser can achieve. The chosen value avoids both extremes. It does not unduly limit the optimisation, and it can still be produced in a real part with the right precision equipment and processes. This is what gives the solutions presented in the later sections a practical value.

Figure 3.2 plots this law for the calibrated CFRP coefficients.

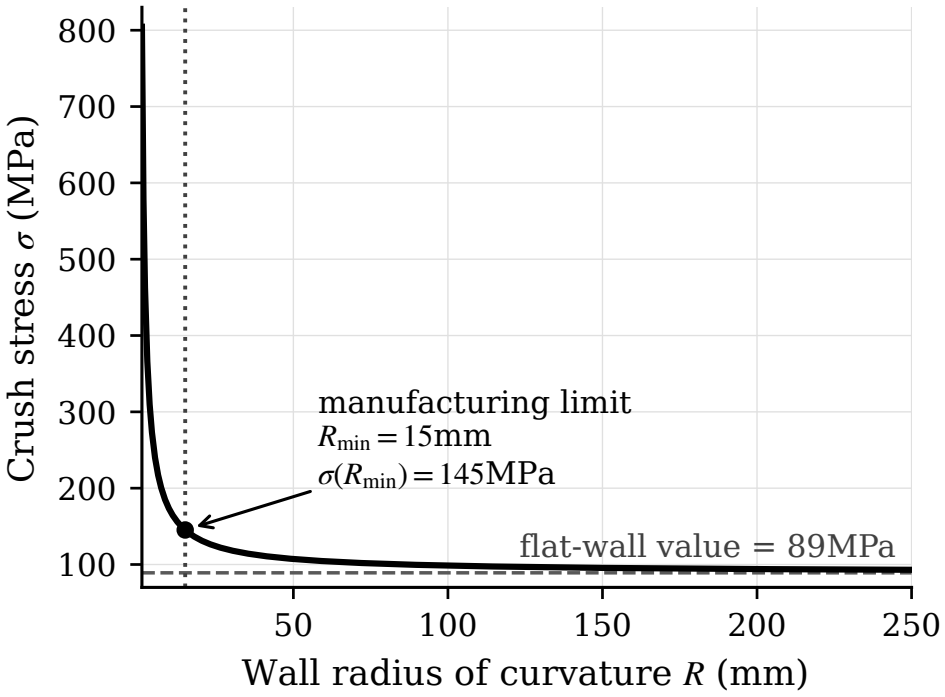


Figure 3.2: Crush stress as a function of wall radius of curvature for a representative woven CFRP laminate.

The coefficients in Equation (3.1) are the woven-CFRP curvature fit held in the material database of the tool, calibrated by Cardoso [6] to the FEUP crush-coupon campaigns of Moreira [35] and Lobão [34], which measured how the crush stress of CFRP laminates varies with the wall radius of curvature. They follow the general form $\sigma(R) = \sigma_0(aR^b + c)$ of Section 2.4.1, here with $a = 7.95$, $b = -0.94$ and $c = 0.99$.

3.1.2 Force integration

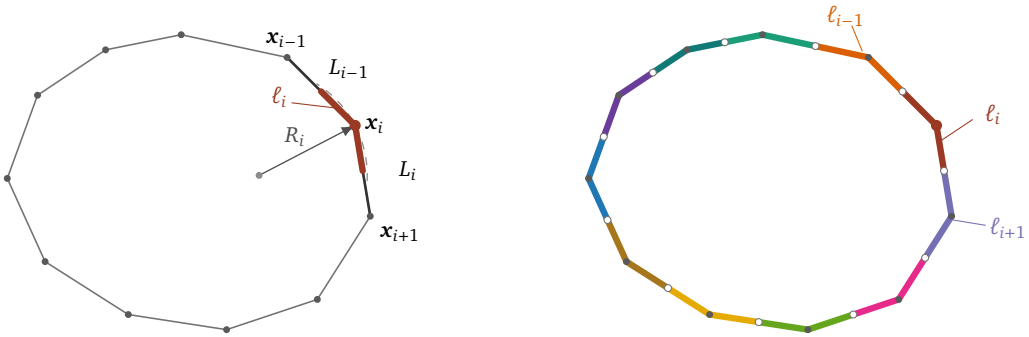
For a closed cross-section discretised into boundary nodes $\boldsymbol{x}_i$, each edge has length $L_i = \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|$ (with wrap-around), the perimeter is $P = \sum_i L_i$, and each node carries a tributary length

$$\ell_i = \tfrac{1}{2}\,(L_i + L_{i-1})\,. \tag{3.2}$$

The total crush force of the section is the nodal integration

$$F = \sum_i \sigma_c(R_i)\, \ell_i\, t, \tag{3.3}$$

where t is the wall thickness and R_i the radius of curvature at node i. Equation (3.3) makes the design trade explicit: force is the perimeter integral of a curvature-weighted stress, so a section can buy force either with perimeter (mass) or with curvature (shape). Figure 3.3a illustrates the construction for a single node. Figure 3.3b shows the tributary lengths of every node, which together partition the section perimeter.



(a) Length ℓ_i and radius of curvature R_i at a single node i.

(b) Tributary lengths of all nodes.

Figure 3.3: Force integration over a discretised cross-section.

3.1.3 Curvature reconstruction in CrushAnalytica

Unlike the optimisers of the later chapters, CrushAnalytica does not hold the curvature of a section as a known quantity. It has to reconstruct it from the sliced node positions. For each section the nodes are first ordered into a continuous polyline and parameterised by normalised chord length. A smoothing spline (the MATLAB `spaps` routine, a smoothing spline after de Boor [72]) is then fitted to the x, y and z coordinates as separate functions of this parameter, and the curvature is taken from the spline derivatives. A closed loop is handled by wrapping a few nodes from each end before fitting, so the curve is continuous across the seam, and a ribbed section is split into branches at its junctions and fitted branch by branch.

The curvature of the curve $\boldsymbol{r}(s)$ follows from its first and second derivatives,

$$\kappa = \frac{\|\boldsymbol{r}' \times \boldsymbol{r}''\|}{\|\boldsymbol{r}'\|^3}, \qquad R = \frac{1}{\kappa}, \tag{3.4}$$

evaluated on a smoothing spline fitted to the partition nodes, with R clamped to $[R_{\min}, R_{\text{flat}}]$.

3.2 Reference structure: the Side Impact Structure

The reference geometry used throughout this thesis is the Side Impact Structure (SIS) of a Formula 1 car, the same component on which CrushAnalytica was validated [6]. It is a closed, thin-walled CFRP tube that runs along the side of the survival cell and absorbs energy in a lateral impact by crushing progressively along its length. Its cross-section has roughly a stadium shape and tapers from one end to the other, so the section changes shape and size along its axis (Figures 3.4 and 3.5).

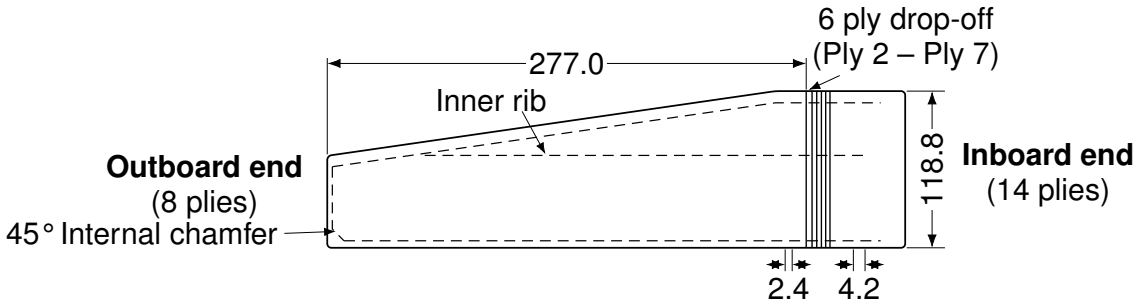


Figure 3.4: Side-view sketch of the Side Impact Structure, showing its taper, internal rib and ply drop-off from the inboard to the outboard end (adapted from [3]).

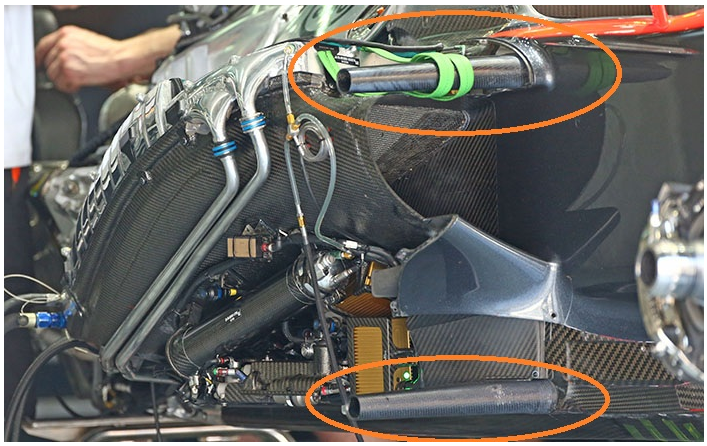
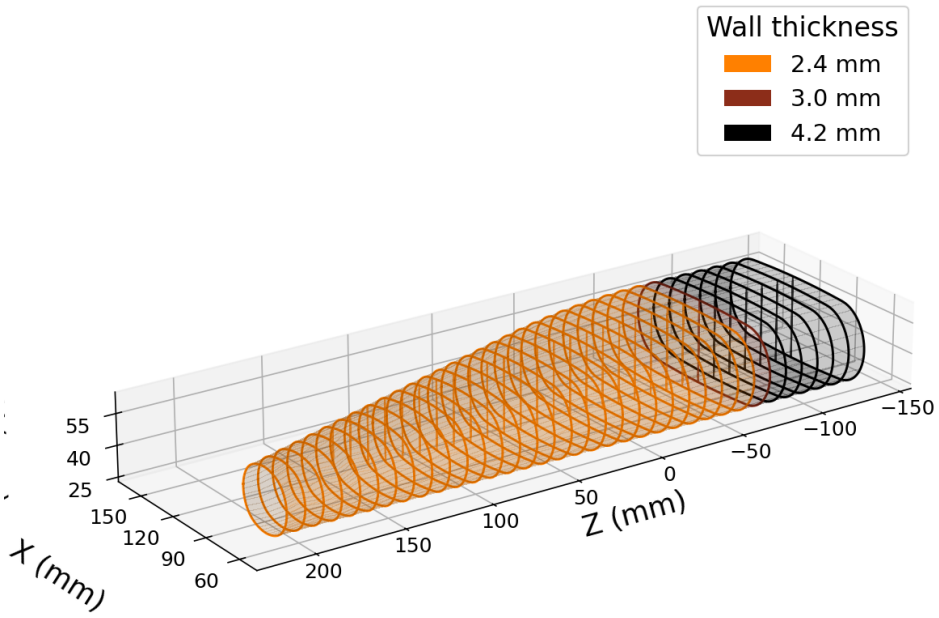


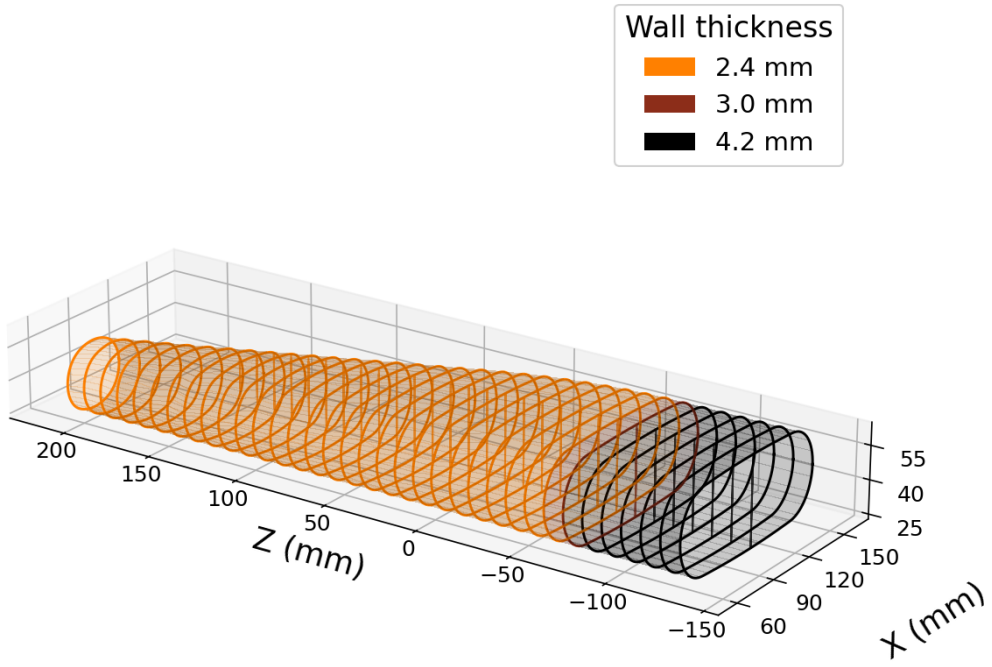
Figure 3.5: Upper and lower SIS (circled) inside the side-pod of an F1 car (adapted from [73]).

CrushAnalytica partitions the FE model of the SIS into a chosen number of cross-sections. Here a spacing of 10 mm was used, which gives 35 sections along the crush direction and covers about 340 mm of crushed length.

Each section carries a wall thickness inherited from the laminate of the FE model. The thickness falls into three bands along the structure: 4.2 mm, 3.0 mm and 2.4 mm. These correspond to 14, 10 and 8 plies respectively, at a ply thickness of 0.3 mm.



(a)



(b)

Figure 3.6: Two views of the Side Impact Structure partitioned sections assembled in 3D.

Most sections of this structure fall into these three bands. A closer look at the CrushAnalytica output, however, reveals a few outliers. They come from the sections that contain a rib, which gives the cross-section a branched topology. The rib is 1.2 mm thick (4 plies) and joins a thicker wall. The thickness values outside the three bands are therefore the nodes in the transition from the nodes belonging to the rib alone to the nodes shared by the rib and the outer wall. Figure 3.7 shows this for the last section, where a rib divides the stadium-shaped outer wall.

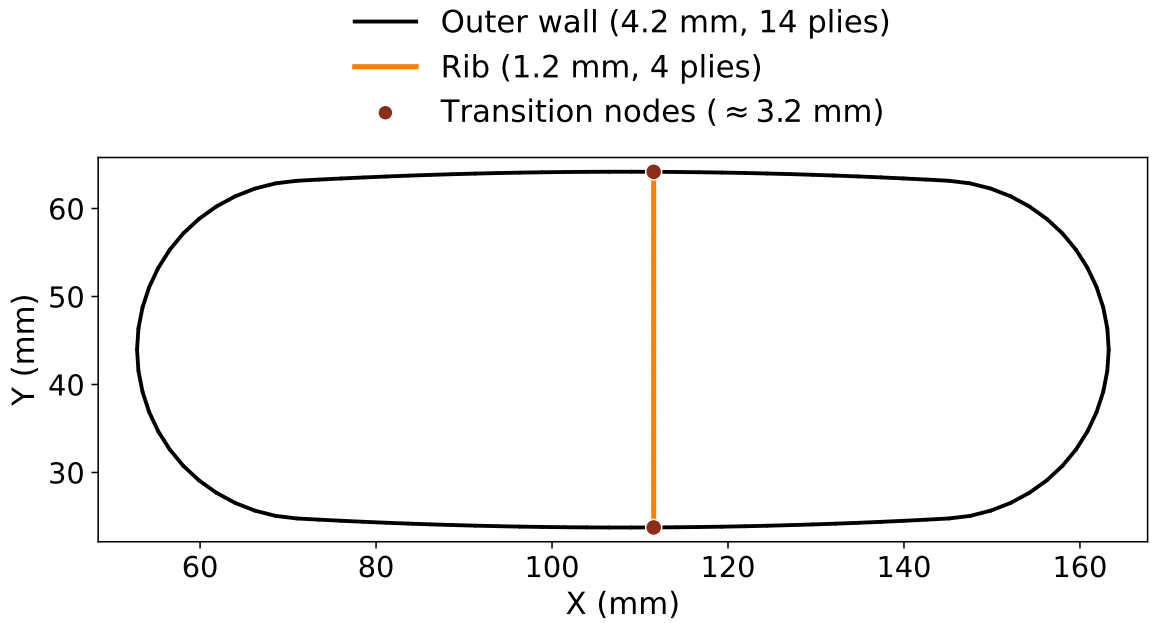


Figure 3.7: Cross-section of the SIS, showing the branched topology. The stadium-shaped outer wall (4.2 mm, 14 plies) is divided by an internal rib (1.2 mm, 4 plies).

Each section carries a crush force, computed by the analytical model from its original geometry. These cross-section forces are the targets the optimiser must meet (Figure 3.8).

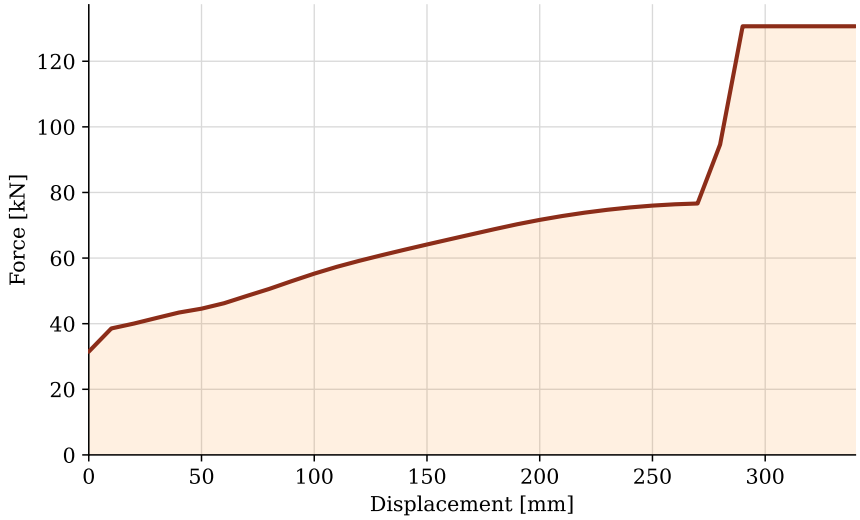


Figure 3.8: Force-displacement curve obtained from CrushAnalytica, crushing direction from outbound to inbound end.

They are highest at the thick sections, near the inbound end, about 130.7 kN, and fall along the structure to about 31.5 kN at the outbound end, reflecting both the thickness taper and the shrinking section.

Another aspect taken into account is that the original SIS structure is constrained to a

certain size, whether by mounting requirements or by the surrounding side pod fittings. A maximum width and height were therefore determined for each partition of every section. These dimensions were measured relative to the centroid of the section and stored for later use as a size constraint, ensuring that the solution fits within the same envelope as the original structure (Figure 3.9).

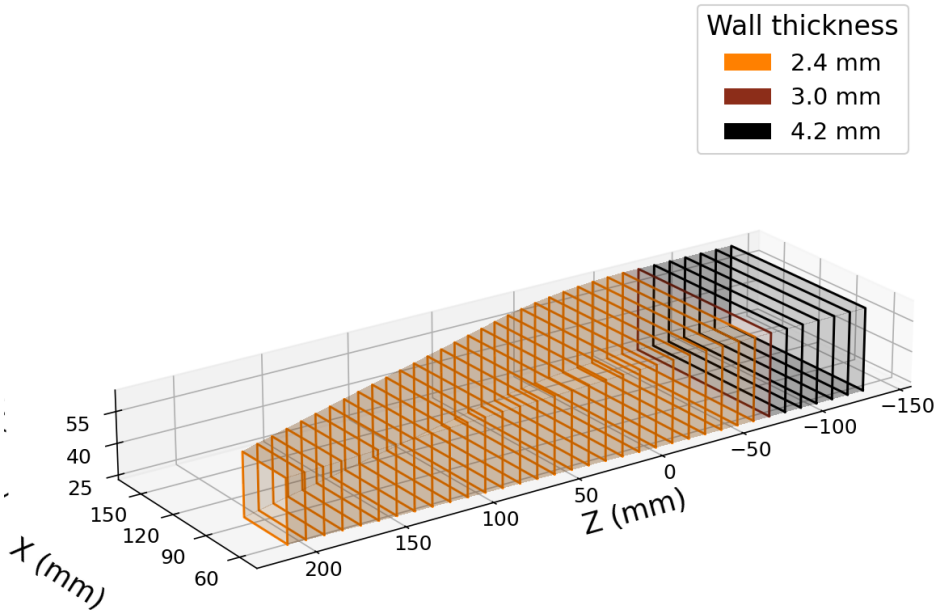


Figure 3.9: Original SIS design envelope.

A final property carried over from the original SIS is a measure of how smoothly its cross-section evolves along the crush direction. Because the real part is a continuous lofted shell, the change in shape from one partition to the next is gradual, and any optimised replacement must preserve a comparable smoothness to remain manufacturable. To compare partitions node by node, every partition is resampled to the same number of boundary nodes N, equispaced in arc length and sharing a common datum, so that a given index i denotes the same position around the perimeter in every partition.

Each node then has a position in space,

$$\boldsymbol{P}_{k,i} = (x_{k,i},\, y_{k,i},\, z_k), \tag{3.5}$$

where k indexes the partition along the crush direction (z) and i the node around the perimeter.

For a fixed node i, consecutive partitions are joined by the loft vector

$$\boldsymbol{v}_{k,i} = \boldsymbol{P}_{k+1,i} - \boldsymbol{P}_{k,i}. \tag{3.6}$$

At every interior partition k the two loft segments meeting at node i form a kink, whose angle is

$$\theta_{k,i} = \arccos\left(\frac{\boldsymbol{v}_{k-1,i} \cdot \boldsymbol{v}_{k,i}}{\|\boldsymbol{v}_{k-1,i}\|\,\|\boldsymbol{v}_{k,i}\|}\right). \tag{3.7}$$

A value $\theta_{k,i} = 0$ means the loft line passes straight through node i (a perfectly smooth transition); larger values indicate a sharper direction change between partitions. Figure 3.10 illustrates the construction.

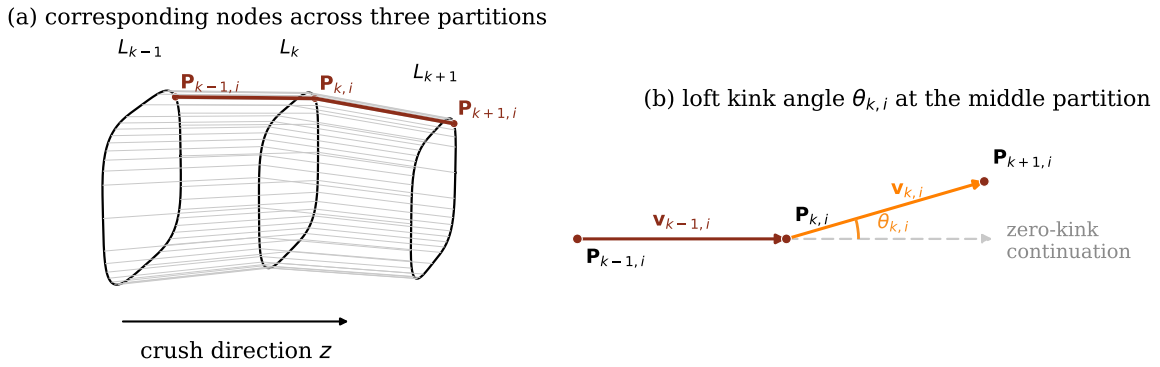


Figure 3.10: Loft kink-angle procedure. (a) A node index i is tracked across three consecutive partitions, giving $\boldsymbol{P}_{k-1,i}$, $\boldsymbol{P}_{k,i}$ and $\boldsymbol{P}_{k+1,i}$. (b) The kink angle $\theta_{k,i}$. The shape change is exaggerated for clarity.

The manufacturability allowable is taken as the largest kink found anywhere in the original SIS,

$$\theta_{\max} = \max_{k,i} \theta_{k,i} \approx 2.43^\circ, \tag{3.8}$$

which sets the smoothness budget for the multi-section optimisation of Chapter 5: the optimised stack is required to keep its inter-partition kink at or below $\theta_{\max}$, so the result is no less smooth, and therefore no less manufacturable, than the original.

The full partition stack, with its per-node geometry and properties, is stored in `BigArray.mat` and is the starting point for the multi-section optimisation of Chapter 5. This last referenced file is nothing more than a MATLAB data file, that results from extracting the most meaningful values after running CrushAnalytica. The design task for the full structure follows directly: replace each original section with a smaller cross-section that is able to achieve equal crush force.

3.3 Problem statement

At fixed wall thickness, the mass of a section is proportional to its perimeter, so the design problem solved throughout this thesis is the minimum perimeter problem

$$\begin{aligned} \min_{\partial\Omega} \quad & P(\partial\Omega) \\ \text{s.t.} \quad & F(\partial\Omega) \geq F_{\mathrm{t}}, \\ & |\kappa(s)| \leq 1/R_{\min} \quad \forall s, \\ & \partial\Omega \text{ a closed, non-self-intersecting curve,} \end{aligned} \tag{3.9}$$

where $\partial\Omega$ is the section boundary, F is evaluated by the analytical model of Section 3.1, and the curvature bound encodes manufacturability. For the multi-section structure the problem is solved for every partition, with additional inter-cross-section continuity requirements introduced in Chapter 5.

Several effects are deliberately left out of this formulation. The crush force is taken from the analytical model, which assumes stable progressive crushing throughout. Global and local buckling, and the transition to catastrophic collapse, are not modelled, so the optimiser is trusted to remain in the progressive regime rather than to predict its loss. The wall thickness and the laminate are fixed inputs for each section, not design variables, so the only freedom is the shape of the boundary curve. The impact is treated as axial and quasi-static, so the inclination and strain-rate dependencies that the analytical model can represent (Sections 2.4.2 and 2.4.4) are not used. These choices keep the problem a pure question of cross-sectional shape, which is where the curvature opportunity of Chapter 2 lies. The only requirement added beyond problem (3.9) is inter-cross-section continuity, needed so that the stack of optimised sections forms a manufacturable body. It is introduced in Chapter 5.

3.4 Theoretical limits and the expected optimal shape

Two elementary bounds characterise the solution of problem (3.9) for the single-section target force.

First, consider the smallest closed shape the manufacturing constraint permits. The curvature bound $|\kappa(s)| \leq 1/R_{\min}$ caps how sharply the wall can turn: it can never bend to a radius tighter than $R_{\min}$. A closed boundary, on the other hand, must turn through a full revolution, a total angle of 2π, in order to return to its starting point. The shape that does so is a circle of radius $R_{\min}$, whose perimeter is

$$P_{\text{circle}} = 2\pi\, R_{\min} \approx 94.2\,\text{mm}. \tag{3.10}$$

No admissible section can be shorter: any smaller closed loop would have to bend below $R_{\min}$ somewhere and so break the manufacturing limit.

This minimal circle carries a second meaning. Because it lies entirely at $R_{\min}$, and the crush stress is largest exactly there, with Equation (3.1) giving $\sigma_c(R_{\min}) \approx 145\,\text{MPa}$, every millimetre of its perimeter develops the highest crush stress the laminate can reach.

Evaluating the force integral of Equation (3.3) for this constant radius section, where every node shares the same stress, gives

$$F_{\text{circle}} = \sigma_c(R_{\min})\, t\, P_{\text{circle}} \approx 33\,\text{kN}, \tag{3.11}$$

with t the section wall thickness.

This yields the first conclusion. The minimal circle is then never sufficient when the target

force is above 33 kN, and since it is already saturated at the maximum crush stress, the only remaining way to raise the force is to add perimeter. How much length is the subject of the second bound.

Secondly, since σ_c has maximum value at $R = R_{min}$, a lower bound on the perimeter of *any* feasible section follows from assuming every point of the boundary fully saturated at the manufacturing limit:

$$P_{min} = \frac{F_t}{\sigma_c(R_{min})\, t} \tag{3.12}$$

This bound is not achievable by a smooth convex shape: a closed curve with $|\kappa| = 1/R_{min}$ everywhere and total turning 2π must alternate the sign of its curvature. The expected optimum is therefore a shape whose boundary oscillates at the curvature limit.

Any boundary parameterisation unable to express alternating sign curvature cannot reach it. This prediction drives the parameterisation choices examined in Section 3.6 and Chapter 4. One important remark at this stage is that the lower bound of Equation (3.12) is very valuable to assess the performance of the optimisation and, simultaneously, how close the solution is to what the theory predicts (Figure 3.11).

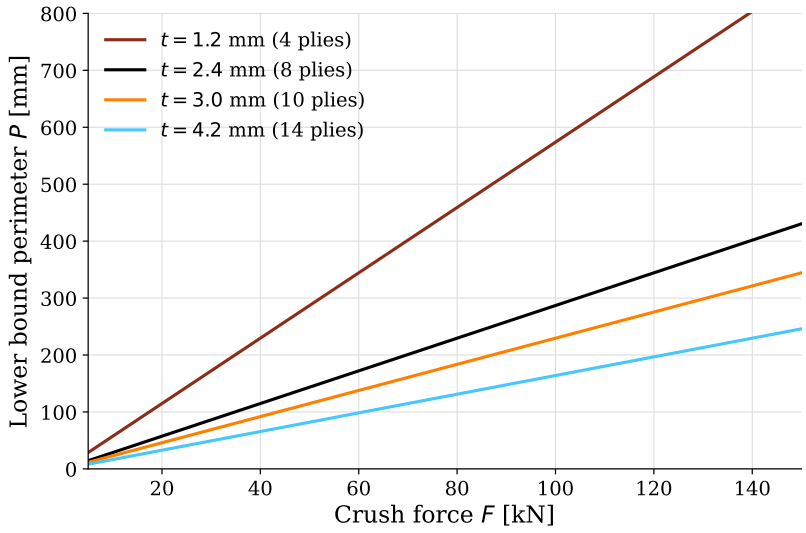


Figure 3.11: Perimeter as a function of target crush force for different ply counts at $R_{min} = 15$ mm.

3.5 Correction of a partitioning cross-section artefact

While running the inherited optimiser, some results came out wrong in a way that pointed to the tool rather than to the optimisation. The input geometry was symmetric and had a regular stadium shape, so the optimised sections should have been smooth and symmetric too. Instead a few of them carried small wrinkles, short kinks in an otherwise smooth wall, always in the same places. They were easy to miss, because each wrinkle is only a fraction of a millimetre and shows up only on close inspection. This matters because every downstream quantity, wall thickness, ply count, inclination, radius of curvature and ultimately the crush

force, is read off the cross-section nodes, so a wrinkle in the geometry feeds straight into the force. Figure 3.12 shows one such result.

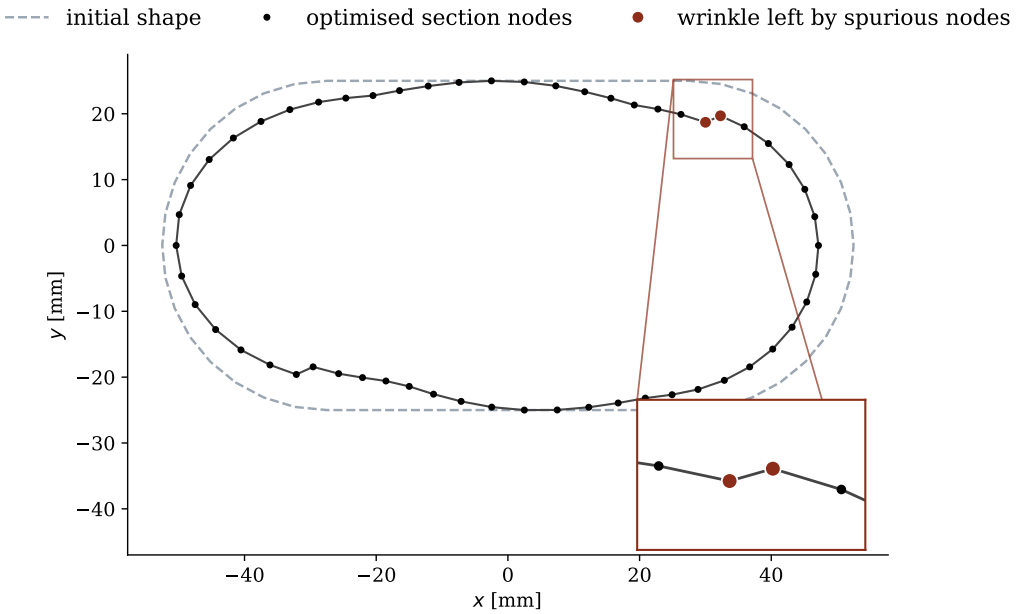


Figure 3.12: An optimised elliptical section, shown against its smooth initial shape.

The wrinkles were traced back to the partitioning stage, the step that turns the meshed structure into cross-sections. Comparing a partitioned section against the mesh it was cut from made the problem plain: the partition carried a few extra nodes that the mesh itself does not have (Figures 3.13 and 3.14). In both cases the extra nodes sit a fraction of a millimetre off the wall.

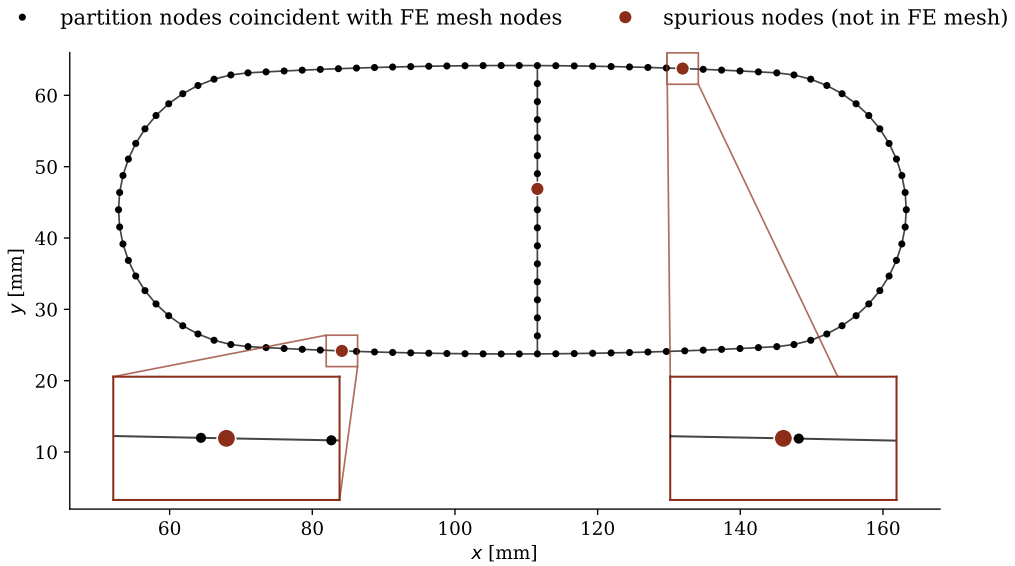


Figure 3.13: Partitioned SIS cross-section at $z = -132.4\,\text{mm}$.

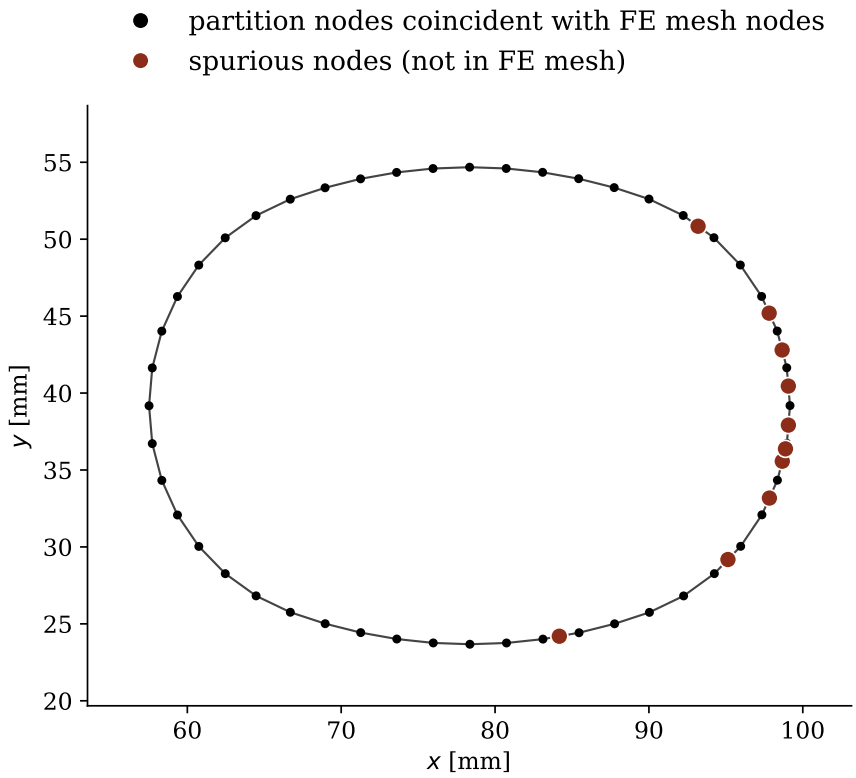
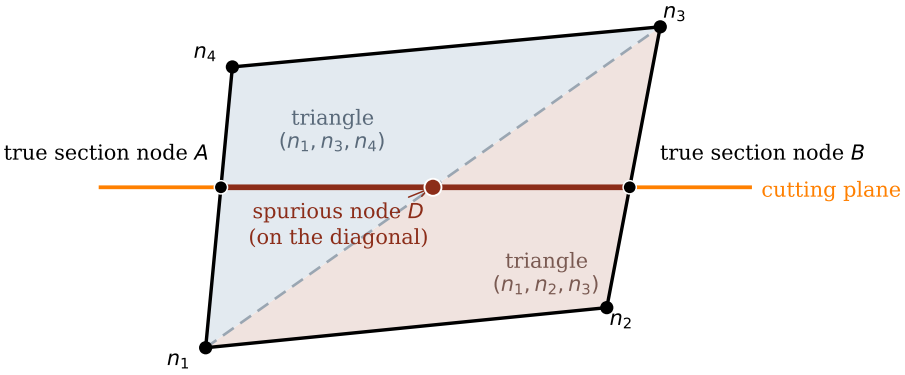
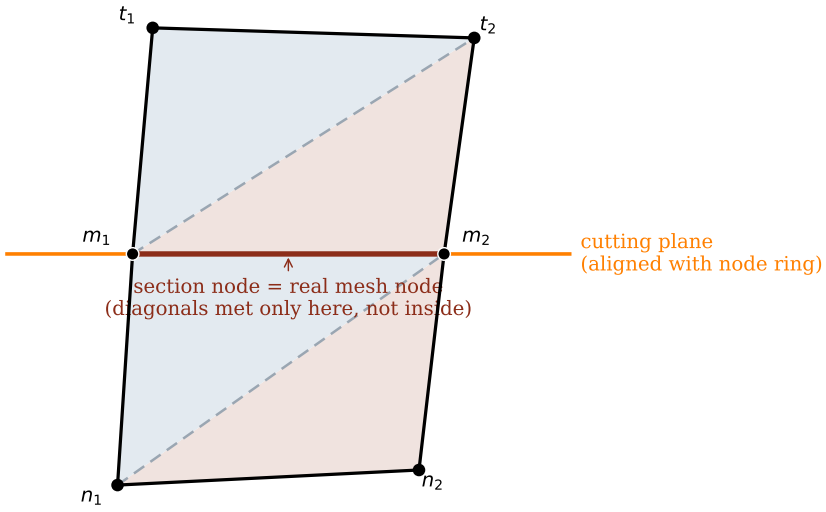


Figure 3.14: Partitioned SIS cross-section at $z = 197.6\,\text{mm}$.

The cause lies in how the partitioning algorithm cuts the mesh. To work on triangles, it first splits every quadrilateral element into two along a fixed diagonal. It then intersects the cutting plane with each triangle on its own. When the plane passes through the nodes of the element, the two triangles meet cleanly at those nodes. When the plane instead crosses the interior of the element, it also crosses the added diagonal, and the two triangles meet at a point on that diagonal that is not a real node. Whether this happens is dictated by the meshed model, which in this case showed that the nodes were not sitting exactly all in the same plane, by a small difference of around 0.02 mm. The algorithm does run a duplicate-node check meant to catch points like these, but its tolerance is far tighter than the offset involved, so the false nodes slip through. Figure 3.15 contrasts the two cases.



(a) Plane through the element interior: the two triangle segments meet on the diagonal, planting a false node.



(b) Plane aligned with the node ring: the triangles meet at real nodes, so no false node appears.

Figure 3.15: How the partitioning algorithm plants a false node.

A cleanup routine, `cleanPartitionGeometry`, was written to remove these nodes before any property is computed. It runs at the start of each section inside `layersProp`, so the crush force is always evaluated on the cleaned section. To avoid touching genuine geometry, it flags a node i as spurious only when three criteria hold at once. Let ℓ^- and ℓ^+ be the lengths of its two edges, c the straight chord across them, θ_i the turning angle at the node and ℓ_{med} the median edge length of the branch. The criteria are

$$\begin{aligned} &\min(\ell^-, \ell^+) \;\le\; 0.35\, \max(\ell^-, \ell^+), \\ &|180^\circ - \theta_i| \;\le\; 12^\circ, \\ &(\ell^- + \ell^+) - c \;\le\; \max\!\left(10^{-3}\,\text{mm},\; 0.02\,\ell_{\text{med}}\right). \end{aligned} \tag{3.13}$$

The first makes one edge much shorter than the other, so the node sits right next to a real one; the second keeps the wall almost straight through it; the third ensures that dropping the node barely changes the length of the section, so it lies essentially on the line between its neighbours. A node that meets all three is deleted and its neighbours reconnected, while rib junctions, branch ends and genuine sharp corners are always kept. With the routine in place the wrinkles disappear and the optimised sections come out smooth and symmetric, as the input geometry implies they should.

Beyond restoring the symmetry of the optimised sections, the fix improves every force prediction the tool makes, since all downstream quantities are read off the partition nodes. Because the routine runs inside CrushAnalytica itself, the improvement is carried back into the underlying tool, meeting the robustness objective of Section 1.3.

3.6 The inherited radial optimiser

The optimiser presented in this section was not developed from scratch. It is the geometry-optimisation module that Cardoso built alongside CrushAnalytica [6] (the `optimizationModule`), which the present work inherited as its starting point. The module describes the cross-section boundary by nodal radii placed on fixed polar rays and drives those radii with a gradient-based optimiser, reducing the section perimeter while holding its crush force above an initial floor.

This section documents that inherited module exactly as it was received: how it represents the geometry (Section 3.6.1), how it evaluates the crush force in a differentiable way (Section 3.6.2) and the objective it minimises (Section 3.6.3). The refinements the present work then made to it, and the results they produced, are the subject of Section 3.7.

After fixing some bugs in the original code, and using different strategies to try and improve the optimiser, even in that refined form the optimiser could not reach the optimum that the saturation bound of Section 3.4 predicts. The conclusion reached was that some problems were structural, meaning that without changing the foundation of the optimiser itself there was no real way around them.

3.6.1 Geometric representation

The boundary is described in polar form about the section centroid. Each node is converted to spherical coordinates relative to the centroid, the two angles of every node are then frozen, and only the radial distance r_i is left as a design variable. The geometry is reconstructed by placing each node back on its own fixed ray at the learned radius. In the plane of a section this reduces to a radius function $r(\phi)$ on fixed angular rays ϕ_i, with the radii as the optimisation variables. This polar description, also called a radial or star-shaped parameterisation, is a standard way to represent a closed boundary in shape optimisation [54, 74] (Figure 3.16).

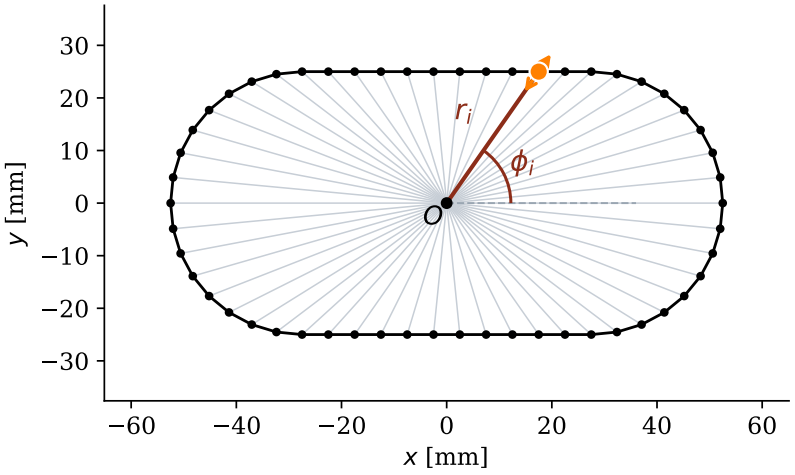


Figure 3.16: Radial boundary parameterisation.

3.6.2 Differentiable force evaluation

The crush force must be a differentiable function of the design radii so that it can be driven by gradient descent. The inherited module builds that pipeline node by node. The ordered boundary nodes are first wrapped periodically, by copying a small overlap of points from each end onto the other, so that the closed loop is treated as a continuous periodic curve. The wrapped nodes are parameterised by normalised cumulative chord length, and a natural cubic spline is fitted through them with an external library (`torchcubicspline`). This interpolating spline passes exactly through every node. Its first derivative is available in closed form, and the second derivative is taken from a central finite difference of the first. The radius of curvature at each node then follows from the Frenet expression

$$R_i = \frac{\|\boldsymbol{x}'(s_i)\|^3}{\|\boldsymbol{x}'(s_i) \times \boldsymbol{x}''(s_i)\|}, \tag{3.14}$$

The original clamp value used by Cardoso was 5 mm. As discussed in Section 3.1.1, this value was changed to 15 mm. The crush stress at each node is then read from the curvature law of Equation (3.1).

The inherited module collapses these nodal stresses into a single force through

$$F = t\,P\,\overline{\sigma_c}, \qquad \overline{\sigma_c} = \frac{1}{N}\sum_{i=1}^{N} \sigma_c(R_i), \tag{3.15}$$

the wall thickness t times the perimeter P times the mean nodal crush stress over the N boundary nodes. Every step from the radii to F and P is a differentiable PyTorch operation [66], so gradients of both quantities with respect to the radii are obtained by automatic differentiation.

3.6.3 Objective and optimisation

The inherited module selects its objective through boolean flags: minimise perimeter or maximise crush force. The thesis problem is the minimum-perimeter case. Its loss adds the constraints of problem (3.9) to the perimeter as soft penalties [63],

$$\mathcal{L} = P + \lambda_s S(\boldsymbol{r}) + \lambda_F \operatorname{relu}(F_0 - F)^2 + \lambda_r \sum_i \operatorname{relu}(r_i - r_i^0)^2, \tag{3.16}$$

where P is the perimeter, F the crush force and F_0 the initial crush force held as a force floor. The third term penalises any drop of the force below that floor, and the last is an inward-only envelope that penalises any node moving outward beyond its initial radius r_i^0, so the section can only shrink. The smoothness term $S(\boldsymbol{r})$ is the sum of squared second differences of the radii along the node ordering,

$$S(\boldsymbol{r}) = \sum_i (r_{i+1} - 2r_i + r_{i-1})^2, \tag{3.17}$$

which discourages neighbouring radii from varying abruptly.

The three penalty weights are all fixed at the same large value $\lambda_s = \lambda_F = \lambda_r = 10^6$, which makes the constraints effectively hard. The loss is minimised with the Adam optimiser [61] at a fixed learning rate over a long, fixed run of the order of 10^5 iterations, with no learning-rate schedule, convergence test or early stopping.

3.7 Refinements to the inherited optimiser

The inherited module of the previous section is a working baseline, but the present work refined it before using it. The polar parameterisation was kept, while the force evaluation, the loss and the optimisation machinery were rebuilt over a series of versions; the last of these produced the results reported in Section 3.7.2. This section records those changes in outline, with the emphasis on what changed and why rather than on the implementation detail.

3.7.1 Summary of changes

Several changes were made to improve the optimiser, namely:

- **A smoothing spline in place of the interpolating one.** The inherited module fitted a cubic spline that passes exactly through every node, which does not match the smoothing CrushAnalytica applies before it reads the curvature, so on an optimised section the two pipelines disagree. It was replaced by a smoothing spline (a Reinsch-form [75] re-implementation of the `spaps` routine of the reference tool [72]) that reproduces the reference curvature pipeline of Section 3.1.3, bringing the reconstructed curvature back into line with CrushAnalytica.

- **Closed-loop wrapping of the smoothness penalty.** The inherited smoothness term of Equation (3.17) took second differences over the open node ordering, so the seam where the boundary closes on itself, its first and last nodes, carried no penalty and the optimiser could leave an unsmoothed kink there. Wrapping the radii around the closed loop before differencing extends the penalty to every node, the seam included.
- **A more robust optimisation loop.** The fixed-length run was replaced by a cosine-annealing learning-rate schedule [62] with early stopping once the feasible objective stops improving, gradient clipping, and the detection and rollback of non-finite states. The best feasible state seen during the run is kept and restored, so the reported result is never worse than the best design the run visited.
- **Logging and batch tooling.** A command-line interface was added that saves a log and timestamped result files for each run, making the later runs easy to reproduce.

The inherited module had offered more than one objective, including a maximum crush force mode. The refined module narrows this to the minimum perimeter objective, which is the formulation relevant to the present application: the lightest section that still meets a required crush force.

3.7.2 Results

The refined optimiser was run in minimum perimeter mode on a set of seed shapes, some of which, such as the square and the rounded rectangle, were generated purely as a means of confirming the correct working of the tool, while the stadium seed approximates the SIS cross-section that is the target geometry of the rest of this thesis.

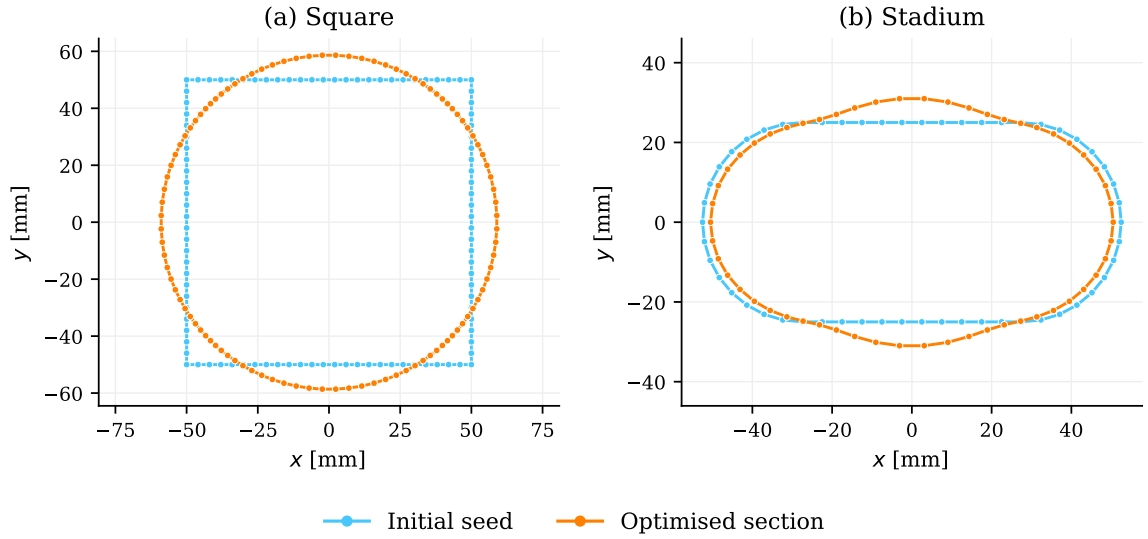


Figure 3.17: Minimum-perimeter results of the refined radial optimiser for the square and stadium seeds of Table 3.1.

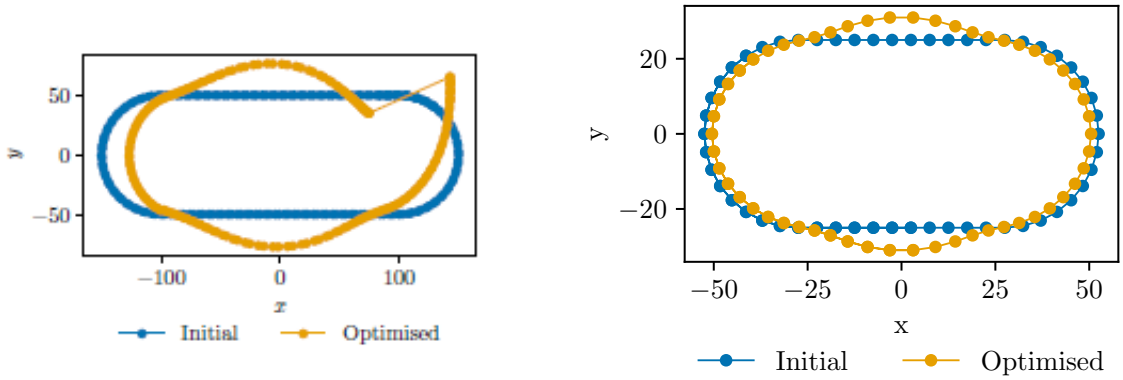
Each was driven down in perimeter while its own initial crush force was held as the floor. Table 3.1 collects three representative runs.

Table 3.1: Minimum perimeter results of the refined radial optimiser for three seed shapes.

Seed	P_0 [mm]	P [mm]	$P_{\min}$ [mm]	ΔP	ΔF
Square	400.0	369.6	266.3	−7.6%	−0.0%
Stadium	266.8	259.8	202.4	−2.6%	+0.2%
Rounded rectangle	600.0	567.8	390.1	−5.4%	+0.2%

Across them the perimeter falls by between about 3 and 8% while the crush force stays within half a percent of its initial value. Another important remark is that the worst case of perimeter loss happens precisely for the target geometry of this work. Even so, every optimised perimeter still exceeds the theoretical lower bound $P_{\min}$ of Equation (3.12), foreshadowing the limitation discussed below. The optimised boundaries are smooth, rounded versions of their seeds, as Figure 3.17 shows.

The effect of the refinements is clearest when the same input is run through both versions of the module. Figure 3.18 contrasts the inherited script and the refined optimiser on identical elliptical seed data.



(a) Inherited script (adapted from [6])

(b) Refined optimiser

Figure 3.18: Minimum perimeter optimisation of the same elliptical seed before and after the refinements: (a) the inherited script (adapted from [6]) (b) the refined optimiser.

The force and perimeter behave as intended, but the radius-of-curvature profile of the optimised sections does not. Plotted around the perimeter, it is strongly spiked: isolated nodes plunge to the minimum radius of curvature while their neighbours stay almost flat, so the profile is a row of narrow spikes rather than a smooth distribution (Figure 3.19).

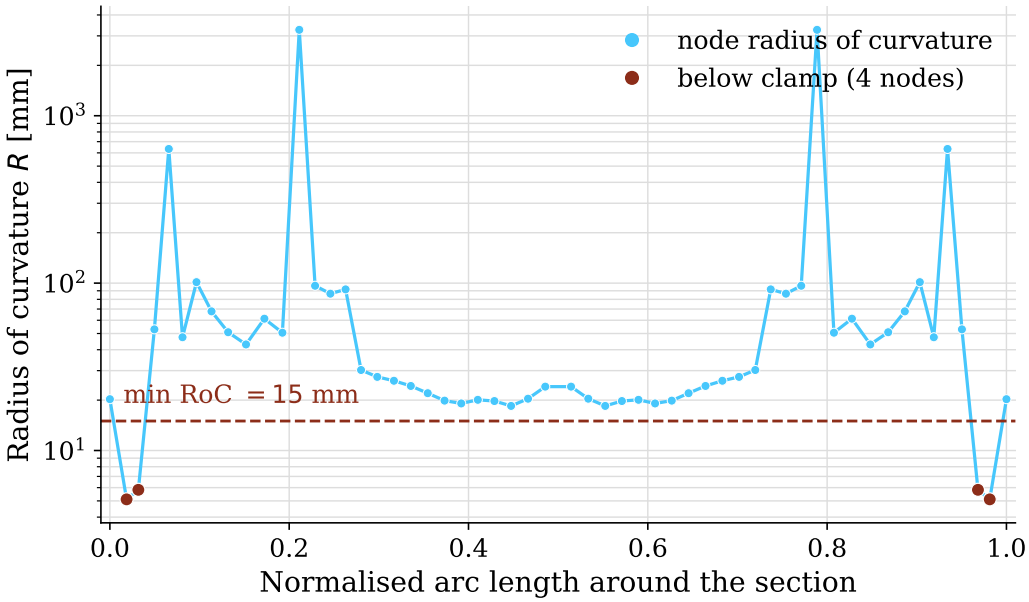


Figure 3.19: Radius of curvature profile around the optimised Stadium section (log scale).

3.7.3 Remaining limitations and the need for a new optimiser

The refinements above raised the force model to the fidelity of the reference tool and made the runs robust, but they leave untouched a restriction that is structural to the radial parameterisation itself. Each polar ray intersects the boundary exactly once, so the representable family is limited to star-shaped curves (Figure 3.20).

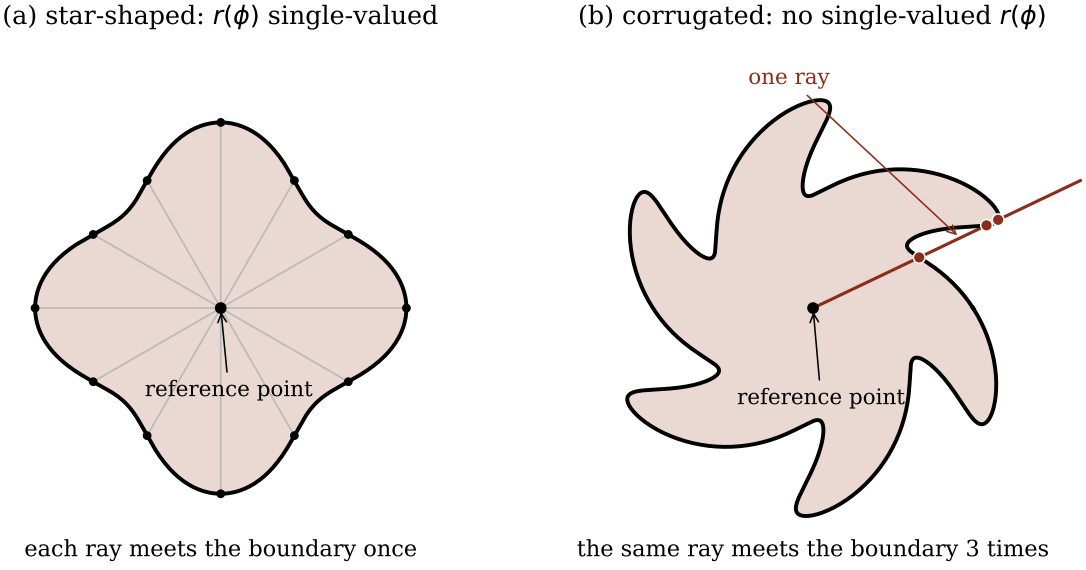


Figure 3.20: Representational limit of a radial description.

The wavy optimum predicted by the saturation bound of Section 3.4, with its alternating-

sign curvature at $|\kappa| = 1/R_{\min}$, therefore lies outside the reachable design space. Another indication of this model limitation is that the different initial geometries all converge to different solutions, pointing to the fact that they are, most likely, stuck in local minima. This is not a limitation that a different seed, learning rate or penalty weight could remove, nor one that the higher fidelity force model could reach around; it is a property of the parameterisation.

The inherited module gave a working radial optimiser; the present work raised its force model to the fidelity of the reference tool and made its runs robust, and in that refined form it reduces perimeter at constant force across a range of seeds. But the spiked, locally saturated curvature of its results and its inability to reach the smoothly alternating optimum are two faces of the same structural gap: a star-shaped boundary cannot fold. Closing that gap requires abandoning the radius as the design variable. The next chapter does exactly this, taking the curvature distribution itself as the unknown and satisfying the manufacturing bound by construction.

Chapter 4

Direct Curvature Formulation

This chapter develops the curvature optimiser (`rocOptimizer`) that constitutes the core contribution of the thesis. It follows directly from the limitation diagnosed at the close of Chapter 3: a radial description cannot fold, so the wavy optimum predicted by the saturation bound of Section 3.4 lies outside the family of shapes it can express (Figure 4.1).

The chapter builds this formulation one piece at a time. Section 4.1 introduces the design variables and shows how a boundary is written as a curvature profile; Section 4.2 reconstructs the closed cross-section from that profile so that the crush force can be read from it; and Section 4.3 states the constraints and assembles the loss function the optimiser minimises. Section 4.4 sets out the optimisation strategy that keeps every reported section admissible. Section 4.5 then characterises the formulation on a single cross-section in isolation: the shape it converges to, how it compares with the radial optimiser of Chapter 3, how many iterations it needs, how much force one section can carry, and whether the result is independent of the starting shape. Finally, Section 4.6 applies the formulation to the real SIS cross-sections.

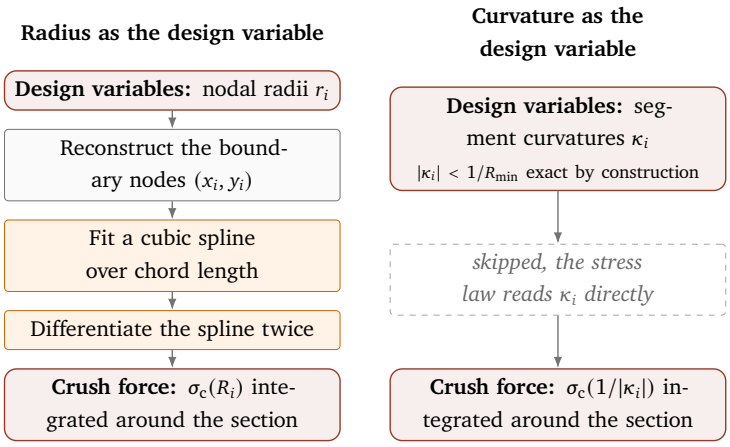


Figure 4.1: From design variables to crush force in the radial formulation (left, Section 3.6.2) and in the direct curvature formulation of this chapter (right).

4.1 Design variables

The closed boundary is discretised into N arc segments of equal length $\Delta s = L/N$. The design vector consists of N unconstrained real variables u_i and the logarithm of the total arc length, $\log L$. The signed curvature of segment i is obtained through a saturating map,

$$\kappa_i = \frac{\tanh(u_i)}{R_{\min}}, \tag{4.1}$$

which enforces the manufacturing bound $|\kappa_i| < 1/R_{\min}$ exactly, for any value of the underlying variable, the constraint is built into the parameterisation rather than penalised.

The hyperbolic tangent saturates towards $\pm 1/R_{\min}$ as $|u_i|$ grows, so no value of u_i can place a segment beyond the bound, and the curvature sharper than the minimum radius allows is unreachable by construction. Storing the arc length in log guarantees $L > 0$ unconditionally.

A starting shape is turned into such a design vector by measuring how it bends. Figure 4.2 traces this on a worked example, a circle of radius 30 mm cut into $N = 8$ segments, in five steps:

- **Step 1 - segment length and heading.** Each segment has a length and a heading θ.
- **Step 2 - perimeter and step.** The segment lengths sum to the perimeter L, which fixes the common step $\Delta s = L/N$.
- **Step 3 - turning per segment.** The heading turns by $\Delta\theta$ from one segment to the next, and the turns sum to 2π, so the loop closes.
- **Step 4 - curvature.** The curvature of a segment is its turning rate $\kappa_i = \Delta\theta_i/\Delta s$, the slope of the accumulated heading against arc length.
- **Step 5 - encoding.** The saturating map encodes each curvature as $u_i = \operatorname{atanh}(\kappa_i R_{\min})$.

The stored design vector is then the N variables u_i and the length $\log L$. For the worked circle every segment bends by the same amount, so all eight values are equal, $u = [\,0.57,\ 0.57,\ \ldots\,]$, with $L = 183.7$ mm: the section is stored in just $N + 1$ elements. Running the same chain forwards recovers the boundary, the subject of Section 4.2.

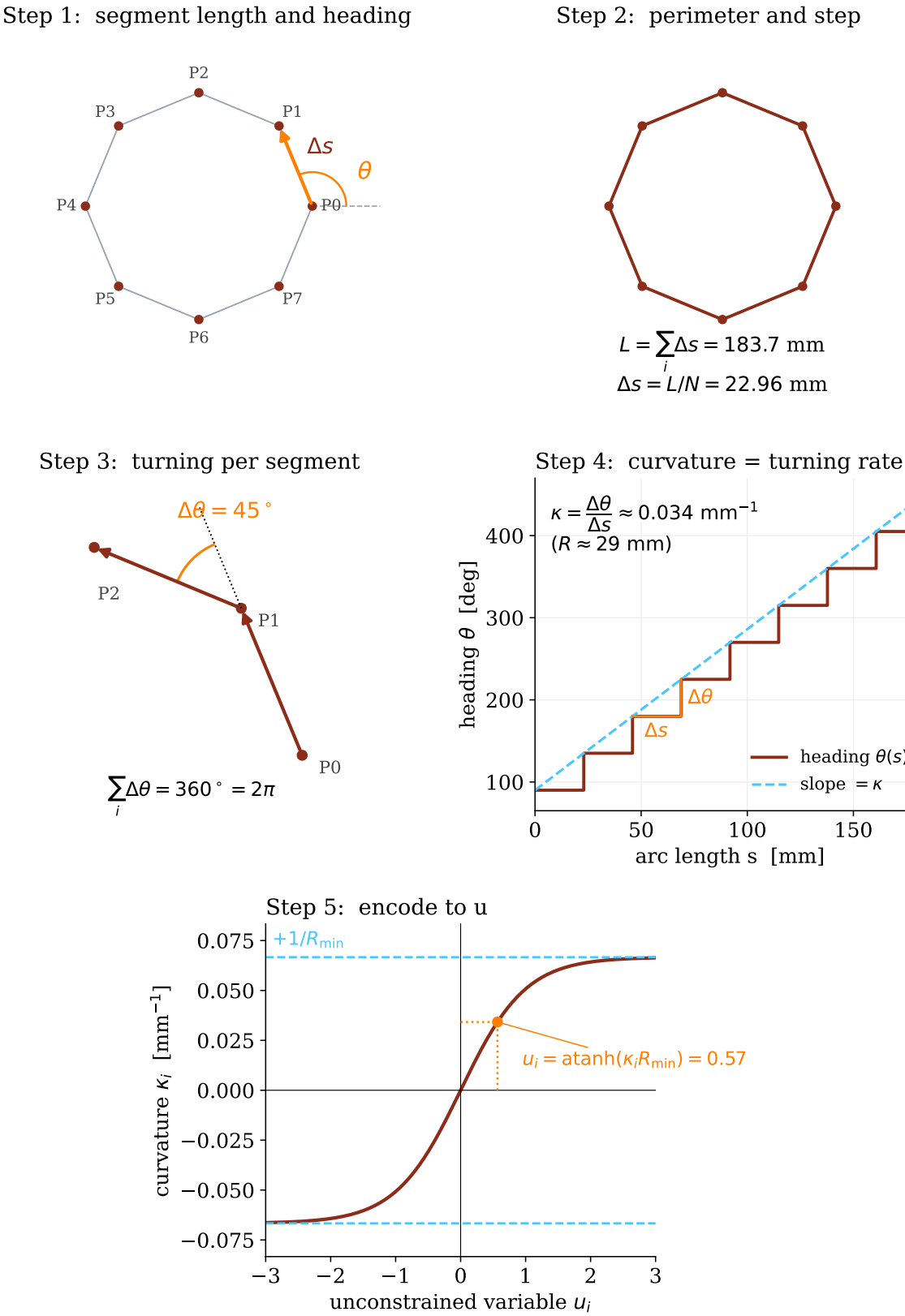


Figure 4.2: Constructing the design vector from a starting shape, worked on a circle of radius 30 mm cut into $N = 8$ segments.

4.2 Geometry reconstruction

The forward map is the seeding of Section 4.1 run backwards, rebuilding the boundary from the design vector on the same worked circle. Figure 4.3 traces this, the inverse of Figure 4.2, on the same circle of radius 30 mm cut into $N = 8$ segments, in five steps:

- **Step 1 - decode to curvature and step.** Each variable is decoded to a curvature through Equation 4.1 and the stored length fixes the common step $\Delta s = L/N$.
- **Step 2 - accumulate the heading.** The curvatures are integrated to the segment headings, $\theta_i = \theta_0 + \sum_{j \le i} \kappa_j \, \Delta s$.
- **Step 3 - heading to displacement.** Each heading becomes a displacement by the midpoint rule,

$$\Delta x_i = \Delta s \cos\bigl(\bar{\theta}_i\bigr), \qquad \Delta y_i = \Delta s \sin\bigl(\bar{\theta}_i\bigr), \tag{4.2}$$

  with $\bar{\theta}_i$ the midpoint heading of the segment.
- **Step 4 - place the nodes.** The displacements are chained head to tail, each starting where the last one ended, to place the nodes.
- **Step 5 - close the loop.** The chain closes on itself, $\sum_i \Delta x_i = \sum_i \Delta y_i = 0$.

The reconstructed section, the exact inverse of the seeding in Figure 4.2, is where the crush force F and the perimeter P are evaluated. The whole reconstruction is a chain of differentiable operations, so gradients of any functional of the geometry flow back to $(u, \log L)$ by automatic differentiation.

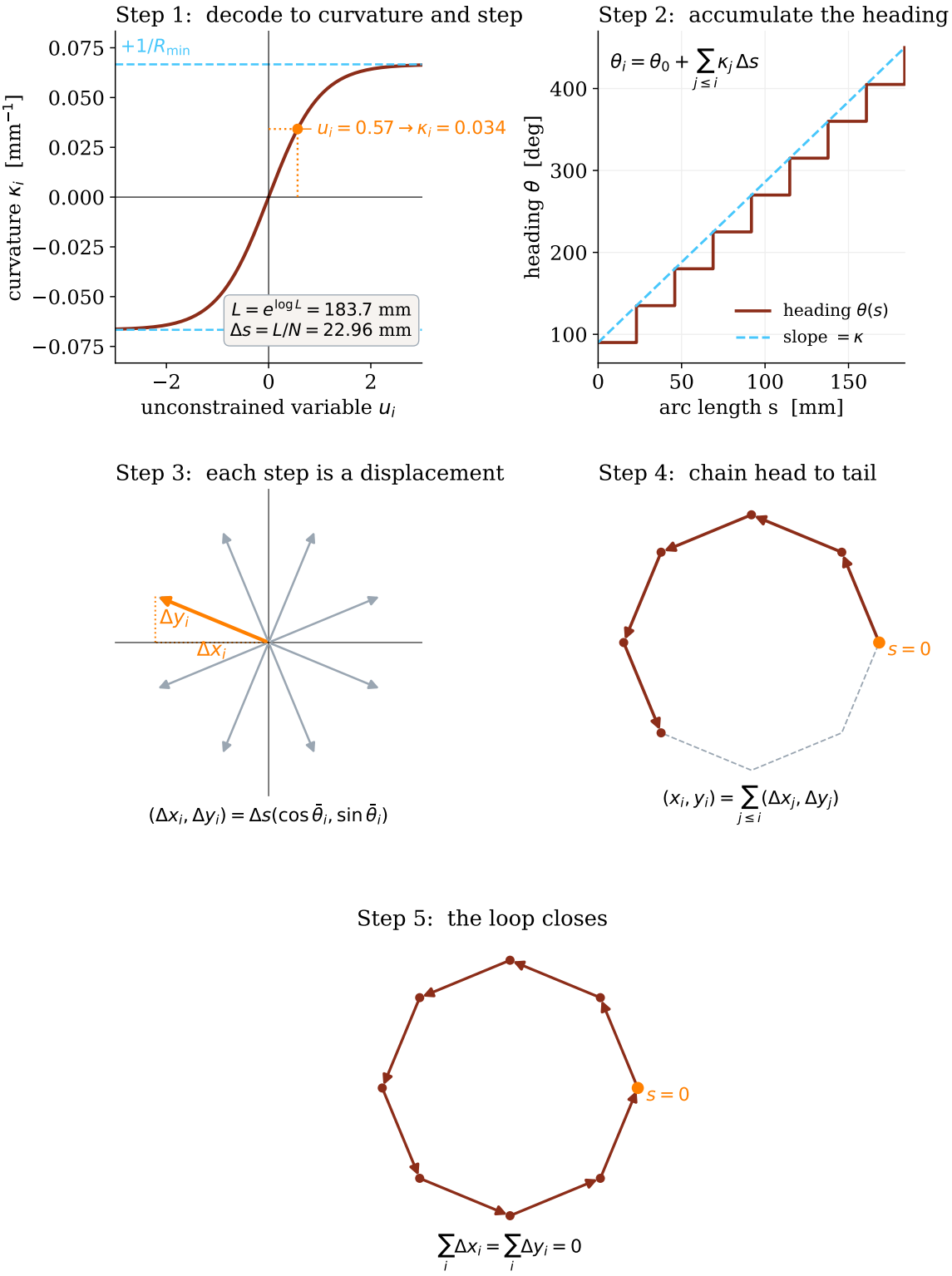


Figure 4.3: Reconstructing the boundary from the design vector, the inverse of Figure 4.2, on the same circle of radius 30 mm with $N = 8$ segments.

4.3 Constraints and loss function

A curve reconstructed from an arbitrary curvature distribution does not automatically close. Closure and total turning requirements,

$$\sum_i \Delta x_i = 0, \qquad \sum_i \Delta y_i = 0, \qquad \sum_i \Delta\theta = 2\pi, \tag{4.3}$$

are imposed as quadratic penalties, and the force requirement as a quadratic penalty that acts only when the force falls below the target, yielding the loss

$$\mathcal{L} = L + \lambda_F \left[\max(0,\, F_\mathrm{t} - F)\right]^2 + \lambda_{xy}\left(c_x^2 + c_y^2\right) + \lambda_\theta\, c_\theta^2, \tag{4.4}$$

where c_x, c_y are the closure residuals, c_θ the turning residual, and F is evaluated with the analytical model of Section 3.1.2 using $R_i = 1/|\kappa_i|$ clamped to $[R_\mathrm{min}, R_\mathrm{flat}]$.

The three penalty weights are deliberately large relative to the perimeter objective, so that the constraints behave as effectively hard while the perimeter is reduced inside the feasible region. A design is declared feasible only when every constraint holds at once within a tolerance: the closure residuals satisfy $|c_x|, |c_y| \le 10^{-2}$ mm, the turning residual $|c_\theta| \le 10^{-3}$, and the force clears its target up to a relative slack, $F \ge (1 - 10^{-3})\, F_\mathrm{t}$. These weights and tolerances are the defaults of the command line interface and are listed in full in Appendix A.

4.4 Optimisation strategy

The loss of Equation (4.4) is minimised with the Adam optimiser [61] under a cosine annealing learning rate schedule [62], so the search explores freely at first and settles into a basin as it cools.

The momentum coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$, the numerical constant $\epsilon = 10^{-8}$ and the zero weight decay are kept at their library defaults; the settings tuned for this problem are collected in Table 4.1.

The learning rate is initially set to 0.05, which is larger than the standard Adam default. A relatively large initial learning rate therefore allows the search to explore the loss landscape quickly, rather than moving through it too slowly. The learning rate is then reduced using a half-cosine schedule to a minimum value of 5×10^{-5}. The larger early learning rate helps the optimiser move between possible basins, while the smaller final learning rate allows the perimeter to settle more carefully into the minimum, without Adam noise pushing it back above the constraints.

Each section update is also controlled by gradient clipping. If the norm of a section gradient is greater than one, the gradient is rescaled so that its norm is equal to one before the update is applied. This helps stabilise the saturating map in Equation (4.1). When a segment is almost straight, its variable u_i lies far out on the flat tail of the hyperbolic tangent. In this region, even

a small change in curvature can correspond to a large change in u_i. Gradient clipping prevents such a change from moving the segment too far across the map in a single step, which keeps the optimisation stable.

The run stops early once the best joint score has failed to improve by more than 10^{-5} for 1500 consecutive steps, and in any case after a ceiling of 100 000 steps, a budget justified in Section 4.5.4. The full command line interface is listed in Appendix A.

Table 4.1: Adam optimiser settings tuned for the curvature formulation. The remaining hyperparameters (β_1, β_2, ϵ and the weight decay) are kept at their library defaults.

Setting	Value
Initial learning rate	0.05
Learning rate schedule	cosine anneal to 5×10^{-5}
Gradient norm clip (per section)	1.0
Maximum steps	100 000
Convergence patience	1500 steps
Convergence tolerance	10^{-5}

4.5 Characterisation of the formulation

Before the formulation is applied to the real structure, its behaviour is characterised on a single canonical seed. The stadium is used for this purpose, because it belongs to the same shape family as the SIS partition cross-sections and, as the seed sweep below confirms, converges to the global optimum. Studying one section in isolation settles, before any production run, what the converged section looks like, how long the optimiser must run, how much crush force a single section can carry, and whether the result depends on the starting shape. How the method then fares on the genuine geometry is taken up separately in Section 4.6. Throughout this section the target force is held at $F_t = 75\,\text{kN}$, the wall thickness at $t = 2.4\,\text{mm}$, the discretisation at $N = 128$ segments and the manufacturing bound at $R_{\min} = 15\,\text{mm}$.

4.5.1 The saturated optimum

Run on the stadium, the optimiser converges to a boundary anticipated by the saturation argument of Section 3.4. The result is a curve whose curvature alternates in sign and sits at the manufacturing limit $|\kappa| = 1/R_{\min}$ (Figure 4.4). Throughout this work the *saturated fraction* of a section is the share of its segments whose curvature magnitude lies within 1% of the bound $1/R_{\min}$; the saturating map of Equation (4.1) approaches the bound only asymptotically, so saturation is always measured to this tolerance. On this run the saturated fraction is 1: every segment of the converged boundary is pinned at the manufacturing limit, as expected.

The converged section measures 215.65 mm against a continuous lower bound of 215.20 mm from Equation (3.12).

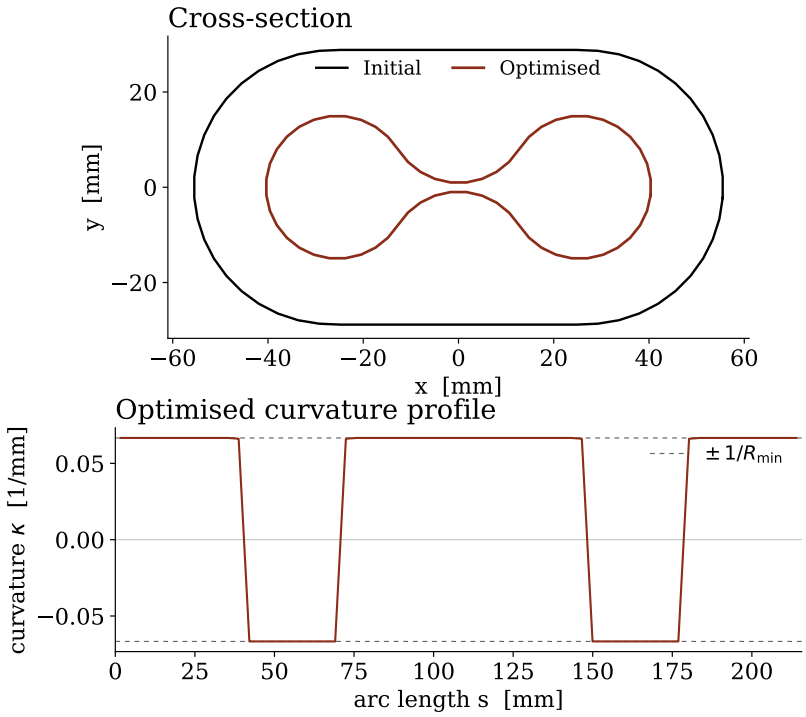


Figure 4.4: Optimised single section on the stadium seed, with initial and final geometry, and curvature profiles.

The optimiser reaches this section by descending the loss surface from the initial stadium, a path shown in Figure 4.6 and Figure 4.5.

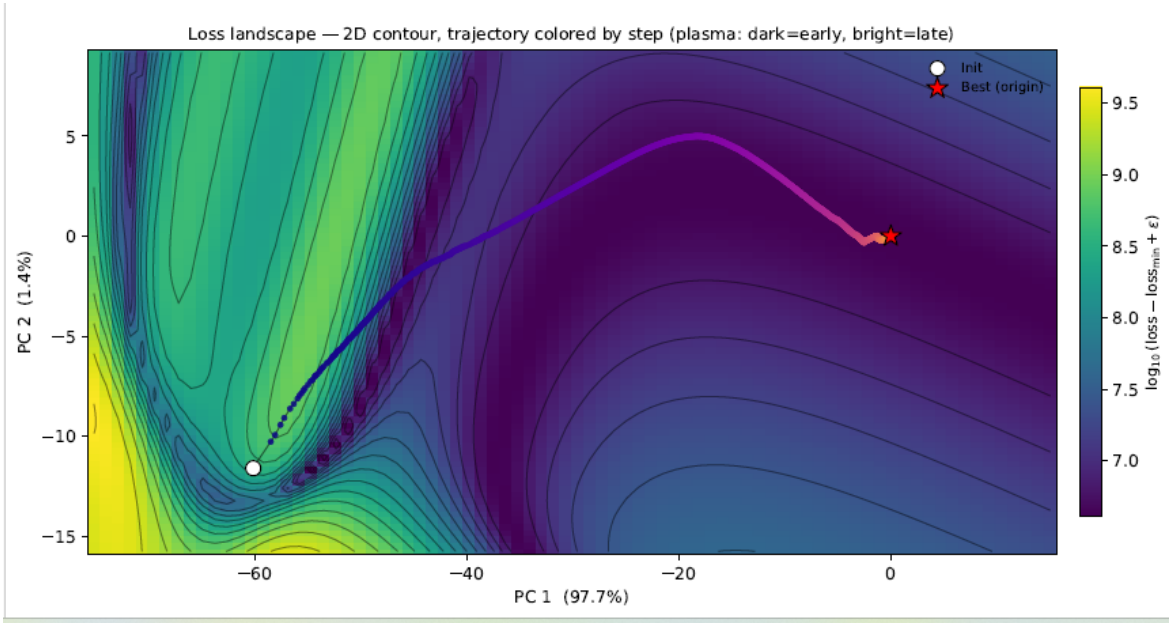


Figure 4.5: Loss landscape of the stadium run, a 2D contour with the descent path.

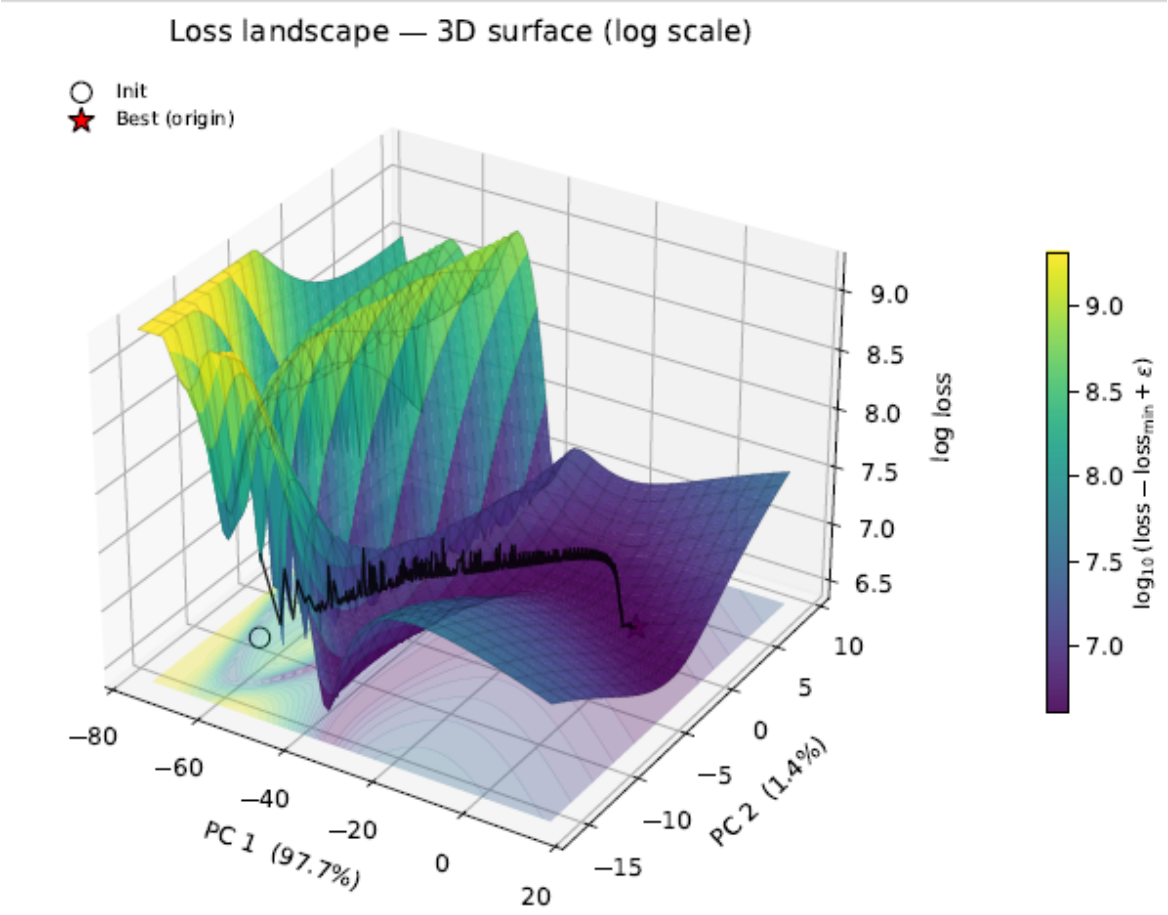


Figure 4.6: Loss landscape of the stadium run, 3D surface.

These two figures are *loss landscapes*: a map of the loss, the single number the optimiser tries to make as small as possible (Equation (4.4)), drawn over the space of possible sections, using a Principal Component Analysis plot. Every point on the map stands for one candidate section, and its height and colour give the loss of that section.

Figure 4.5 draws the result flat, like the height lines on a hiking map, with the path of the optimiser traced from start (dark) to finish (bright); Figure 4.6 draws the same surface in three dimensions as hills and valleys, with the path running over it. In both, the best section sits at the lowest point, and the optimiser is seen rolling straight downhill into that valley from the initial stadium.

4.5.2 Comparison with the radial optimiser

The gain the curvature formulation delivers is clearest when it is set against the optimiser it replaces. Both methods can be asked the same question on the stadium, the canonical SIS shape: minimise the perimeter of a section that carries roughly 70 kN at a wall thickness of 2.4 mm under a specified 15 mm minimum radius. The radial optimiser of Section 3.7.2 settles at 259.79 mm, a mere 2.64% below the stadium seed and still some 29% above the

continuous lower bound of 200.85 mm. The curvature optimiser is free to alternate the sign of the curvature, so from the same seed it folds the section into the corrugated optimum and reaches 207.11 mm, within 3.1% of that bound, with two thirds of its arc length pinned at the manufacturing limit. At a matched load, thickness and radius the curvature optimum is therefore 20.3% shorter in perimeter than the best the radial family can reach (Table 4.2).

Table 4.2: Radial versus curvature optimiser on the stadium seed at the matched target ($F_t \approx 70$ kN, $t = 2.4$ mm, $R_{min} = 15$ mm). The continuous lower bound on the perimeter is $L_{lb} = 200.85$ mm.

	Radial optimiser	Curvature optimiser
Seed perimeter L_{init} (mm)	266.83	264.89
Optimised perimeter L_{final} (mm)	259.79	207.11
Perimeter change ΔL (%)	−2.64	−21.81
Margin above floor L_{final}/L_{lb}	1.293	1.031
Saturated fraction	—	0.989
Final force (kN)	70.64	71.28
Feasible	Yes	Yes

In short, from the same seed and the same target both optimisers return a feasible section, but the curvature formulation reaches a far shorter perimeter, sitting just above the theoretical floor where the radial one stalls well above it. The chosen curvature optimiser is therefore the clearly better method for this problem.

4.5.3 Curve discretisation

The boundary is split into N equal arc segments, and N sets how finely the wavy optimum can be resolved: too few segments and the polygon cannot follow the tight alternating curvature, so the perimeter is left above its true value; too many and every optimisation step costs more for no further gain. The right value is found by repeating the stadium optimisation across a range of segment counts, from $N = 8$ to $N = 512$, with every other setting held fixed.

The converged perimeter falls steadily as the discretisation is refined and approaches the continuous lower bound of 215.20 mm (Figure 4.7): from about 226 mm at the coarsest count it drops to 215.85 mm at $N = 64$ and effectively reaches the bound by $N = 128$, at 215.65 mm, barely moving thereafter. The run time, by contrast, grows with the segment count, gently at first, from 163 s at $N = 8$ to 196 s at $N = 64$, and then more steeply, reaching 230 s at $N = 128$ and 436 s at $N = 512$ (Figure 4.8).

In short, the perimeter is fully resolved well before the finest discretisations while the cost keeps climbing with every added segment. A discretisation of $N = 128$ is therefore adopted for the rest of this work, fine enough that the section sits on its perimeter bound with margin to spare, yet still cheap to run.

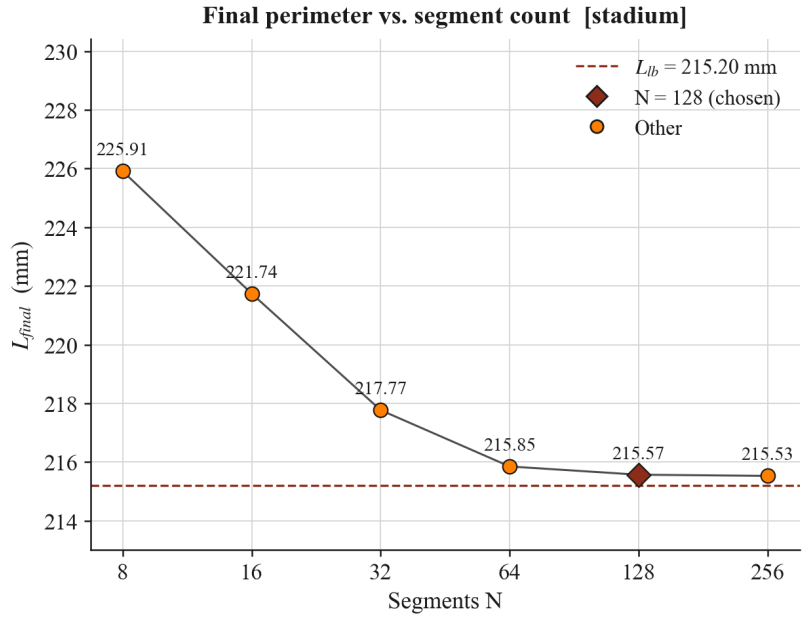


Figure 4.7: Final perimeter against the number of segments N on the stadium seed, with the continuous lower bound $L_{lb} = 215.20\,\text{mm}$.

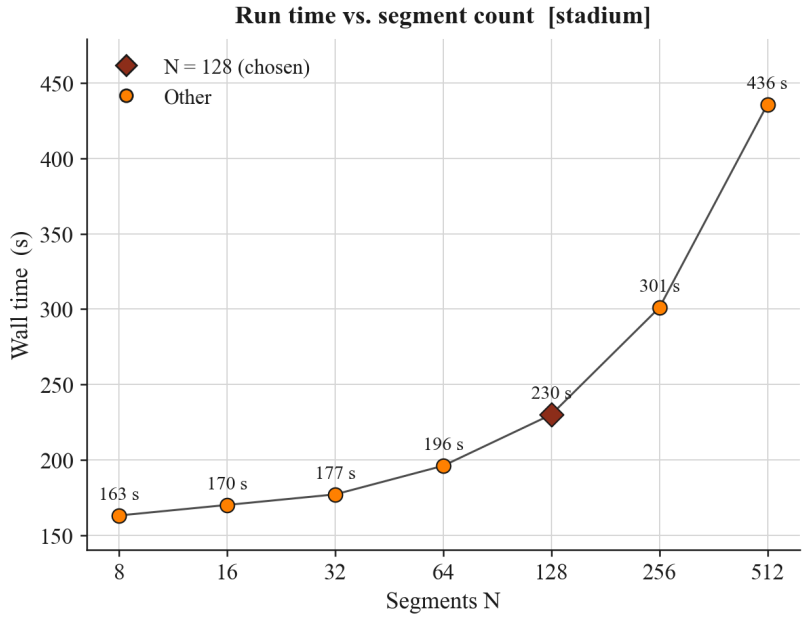


Figure 4.8: Run time against the number of segments N on the stadium seed.

4.5.4 Iteration budget

The number of steps the optimiser needs is found by tracking the converged perimeter against the number of Adam steps, with every other setting held fixed (Figure 4.9). The perimeter falls steeply at first and is within a fraction of a millimetre of the continuous lower bound by about 50 000 steps; past that the gains are negligible, the result at 500 000 steps lying

barely a tenth of a millimetre below the one at 100 000. The run time, by contrast, climbs in proportion to the step count, reaching 888 s at 500 000 steps (Figure 4.10).

In short, the perimeter is all but converged well before 100 000 steps while the cost keeps rising with every step added. A budget of 100 000 steps is therefore adopted as the default for the rest of this work, buying near-optimal sections for a few minutes of computation with only diminishing returns beyond it.

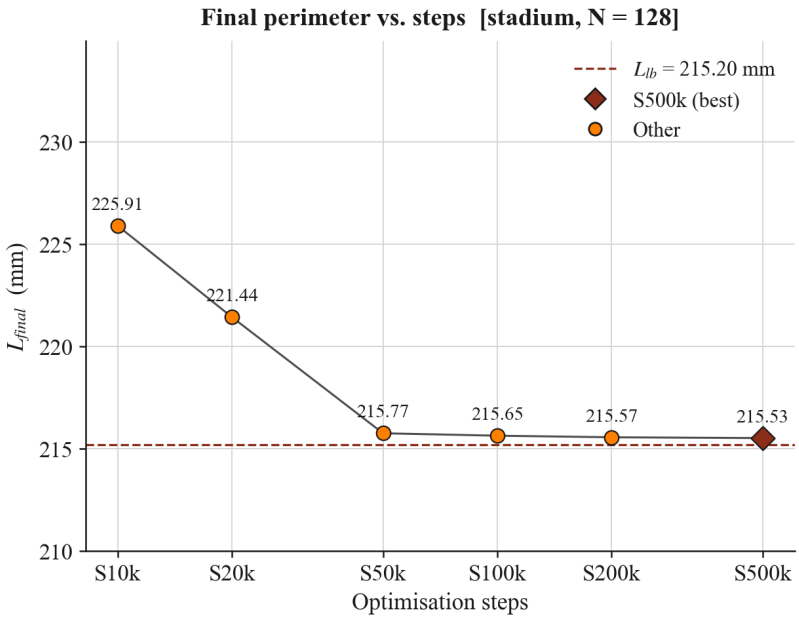


Figure 4.9: Final perimeter against the number of Adam steps on the stadium seed (N = 128), with the continuous lower bound L_{lb} = 215.20 mm.

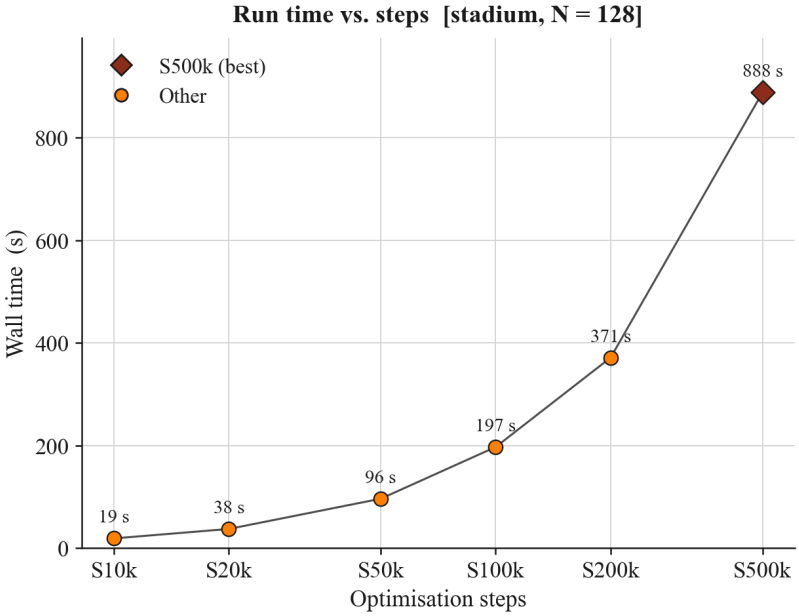


Figure 4.10: Run time against the number of Adam steps on the stadium seed (N = 128).

4.5.5 Force capacity

A single section can carry only so much crush force before the formulation can no longer return a valid shape, and sweeping the force target maps that envelope (Figure 4.11).

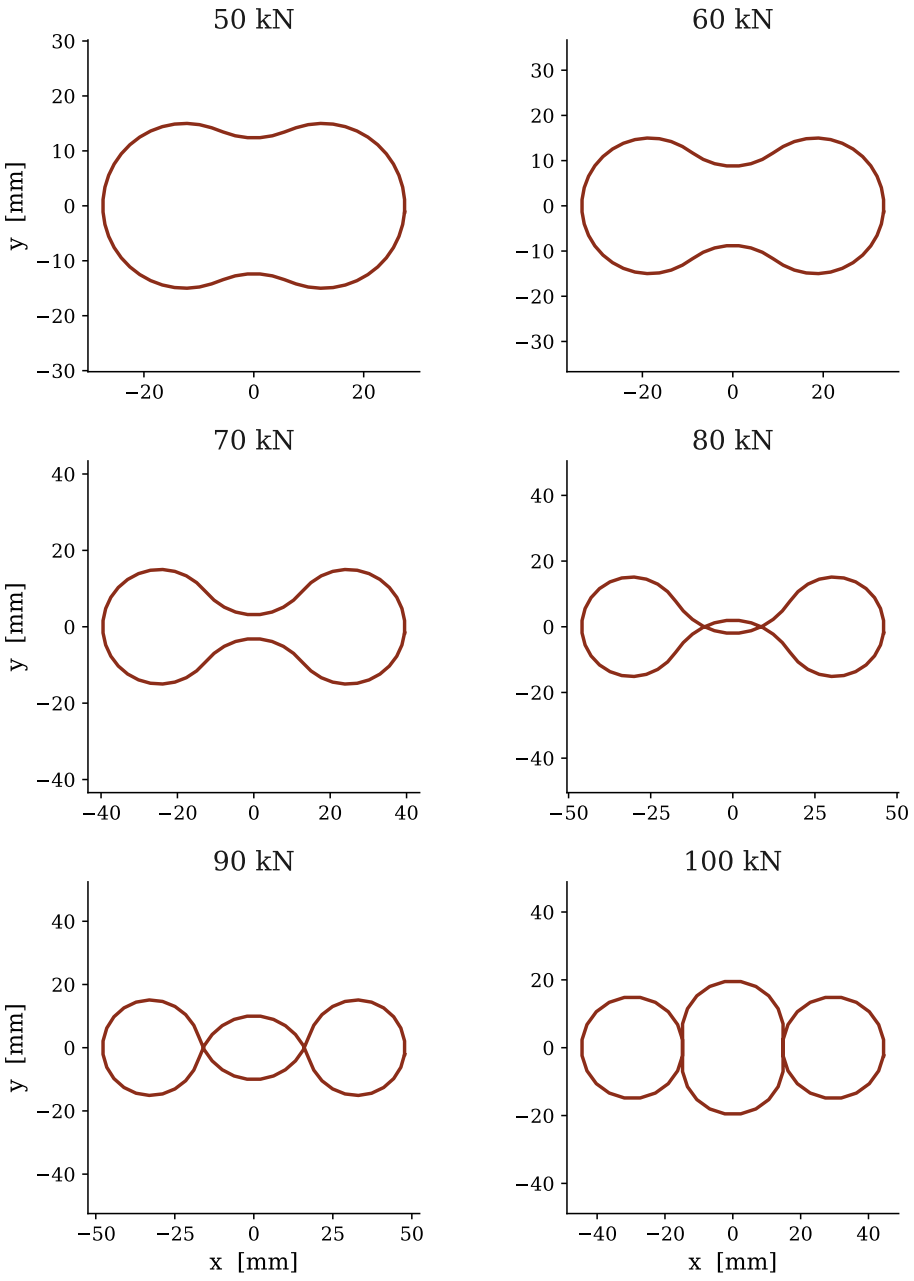


Figure 4.11: Force target sweep, stadium seed ($N = 128$).

As the target rises, the section grows additional lobes; by 90 kN there are three. The envelope ends where the lobes can no longer be packed into a simple closed curve. Beyond roughly 80 kN, at this wall thickness, adjacent lobes begin to touch and then cross, the saturated fraction collapses and the boundary self-intersects, so the result is no longer a physical section. The design target of 75 kN sits comfortably inside this envelope, with a clean section of two

lobes at the saturation bound. The optimiser flags an error whenever such non-physical sections are generated.

4.5.6 Convergence to a global optimum

A run starts from a seed shape: a closed curve resampled to N segments of equal arc length, from which the initial curvature, perimeter and heading are read into the design variables by the inverse map of Section 4.1. The seed fixes the basin the optimiser descends into, and because the loss is not convex a single run could in principle settle in a local minimum that depends on where it started (Figure 4.12).

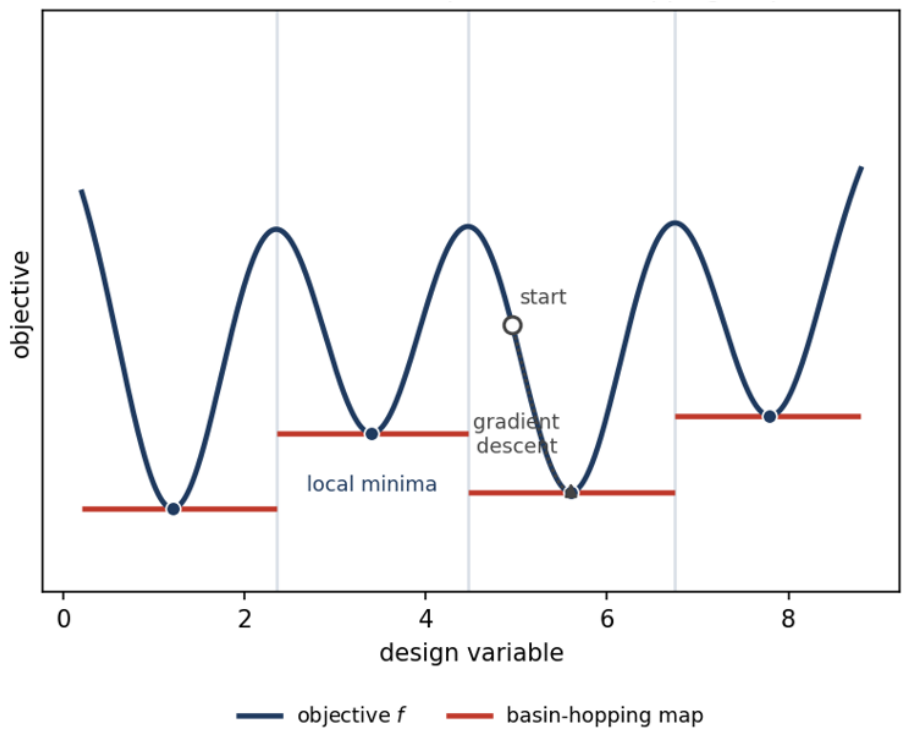


Figure 4.12: Multi-basin landscape and basin-hopping.

To test whether the result is instead the global optimum, the optimiser is run from a library of independent seeds, the perimeters and output geometry that they reach are then compared.

The library spans eight families (Appendix B): the circle, ellipse, stadium, square, rounded rectangle, blob, corrugated and Fourier shapes. A sweep is then run under identical settings (Table 4.3).

Table 4.3: Seed convergence study results ($N = 128$, $F_t = 75\,\text{kN}$, 100 000 steps).

Seed	L_{init} (mm)	L_{final} (mm)	ΔL (%)	F_{final} (kN)	L/L_{lb}	Sat. frac.	Time (s)
circle	287.27	215.87	−24.85	75.21	1.003	0.938	190.1
ellipse	288.20	215.76	−25.14	75.18	1.002	1.000	181.0
stadium	**287.89**	**215.65**	**−25.09**	**75.14**	**1.002**	**1.000**	**182.4**
square	301.76	215.80	−28.49	75.19	1.003	0.938	185.9
rounded_rect.	302.54	215.79	−28.67	75.08	1.003	0.938	182.1
blob	250.30	219.69	−12.23	75.00	1.021	0.938	191.6
corrugated	282.54	219.68	−22.25	75.00	1.021	0.938	190.4
fourier	282.18	215.88	−23.50	75.21	1.003	0.938	183.7

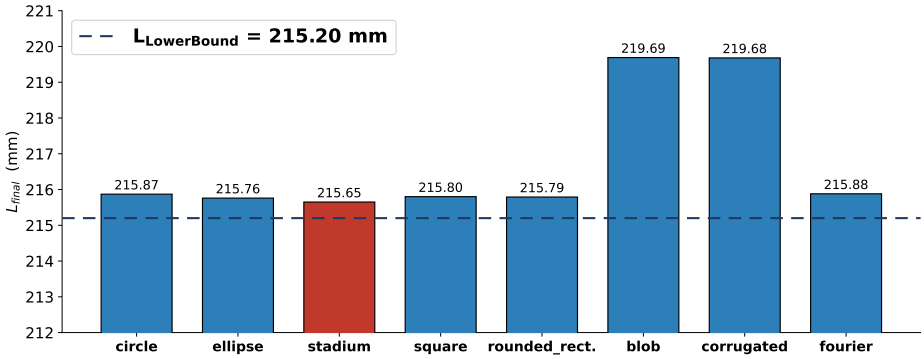


Figure 4.13: Final perimeter by seed ($N = 128$, $F_t = 75\,\text{kN}$, 100 000 steps).

Across the library the outcome is consistent (Figure 4.13): almost every seed converges to the same minimum perimeter and the same geometry, the strongest evidence available that the optimiser reaches the global optimum rather than a local minimum tied to its starting shape. For every seed counted as converged the perimeter settles to within a fraction of a percent of the theorised limit. The only exceptions are the blob and corrugated seeds, which settle at a visibly higher perimeter and so remain trapped in a local minimum; this confirms that such minima exist, while showing that from every other starting shape the optimiser escapes them. The full set of converged geometries, their initial and final curvature profiles are collected in Appendix B.

4.6 Application to the SIS cross-sections

The formulation has so far been exercised on idealised seeds, but its purpose is to redesign the cross-sections of the real structure. The natural first test is to take the parameterisation strategy developed through this chapter and apply it, unchanged, to the original SIS sections.

Each of the thirty-five partition loops is extracted from the structure using a single script,

`extract_sis_seeds.py`. The curvature formulation of Section 4.1 only describes a closed loop, so a partition that still carries its internal rib, a branched geometry with two junctions and three distinct walks between them, cannot be parameterised directly. The rib is therefore stripped before fitting. Removing the rib, while keeping the force it provided, has its own merits, since the rib itself adds weight. The same near colinear short edge cleanup that removed the partitioning artefact of Section 3.5 (Equation (3.13)) is reused here so that the seeds extracted are the best available rather than from raw, occasionally spurious, partition nodes. Each cleaned loop is resampled to N user-specified segments of equal arc length, seeded through the inverse map of Section 4.1 with its own initial curvature, perimeter and heading, and optimised independently under its own wall thickness and force target. Figures 4.14, 4.15 and 4.16 trace this process on a ribbed section at $z = -132.396$ mm, from the original FE partition, through the removal of the rib and short-edge cleanup nodes, to the smoothed and resampled optimiser seed.

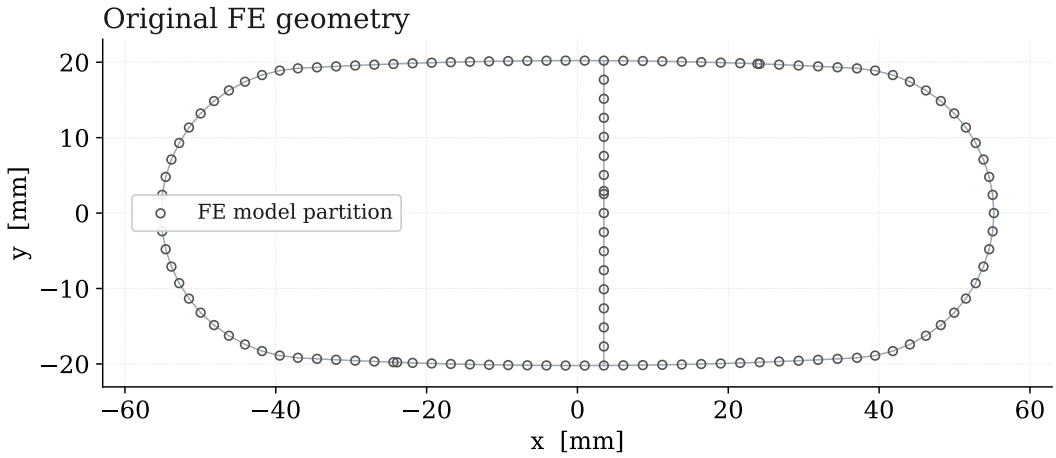


Figure 4.14: Seed extraction, stage 1: the original FE partition of a ribbed section at $z = -132.396$ mm.

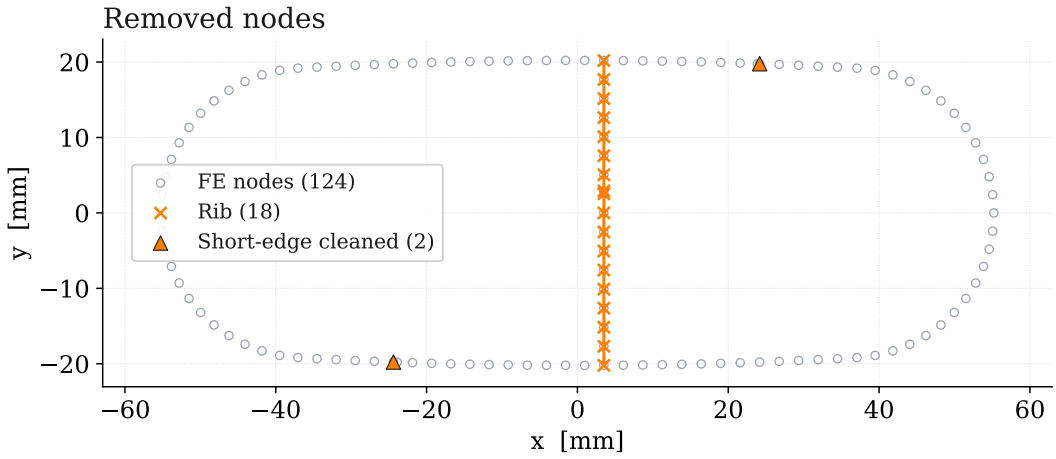


Figure 4.15: Seed extraction, stage 2: the rib (18 nodes) and the near-colinear short-edge nodes (2) flagged for removal.

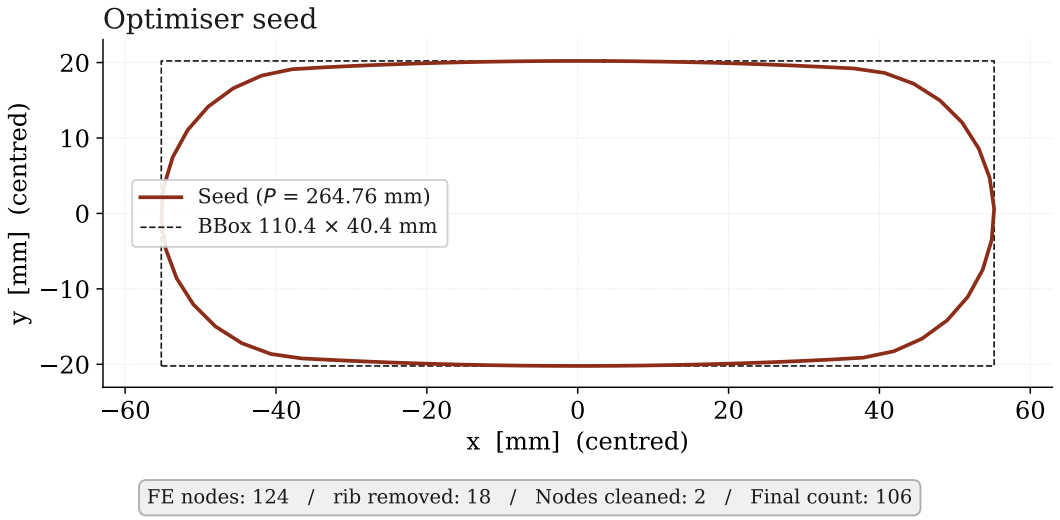


Figure 4.16: Seed extraction, stage 3: the cleaned loop smoothed, resampled to N segments of equal arc length and centred, giving the optimiser seed.

The script is described here for a single section, but it is not a one-off tool: the full-structure optimiser of Chapter 5 calls this same extraction internally for every section. In other words, the sections are optimised *in isolation*: each section is solved on its own, with no knowledge of the sections above or below it, and nothing yet ties the thirty-five sections into a manufacturable stack. The continuity between neighbouring sections, and the coupling needed to hold them in a common shape family, are the subject of Chapter 5. For now, it is shown that the curvature formulation, validated on canonical seeds, carries over to the irregular cross-sections the analytical tool actually produces. It is the first step towards optimising the full structure.

At $N = 128$ and 100 000 steps, every one of the thirty-five partitions converges to a feasible design. Each loop is extracted, seeded and optimised on its own, under its own wall thickness and force target, and in every case the formulation returns a closed, non-self-intersecting section that meets its force floor at the 15 mm radius limit.

The outcome of each section, with sections 1 to 35 ordered from the outbound end to the inbound end, is collected in Table 4.4, and the total perimeter falls from 8431 mm to 6384 mm, a reduction of 24.3%.

Table 4.4: Results of the independent single-section optimisation of the SIS partitions. Sections are ordered from the outbound end (section 1) to the inbound end (section 35).

Section	t (mm)	F_t (kN)	L_{init} (mm)	L_{final} (mm)	ΔL (%)	L/L_{lb}
1	2.40	31.5	98.49	94.28	−4.28	1.000
2	2.40	38.5	116.72	112.50	−3.62	1.018
3	2.40	40.0	123.47	117.61	−4.74	1.025
4	2.40	41.7	130.24	121.24	−6.91	1.013
5	2.40	43.4	137.00	125.77	−8.20	1.010
6	2.40	44.6	143.77	129.65	−9.82	1.013
7	2.40	46.2	166.97	136.21	−18.42	1.028
8	2.40	48.4	180.84	141.38	−21.82	1.018
9	2.40	50.6	191.82	148.05	−22.82	1.020
10	2.40	53.0	201.13	152.70	−24.08	1.004
11	2.40	55.3	209.32	161.40	−22.90	1.017
12	2.40	57.3	217.18	174.14	−19.82	1.059
13	2.40	59.2	224.89	176.34	−21.58	1.025
14	2.40	60.9	232.46	177.54	−23.62	1.016
15	2.40	62.5	239.94	183.50	−23.52	1.023
16	2.40	64.1	247.35	190.35	−23.04	1.035
17	2.40	65.7	254.69	192.93	−24.25	1.023
18	2.40	67.3	262.00	197.80	−24.50	1.024
19	2.40	68.8	269.27	202.42	−24.83	1.025
20	2.40	70.3	276.20	203.80	−26.21	1.010
21	2.40	71.6	282.38	211.47	−25.11	1.029
22	2.40	72.8	287.82	211.47	−26.53	1.012
23	2.40	73.8	292.55	213.01	−27.19	1.006
24	2.40	74.7	296.55	214.60	−27.63	1.001
25	2.40	75.4	299.84	218.75	−27.04	1.011
26	2.40	76.0	302.40	218.75	−27.66	1.003
27	2.40	76.4	304.24	219.92	−27.72	1.003
28	2.40	76.6	305.37	232.76	−23.78	1.059
29	3.00	94.6	305.10	219.34	−28.11	1.010
30	4.20	130.7	305.10	214.22	−29.79	1.000
31	4.20	130.7	305.10	214.53	−29.69	1.001
32	4.20	130.7	305.10	214.53	−29.69	1.001
33	4.20	130.7	305.10	214.53	−29.69	1.001
34	4.20	130.7	305.10	214.53	−29.69	1.001
35	4.20	130.7	305.10	214.53	−29.69	1.001

The fact that every section is feasible *in isolation* is then the limitation. Each of the thirty-five problems is solved on its own, with no knowledge of the adjacent sections, so nothing constrains a section to resemble its neighbours. Two consecutive sections will often both descend to the same peanut-shaped global minimum, yet settle at slightly different aspect ratios, or with a waist that is a little wider on one and narrower on the next, or with the lobes offset relative to each other. Figure 4.17 showcases one of these examples.

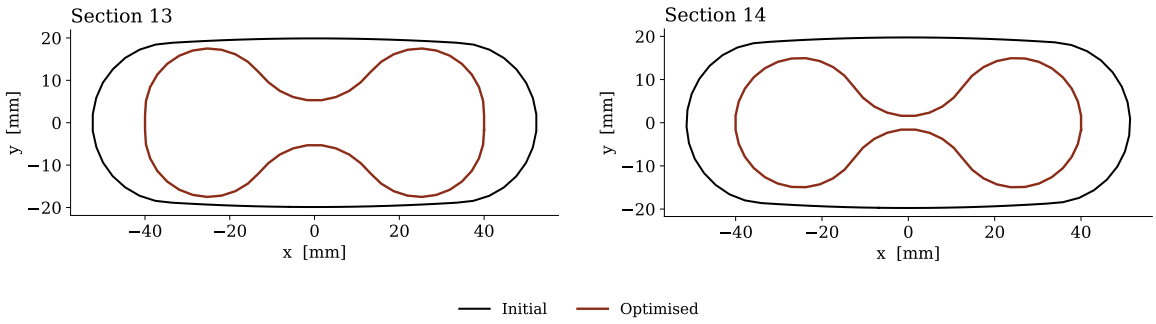


Figure 4.17: Section 22 and section 23 optimised shapes, showcasing the difference between two acceptable solutions.

Each of these sections is perfectly valid on its own terms, closed, feasible and free of self-intersection, but the structure does not vary smoothly and so could not be manufactured. Another example of this can be read directly from Table 4.4. For the current approach, the transition from section 28 to section 29 sees a decrease of perimeter instead of increase, then for the following sections some more decrease, until it finally goes back to being larger than the section before. This is visible in the assembled stack shown in Figure 4.18, produced with the finite element export pipeline presented in Section 5.5. While each section is the best section for its own load, with no link between sections the thirty-five best sections do not lead to one good structure. This is the concrete reason the full structure cannot be treated as thirty-five independent problems, and it is precisely what motivates the coupled formulation of Chapter 5, which ties the sections into a common shape family and enforces continuity between neighbours.



Figure 4.18: Isometric view of the stacked sections.

Chapter 5

Optimisation of the Full Structure

Chapter 4 closed with thirty-five cross-sections that, lofted along the crush axis, trace a surface that is not smooth and could not be manufactured. This chapter gathers those separate problems into one and solves the sections together. Section 5.1 stacks the thirty-five design vectors into a single tensor, so that one optimiser advances the whole structure in a single pass. Section 5.2 diagnoses the difficulty this exposes: sections commit early to different lobe families, and every change of family along the height surfaces as a kink in the lofted body. Section 5.3 presents the remedy, a warm start that initialises every section inside one chosen family. Section 5.4 then aligns the continuous stack to the reference face of the original Side Impact Structure, and Section 5.5 closes the design loop, turning the optimised curvature description back into a finite element model that the analytical tool can re-analyse.

The methods are presented by what they do rather than by the order in which they were written: only the smallest set of mechanisms that takes the independent sections to a single, continuous, manufacturable stack is described, and the command-line switches that expose them are collected in Appendix A.

5.1 Simultaneous optimisation of the multiple cross-sections

The starting point is the extraction of Section 4.6, applied to every partition instead of one. The same script strips the rib, cleans the partition nodes, fits the periodic spline and resamples each loop to N segments of equal arc length, and does so for all thirty-five sections, reading from `BigArray.mat` the wall thickness, force target and height station of each section along the crush axis. Each cleaned loop is seeded through the inverse map of Section 4.1 into its own design vector (κ, L, θ_0). The pipeline is summarised in Figure 5.1.

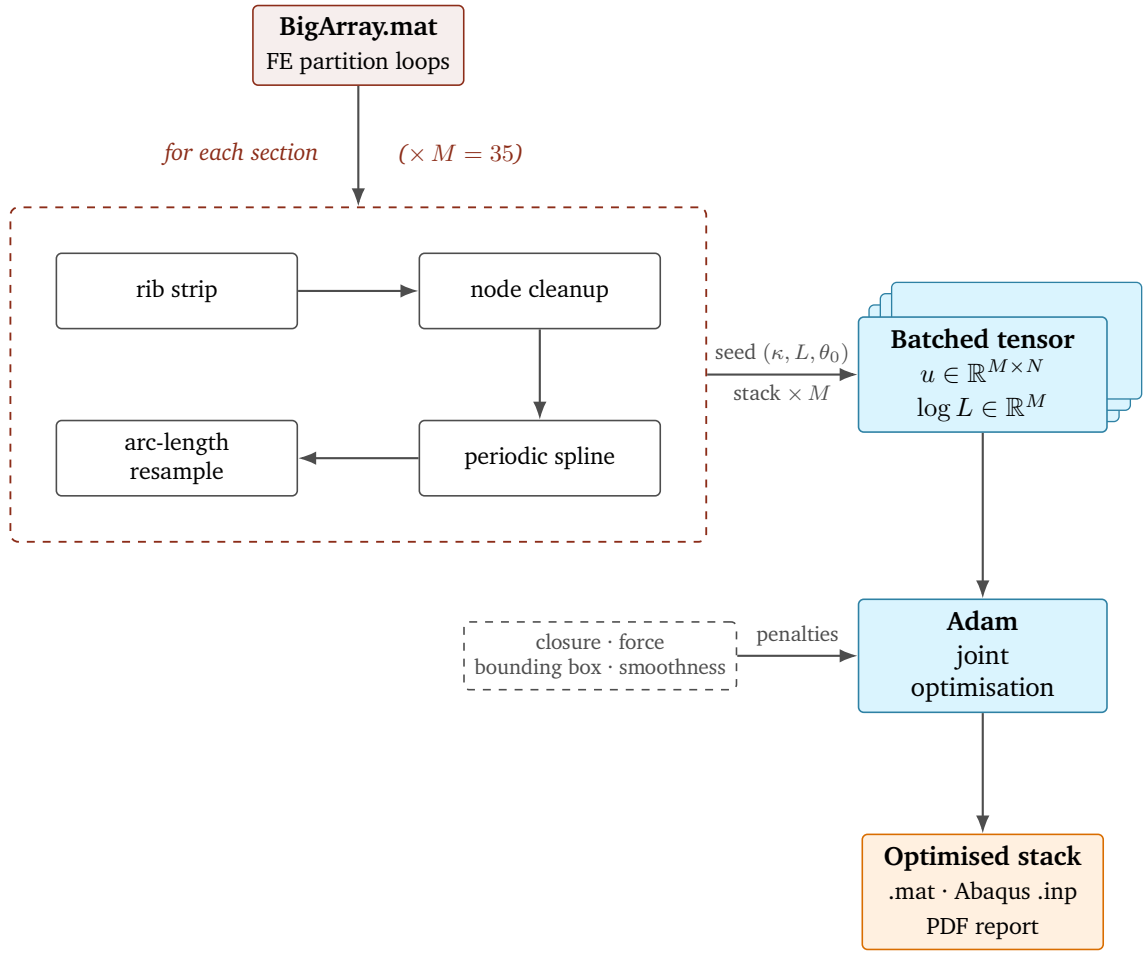


Figure 5.1: Multi-section seed-extraction and optimisation pipeline.

Rather than launch thirty-five separate runs, the $M = 35$ sections are stacked into a single design tensor: the segment variables become a matrix $u \in \mathbb{R}^{M \times N}$, one row per section, and the arc lengths a vector $\log L \in \mathbb{R}^M$. The geometry reconstruction of Section 4.2 and the force model of Section 3.1.2 are written to act on every row at once, and one Adam optimiser advances the whole stack together. Nothing in the loss yet links the rows, so the batched solve is, at this stage, pure vectorisation: it reproduces the independent optima of Section 4.6, all thirty-five feasible sections, but computes them in a single pass of a few minutes rather than thirty-five sequential ones.

The move to the full structure also motivated one new soft penalty. In the results of the single section optimisation, some sections exhibited a tight middle region of the peanut shape, despite not having self-intersection. To avoid poor manufacturability of those sections a soft penalty to the height of the middle section was imposed. If the points closer to the centroid are below a specified height the penalty is applied.

Two lessons about the batched solve itself are worth noting. The first concerns gradient clipping. The saturating map in Equation (4.1) can sometimes produce steps that are too large, so gradient clipping is useful. However, clipping the norm of the entire stack scales down

the update for every section by about $1/\sqrt{M}$. This unintentionally slows down all thirty-five sections. The solution is to clip the gradient norm separately for each section. The second lesson is about convergence detection. Stopping early based on the total perimeter of all feasible sections is unreliable, because the total changes suddenly whenever a new section becomes feasible. This can trigger early stopping. Instead, convergence should be checked separately for each section, using its own best feasible state.

5.2 Basins of attraction and consistency between sections

Two sections may carry the same load and reach the same perimeter while still converging to different *geometric* shapes. This deeper inconsistency is the main difficulty in extending the method to the full structure.

The source of the problem is the non-uniqueness of the corrugated optimum. A closed curve saturated at the curvature limit can distribute its total turning of 2π across different numbers of lobes. Each lobe count defines a distinct family of solutions. For example, an eight-lobe peanut and a twelve-lobe rosette may both be valid optima for the same sub-problem. These solution families are separated by barriers in the loss landscape.

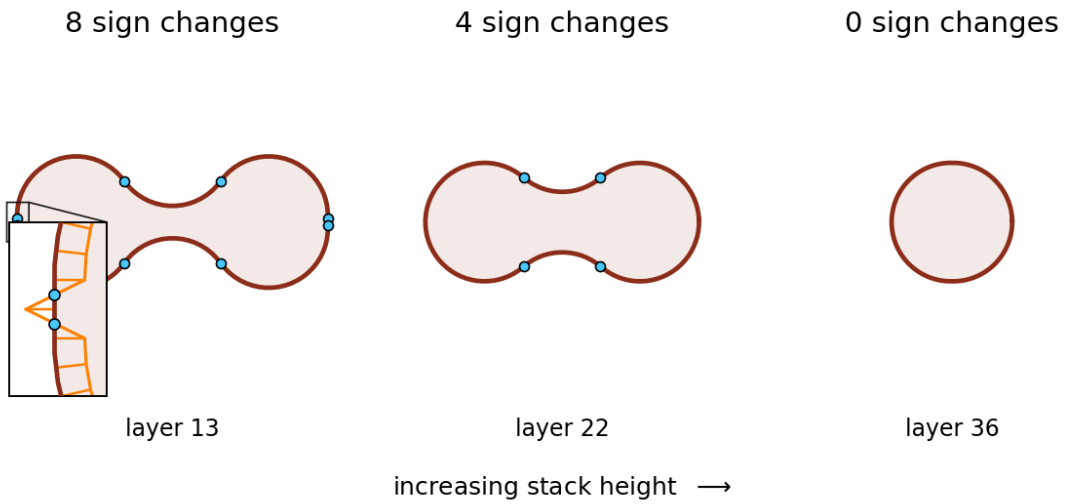


Figure 5.2: Distinct curvature lobe families in an uncoupled run.

The family selected by a section is usually determined early in the optimisation, during the infeasible transient. At this stage, the force and closure penalties dominate the loss, and the section commits to a lobe count while the curvature is still saturating. This commitment can occur so early that even identical sections may converge to different families if the phase of the resampling grid differs; that is, if a different node of the loop is treated as the first node.

When sections are considered independently, this issue is not apparent. However, when the optimised sections are read as a stack, it becomes a real problem. Curvature sign changes in an uncoupled run shows that the sections separate into several lobe families along the structure.

The lobe count changes from one band of sections to the next, as shown in Figure 5.2. If two adjacent sections belong to different families, their boundaries are geometrically incompatible. These changes in family are precisely where kinks appear in the lofted surface.

The objective for the remainder of the chapter is therefore not to improve each section in isolation. The individual sections are already close to optimal. Instead, the aim is to place the entire stack within a single solution family and to maintain that family consistently along the height.

5.3 Single-family warm-starting

The remedy follows directly from this diagnosis. If the lack of consistency arises because different sections commit to different families during the early transient, then all sections should be initialised within the same chosen family. The optimiser then does not need to cross barriers between families. Instead, it only adjusts the amplitude of a corrugation whose lobe count has already been fixed.

This is also the reason the original structure is smooth. It behaves as a single shape that varies continuously along the height, rather than as a collection of thirty-five independently selected shapes.

A converged section, which was handpicked because of good manufacturability from an earlier run is used as the *donor* that defines the target family. Each section is initialised by blending this donor with a plain rounded rectangle whose size matches the extents of that section. The blend is performed in curvature space:

$$\kappa_l(a) = (1-a)\,\kappa_{\text{rect}} + a\,\kappa_{\text{donor}}, \qquad L_l(a) = (1-a)\,L_{\text{rect}} + a\,L_{\text{donor}}, \tag{5.1}$$

with the blend fraction solved in closed form so that the initial perimeter matches the perimeter of the section itself $P_l^\star$, known from a prior uncoupled run,

$$a = \frac{P_l^\star - L_{\text{rect}}}{L_{\text{donor}} - L_{\text{rect}}}. \tag{5.2}$$

With this initialisation, the selected family is carried continuously through the thickness transitions, and the constant-thickness regions remain smooth. The assembled stack that results has a maximum inter-cross-section kink of 29.7°, concentrated at a few family transitions; in the final delivered stack this falls to 3.29° at a single tip section, with all other sections below 3° (Table 5.1).

Table 5.1: Maximum kink between adjacent sections of the assembled stack (before) and of the final delivered stack (after). Sections run from the outbound end (section 1) to the inbound end (section 35); the kink is undefined at the two end sections.

Section	Before (°)	After (°)	Section	Before (°)	After (°)
L1	–	–	L19	4.4	2.8
L2	17.1	3.3	L20	9.1	2.2
L3	2.1	2.9	L21	10.8	1.7
L4	1.2	1.1	L22	2.3	1.4
L5	1.2	1.6	L23	0.1	1.3
L6	3.6	3.0	L24	2.5	1.0
L7	2.1	1.5	L25	3.4	1.6
L8	1.9	1.9	L26	1.5	2.3
L9	2.7	1.3	L27	15.9	1.7
L10	5.3	1.6	L28	29.7	2.9
L11	4.3	0.6	L29	9.4	0.6
L12	15.6	2.5	L30	5.4	1.3
L13	4.1	1.9	L31	0.5	1.2
L14	4.5	1.8	L32	0.0	0.6
L15	3.0	1.2	L33	0.0	0.1
L16	6.9	1.6	L34	0.0	0.1
L17	3.0	1.2	L35	–	–
L18	0.6	1.6			

Maximum kink: 29.74° (before) vs 3.29° (after). **Total perimeter:** 6384.3 mm (before) vs 6347.3 mm (after).

To see whether the warm start genuinely holds the whole stack in one family, the converged curvature of every section is laid out as a single map in Figure 5.3. Each section contributes one horizontal row: its signed curvature around the boundary, plotted against arc-length fraction, so a change of lobe family appears as a break in the vertical banding. In the uncoupled run the bands fragment from one group of sections to the next, while with the warm start the corrugation pattern runs continuously from base to tip, interrupted only by the single marked family flip. One such flip is unavoidable: the section size and force target change so much from the inbound end to the tip that a single lobe count cannot remain optimal, or even geometrically admissible, over the whole range (Section 5.2). The warm start cannot remove this change of family, but it confines it to a single joint instead of letting flips scatter along the stack as they do in the uncoupled run.

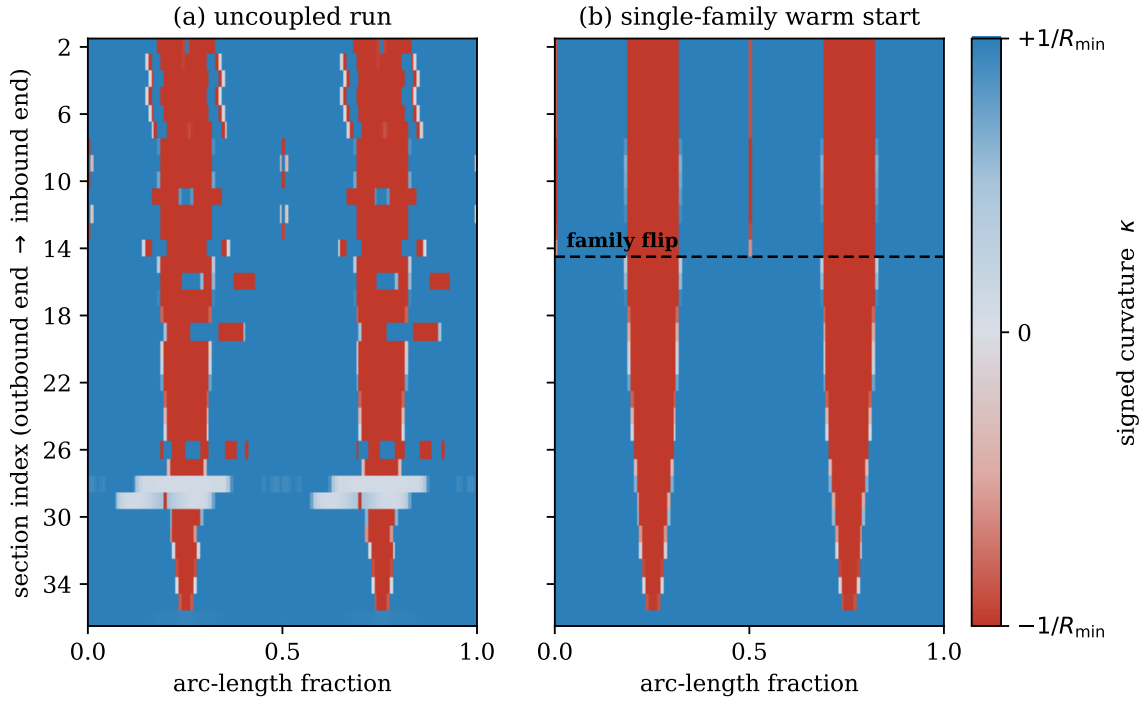


Figure 5.3: Curvature sign of the whole stack: rows - sections along the crush axis; columns - arc-length fraction.

Across these runs, the total perimeter remains within the same range as the independent optima reported in Section 4.6. The improvement in continuity is therefore achieved without sacrificing the perimeter reduction that motivated the optimisation.

5.4 Aligning the stack to the reference structure

The optimiser fixes the shape and size of each section but not its position: every section is solved about its own centroid, with no notion of where it sits in the assembled body. Because the sections are the SIS partitions of the original component, the natural choice is to assemble them the way that body is built. Rather than centring each section on its own centroid, the sections are aligned along one side, so that a single face of the stack follows the corresponding face of the reference SIS geometry. The optimiser returns each section centred on its own centroid. Each section is then translated in x until its left edge meets the reference face carried over from the original SIS partition, while the vertical centre is kept. Done for every section, with the offsets smoothed along the height, this places one face of the stack against the reference surface and throws the whole taper onto the opposite side, reproducing the one-sided body. The procedure is shown in Figure 5.4.

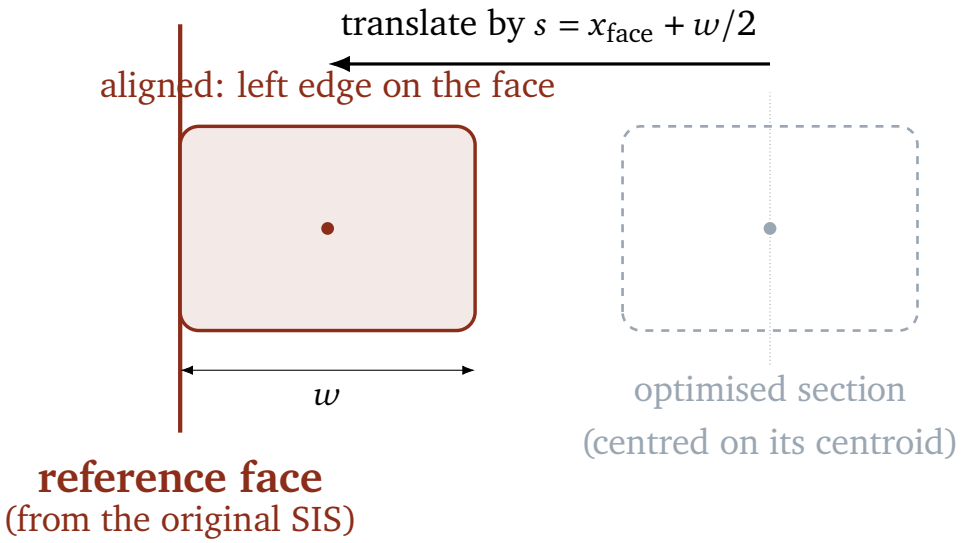
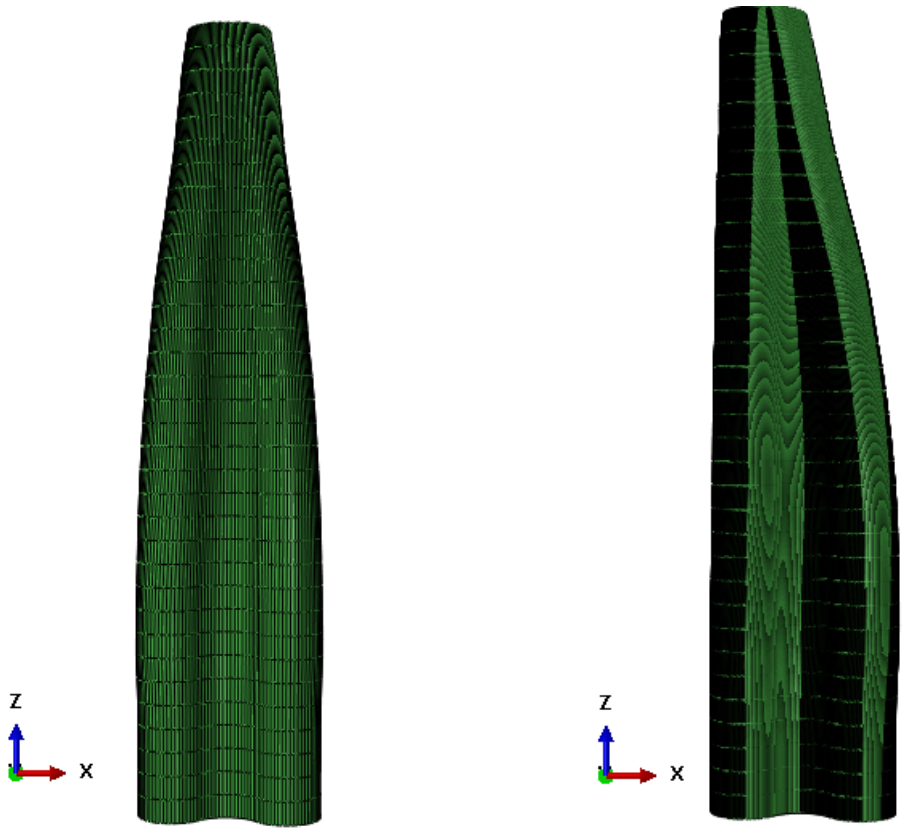


Figure 5.4: Aligning an optimised section to the reference structure.

Figure 5.5 shows the results of the described procedure applied to the structure.



(a) Centre alignment

(b) Side alignment

Figure 5.5: Alignment of the optimised SIS structure: (a) centre alignment; (b) side alignment, with the left face on the reference surface.

5.5 Requirements for validation

The optimiser is not a standalone script but one stage of a loop that begins and ends with a finite element model. Having produced a continuous, manufacturable stack, the last step is to turn its curvature description back into a model of the same kind that entered the analytical tool, so that the result can be re-tested. This is a crucial part of the work, because it makes it possible to validate whether the curvature discretisation behaves properly outside the optimiser.

5.5.1 The closed design loop

The full round trip runs through four file based stages and no manual geometry rebuilding at any point. The finite element model of the original structure enters CrushAnalytica, in the form of an .inp file. CrushAnalytica partitions it into cross-sections and exports the geometry, thickness and force data of each section as `BigArray.mat`. The optimiser consumes that file, replaces every section with a minimum perimeter equivalent through the mechanisms of this chapter, and writes the optimised assembly back out as a `.mat` file. The fields these files exchange are listed in Appendix C. A builder then generates an Abaqus shell model directly from that optimised curvature description. The loop closes. The consumer of that model is CrushAnalytica, which checks that the optimised sections carry the force the optimiser claims (Section 6.2). Every interface between the stages is a file, and every stage is scripted, so the structure travels the whole loop without a human redrawing a section (Figure 5.6).

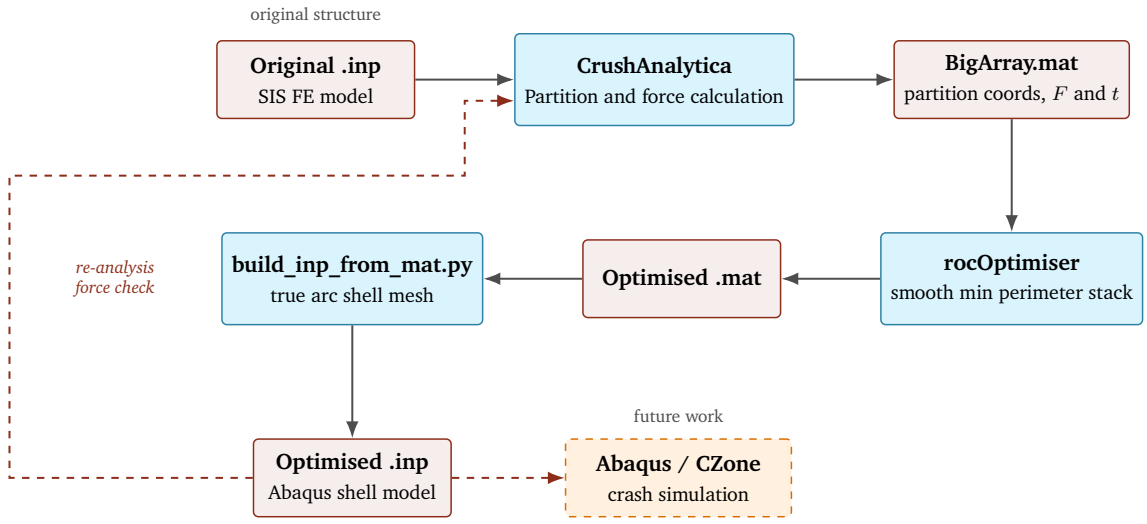


Figure 5.6: The closed design loop around the analytical tool: from the original finite element model to a verifiable optimised one.

5.5.2 Building the finite element model

The defining property of the builder is that it meshes the *true* optimiser geometry rather than a facetted copy of it. Intermediate nodes are placed on the arcs implied by the stored curvature

distribution, not by subdividing the straight chords between the nodes of the optimiser itself, so refining the mesh converges towards the optimised section instead of towards a polygonal approximation of it (Figure 5.7). In early testing it was discovered that a non-refined mesh would result in 10 – 15% loss of predicted force. The solution found was to quadruple the nodes of each section, which avoids the polygonal approximation and achieves the predicted force in CrushAnalytica.

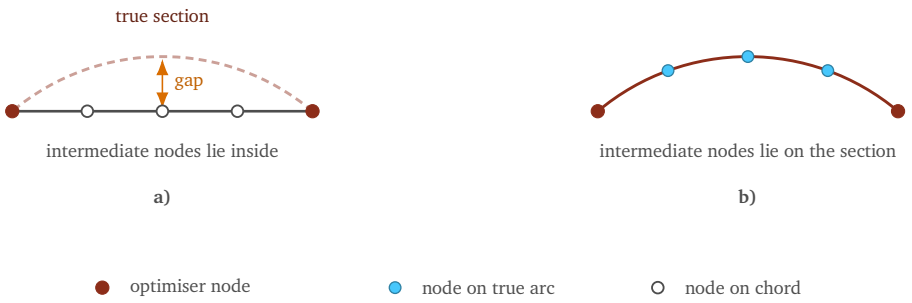


Figure 5.7: Circumferential mesh refinement. Intermediate nodes are placed on the true arc implied by the stored curvature (b), not by subdividing the straight chord between the nodes of the optimiser itself (a).

The choice of a fourfold refinement is backed by a convergence study. A representative optimised section was exported at refinement factors $r = 1$ to 5 and run in CrushAnalytica, and each resulting crush curve was reduced to its mean crush force and its peak force (Table 5.2). There is no monotonic convergence here: $r = 2$ actually comes out below $r = 1$ (70.03 kN against 71.74 kN), since the coarse meshes still have too few points around the section for the result to settle. It is only from $r = 3$ onwards that both measures rise towards the finest mesh. What matters is not the shape of that curve but how much is still changing at the end: going from $r = 4$ to $r = 5$ moves the mean crush force by only 0.8% and the peak force by less than 0.01%. The force is effectively mesh independent by $r = 4$, which gives the converged value at a fraction of the element count of the finest mesh, so $r = 4$ is used for every model built in this work.

Table 5.2: Mesh convergence of the CrushAnalytica force on a representative optimised section, exported at circumferential refinement factors $r = 1$ to 5. Deviations are measured against the finest mesh ($r = 5$).

Refinement factor r	Mean crush force (kN)	Deviation (%)	Peak force (kN)	Deviation (%)
1	71.74	−3.50	129.47	−3.26
2	70.03	−5.79	125.18	−6.47
3	72.19	−2.88	131.43	−1.80
4	73.72	−0.82	133.83	−0.00
5	74.34	0.00	133.84	0.00

Consecutive sections are then lofted with S4R shell elements, the same element used in the original SIS structure.

The wall thickness of each section is read from the `.mat` and used to group the elements into thickness bands; each band is assigned a composite shell section of round$(t/0.3)$ plies of the 0.3 mm fabric material, matching the layup convention of the original SIS model so that CrushAnalytica can read the generated model exactly as it reads the original.

Finally, a companion script imports the resulting `.inp` into Abaqus/CAE for inspection and downstream simulation setup, `import_inp_to_cae.py`. Figure 5.8 and Figure 5.9 display examples of an inspection on the CAE file.

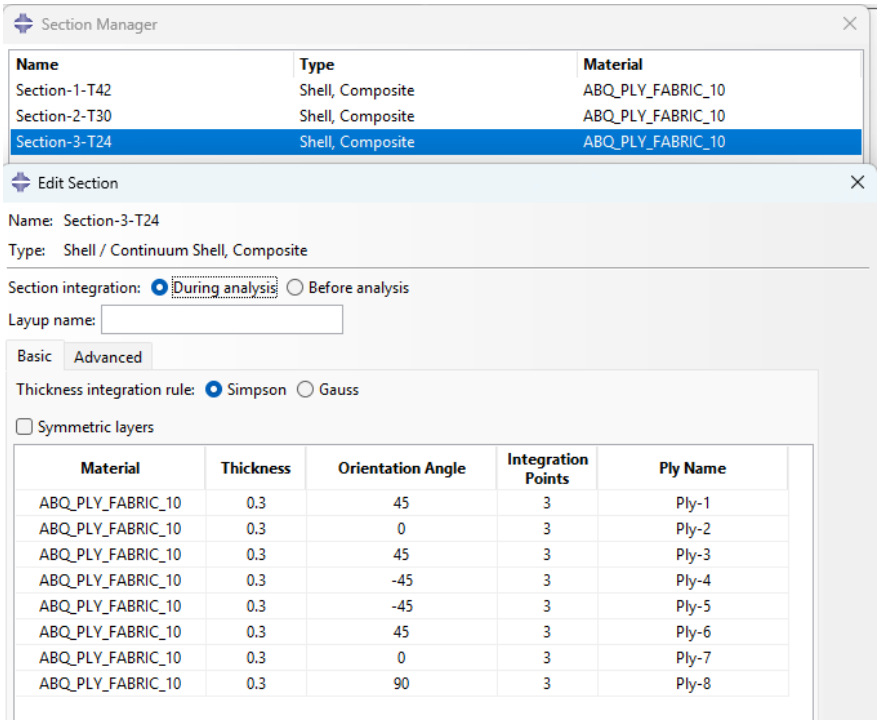


Figure 5.8: Abaqus section manager resulting from running the script, alongside the properties of one section.

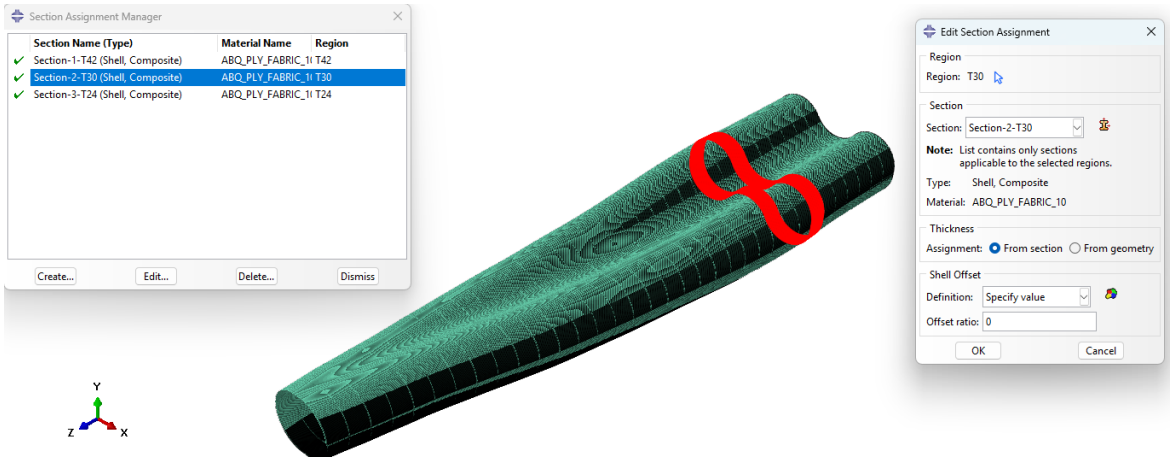


Figure 5.9: Abaqus section assignment example from running the script.

The concrete models this pipeline produced, their size and their fidelity check, are reported in Section 6.3.

5.6 Concluding remarks

The chapter has scaled the single section optimiser of Chapter 4 to the whole structure through a small stack of mechanisms. The central obstacle, that the section commits early to a corrugated design space split into lobe families it cannot later leave, is solved not by the penalties applied to curvature coupling and loft smoothness, which can only smooth within a family, but by single family warm-starting, which fixes the family by construction before the optimiser runs, leaving only the single discontinuity of the one unavoidable family flip.

The assembled stack is then aligned to the reference face of the original structure, throwing the taper onto one side as in the SIS body, and the export pipeline closes the loop back to a finite element model. Because the builder meshes the true arcs implied by the stored curvature rather than the straight chords between nodes, the generated model reproduces the optimised geometry closely enough to carry the force the optimiser predicts, recovering the 10–15% that a polygonal mesh would lose. Its quantitative assessment, against the theoretical bounds and against the re-analysis of the analytical tool itself, is the subject of Chapter 6.

Chapter 6

Results and Validation

The preceding chapters built the machinery; this chapter reports what it delivers. The focus is the final optimised assembly generated by the full-structure optimiser introduced in Chapter 5: a set of thirty-five cross-sections, all kept within one shape family and aligned to the reference face of the original Side Impact Structure.

The chapter first states the headline result, the perimeter saved and the continuity achieved (Section 6.1). It then closes the loop opened in Section 5.5.1: the optimised geometry is fed back through CrushAnalytica, and the force the reference tool reads from the finished structure is compared with the force the optimiser claims for it (Section 6.2). The finite element models generated from the result are documented in Section 6.3, the computational cost of the whole framework in Section 6.4, and the chapter ends with an assessment of how close the result comes to the theoretical bounds and of the limitations that remain (Section 6.5).

6.1 Final optimised assembly

The delivered structure meets every force requirement of the original. All thirty-five sections carry their crush force target: the final force sits between +0.10% and +0.70% above the target on every interior section, and +12.3% above it on the tip section, whose size is set by a geometric floor rather than by its force requirement (Section 6.5). This is achieved with substantially less wall length. In the original partitions, the thirty-five outer loops had a combined perimeter of 7385.5 mm. In the optimised assembly, the same loops measure 6347.3 mm, giving a reduction of 14.1% while maintaining the required crush force capability. If the internal rib walls removed by the redesign are also included, the total load-bearing wall length falls from 8430.6 mm to 6347.3 mm. This corresponds to a 24.7% reduction in wall material.

In mass terms, the most important conclusion of the work, the weight extracted from the finite element models of Section 6.3 falls from 0.479 kg for the original structure to 0.274 kg

for the optimised assembly: the redesign removes 0.205 kg of crushable mass at unchanged crush force capability.

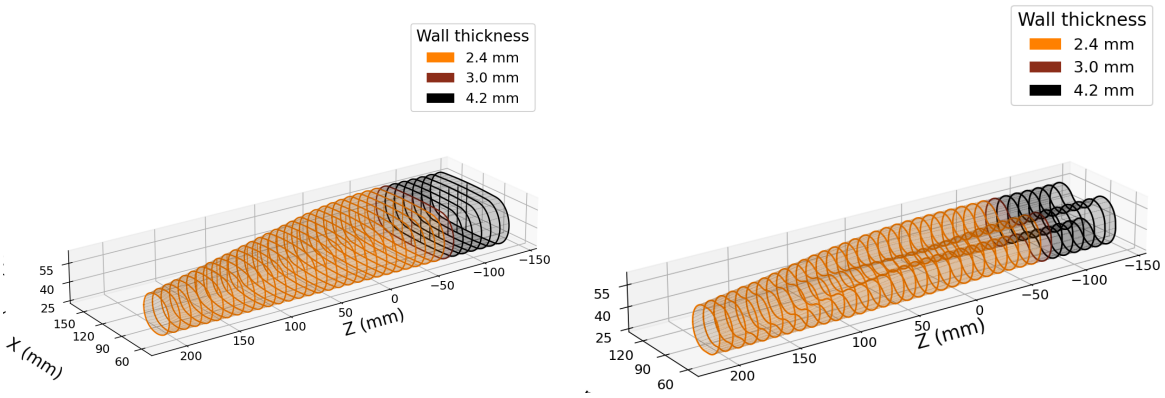
Table 6.1: The full structure result across the optimisation campaign. Perimeter reductions are quoted against the total load-bearing wall length (8430.6 mm).

Stage	$\sum P$ [mm]	ΔP	At force target	Max kink [°]
Original structure	8430.6	—	35/35	2.43
Independent optima	6384.3	−24.3%	35/35	29.74
Final delivered assembly	6347.3	−24.7%	35/35*	3.29

* The tip section exceeds its bounding-box height by 0.14 mm and is the only section flagged infeasible by the bookkeeping of the optimiser; its force clears the target by 12.3%.

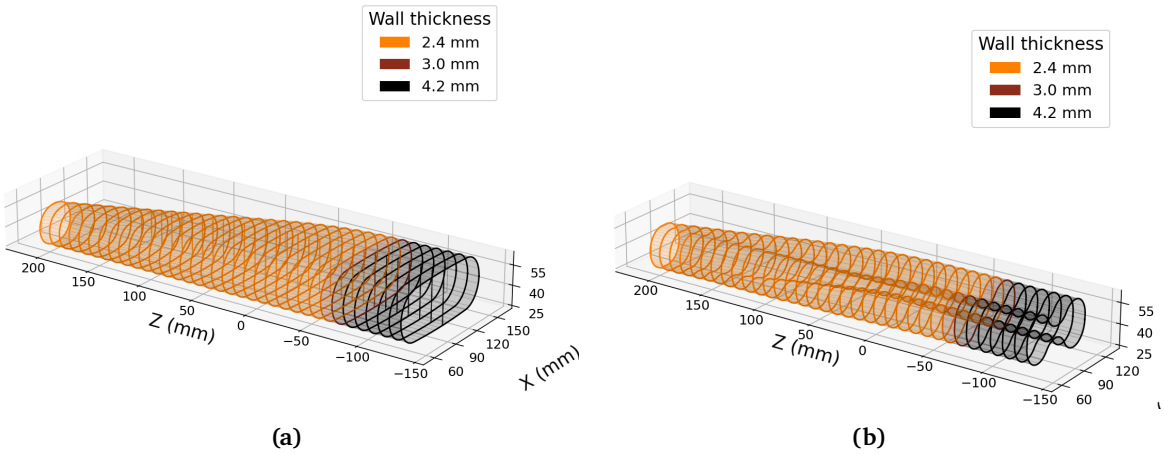
The result is also continuous. Measured with the loft angle metric of Section 3.2, with the same criteria used to derive the manufacturability allowable from the original structure, the maximum inter-cross-section kink of the assembly is 3.29°, with a mean kink of 0.90°. The original structure, a production geometry, sets the benchmark at 2.43°: six joints of the optimised stack sit marginally above that allowable, and none beyond 3.29°. For comparison, the stack of independently optimised sections presented in Section 4.6 has a maximum kink of 29.7°, with twenty-one of its thirty-three interior joints above the allowable. Table 6.1 summarises the progression.

The detail of each section leading to these results, force target against final force and original perimeter against optimised perimeter for all thirty-five sections, is collected in Appendix D. Graphically, the original and optimised stacks are compared under the two views of Figure 3.6 in Figures 6.1 and 6.2, and Figure 6.3 compares a single section of the two assemblies as read by CrushAnalytica.



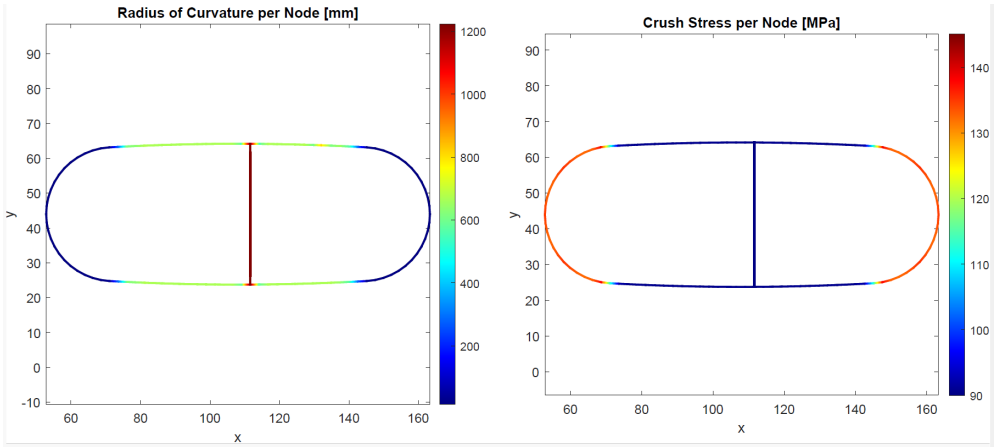
(a) **(b)**

Figure 6.1: Original (a) and final optimised (b) Side Impact Structure cross-sections, coloured by wall thickness, view from outbound end.

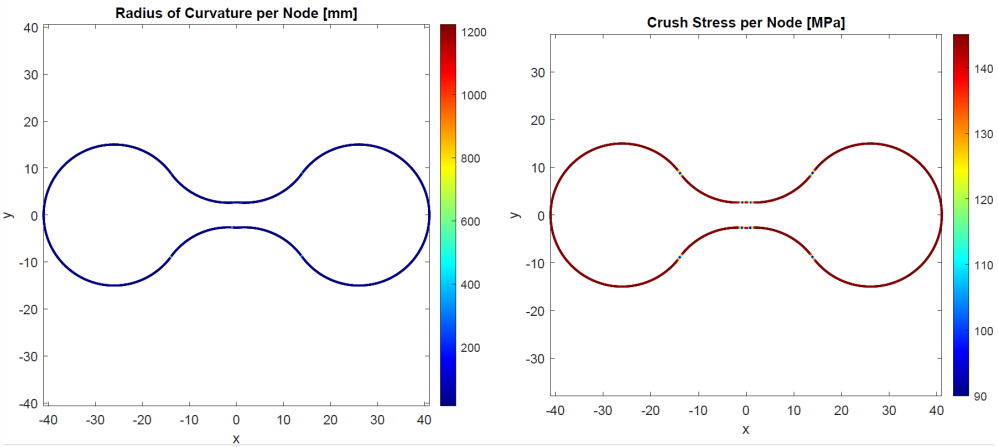


(a) **(b)**

Figure 6.2: Original (a) and final optimised (b) Side Impact Structure cross-sections, coloured by wall thickness, view from inbound end.



(a)



(b)

Figure 6.3: CrushAnalytica output for the same section in the different assemblies: (a) original SIS structure; (b) final assembly.

6.2 Verification through CrushAnalytica

The perimeter savings of the previous section are only worth reporting if the optimised sections truly carry the forces the optimiser claims. The strongest check available short of a crash simulation is the closed loop of Figure 5.6: build the finite element model of the optimised structure, hand it to CrushAnalytica, the same tool that produced the force targets from the original geometry in the first place, and let it partition, reconstruct and integrate the crush force with no knowledge of how the geometry was generated. If the analytical tool reads back the forces the optimiser designed in, the loop closes.

The results for a run using the final assembly are displayed in Table 6.2.

Table 6.2: Closed loop force check: the re-analysis by CrushAnalytica of the generated model against the force accounting of the optimiser itself.

	Final assembly
$\sum F$ CrushAnalytica [kN]	2506.6
$\sum F$ optimiser [kN]	2473.1
Total force deviation	+1.4%
Cross-section force RMS deviation	1.8%
Worst section force deviation	−4.1%
Reconstructed perimeter deviation	+0.04%

Table 6.2 shows that the loop closes: the total force read back is within 1.4% of the optimiser value, the scatter between cross-sections is 1.8% RMS with no station deviating beyond 4.1%, and the reconstructed perimeter matches to 0.04%. The force capability designed into the structure is confirmed by the independent implementation of the same physical model, applied to the finished finite element geometry rather than to the internal description of the optimiser.

Table 6.3: CrushAnalytica readings of the original SIS and of the final optimised assembly. Crushed mass, absorbed energy and specific energy absorption are the totals of the tool itself for each run.

	Original SIS	Final assembly
$\sum F$ CrushAnalytica [kN]	2464.4	2506.6
Force per wall length [N/mm]	299.5	405.0
Crushed mass [kg]	0.346	0.272
Energy absorption [kJ]	24.87	25.31
Specific energy absorption [kJ/kg]	73.2	91.5
Change in total force	—	+1.7%
Change in wall length	—	−24.7%
Change in specific energy absorption	—	+25.0%

The same closed loop also measures the redesign itself. Table 6.3 sets the CrushAnalytica reading of the original structure against its reading of the optimised one, and the two force-displacement responses are shown in Figure 6.4. The independent tool confirms the headline

of Section 6.1: the optimised assembly carries the full crush force of the original, in fact 1.7% more, with a quarter less wall, raising the force carried per millimetre of wall by 35%. The same reading holds in energy terms: the optimised structure absorbs essentially the energy of the original while crushing 21% less mass, lifting the specific energy absorption from 73.2 to 91.5 kJ/kg, a gain of 25.0%.

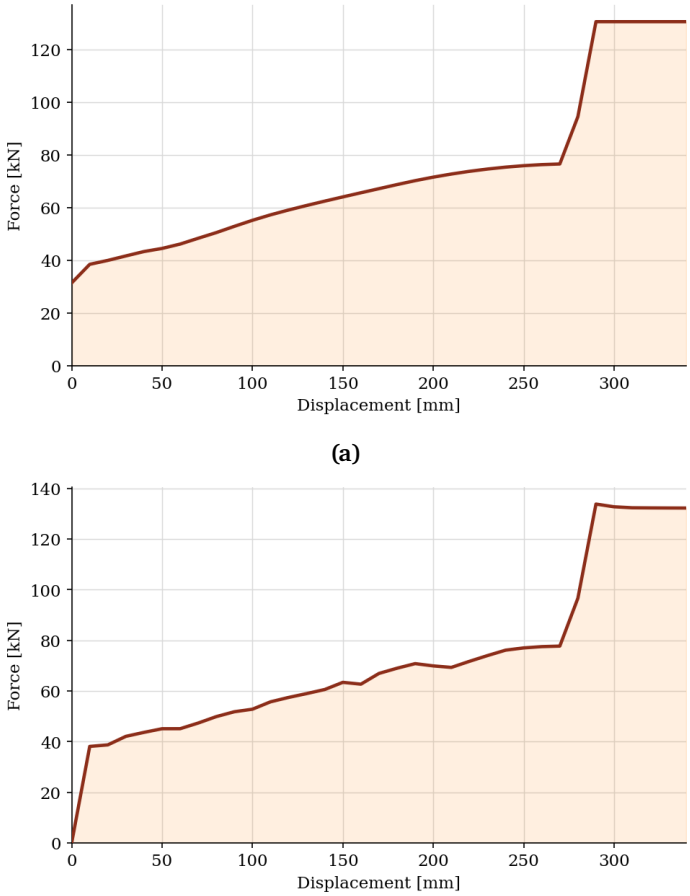


Figure 6.4: Force-displacement response read by CrushAnalytica for the original structure (a) and for the final optimised assembly (b).

Reaching this level of agreement required first identifying a systematic bias in the verification process, and that diagnosis is also a useful contribution. CrushAnalytica does not take curvature directly from the finite element model. Instead, it reconstructs the curvature by fitting a smoothing spline through the nodes of each sliced section and then differentiating that spline. The spline tolerance had originally been tuned for production geometries, where curvature changes gradually and small slicing artefacts need to be smoothed out.

The optimised sections are much more demanding for this approach. Almost the entire boundary lies at the 15 mm curvature limit. If the spline smooths the corners even slightly, it reports radii of 17–30 mm in regions where the true radius is 15 mm. Since the crush stress decreases sharply as radius increases, as described in Section 3.1.2, this curvature error alone

reduced the reported force by roughly 10% in the first verification attempts.

Two refinements closed this gap. The first was the arc exact circumferential refinement of the generated mesh, described in Section 5.5.2, which quadruples the node density available to the spline. The second was a corresponding tightening of the spline tolerance in CrushAnalytica. With both changes in place, the reconstruction captures the saturated corrugation correctly, as shown by the agreement in Table 6.2.

This sensitivity is an important finding. Spline-based curvature recovery is adequate for smooth production parts, but for geometries that deliberately saturate a curvature bound, it needs node spacings on the order of a millimetre before it can be trusted.

6.3 Generated finite element models

The concrete engineering output of the export pipeline of Section 5.5 is the finite element model of the final assembly used for the verification above, delivered in both centred and side-tapered positioning, the latter with every section aligned to the reference face of the original SIS as described in Section 5.4. Its mesh properties are collected in Table 6.4: the circumferential refinement places the intermediate nodes on the true arcs of the stored curvature description, and the composite shell sections match the layup conventions of the original SIS model.

Table 6.4: Mesh properties of the generated finite element model of the final assembly.

Property	Value
Element type	S4R shell
Nodes	17 920
Elements	17 408
Cross-section rings	35
Nodes per ring	512 (N = 128, refinement 4 × 128)
Ring spacing	10 mm
Shell section, 4.2 mm wall	14 plies of 0.3 mm fabric
Shell section, 3.0 mm wall	10 plies of 0.3 mm fabric
Shell section, 2.4 mm wall	8 plies of 0.3 mm fabric

The geometric fidelity of the export was checked directly: the perimeter of each refined ring reproduces the perimeter of each section computed by the optimiser to within 0.11 mm, and the assembly total to within 0.05%. The scripted companion import builds the corresponding `.cae` database with sections and assignments in place (Figure 6.5).

These models are also where the headline weight saving of Section 6.1 is measured. The mass properties of each mesh were extracted with the Abaqus mass properties query and are collected in Table 6.5: the optimised assembly weighs 0.274 kg against the 0.479 kg of the original structure, a saving of 0.205 kg. Nearly all of the saving is already present in the stack of independent optima; the continuity mechanisms of Chapter 5 give none of it back.

Table 6.5: Mass properties of the finite element meshes, extracted with the Abaqus mass properties query.

	Original	Independent optima	Final assembly
Shell face area [mm^2]	84 143.7	62 428.5	62 019.2
Volume [mm^3]	219 087.1	174 396.2	173 413.3
Total mass [kg]	0.479	0.276	0.274
Mass saving [kg]	—	0.203	0.205
Mass for 260 mm from outbound end [kg]	0.302	—	0.169

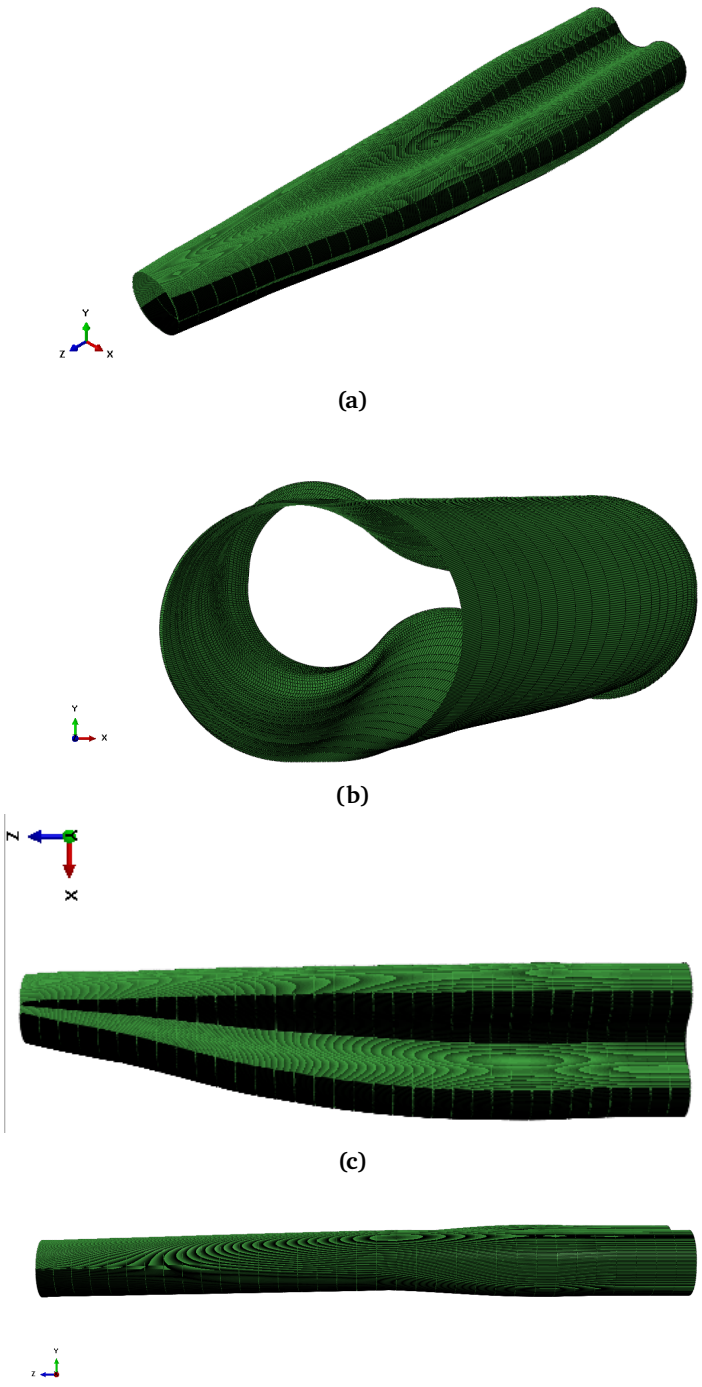


Figure 6.5: Finite element shell model of the final optimised assembly in Abaqus/CAE: (a) isometric view; (b) XY view; (c) XZ view; (d) YZ view.

Figure 6.6 closes the section with the three bodies this pipeline produced over the campaign, from the original structure, through the stack of independent optima, to the final delivered assembly.

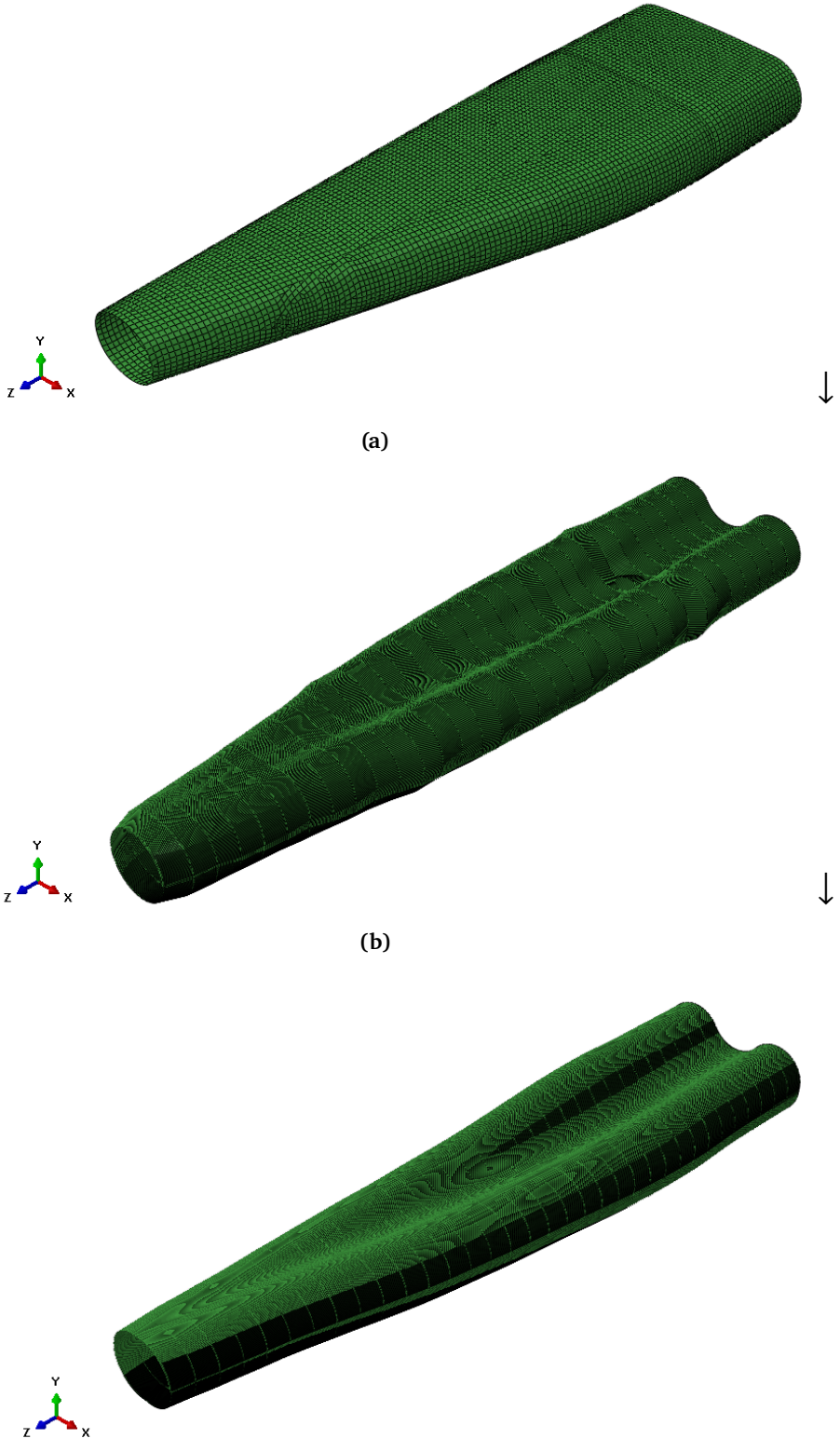


Figure 6.6: Progression of the structure across the optimisation campaign: (a) original structure; (b) independently optimised stack; (c) final delivered assembly.

The next step would be to test this model in the CZone crash simulation of the optimised structure, the definitive test of the redesign, but this is deliberately left as future work (Section 7.2).

6.4 Computational performance

The practical value of the framework rests on its speed, so the cost of the definitive runs is worth recording. All timings in Table 6.6 are wall-clock times from the run logs, on eight threads of a single sixteen-core desktop processor, with no GPU.

Table 6.6: Wall-clock cost of the definitive optimisation runs (desktop CPU, eight threads).

Configuration	N	Adam steps	Wall time
Single section (Section 4.5.4)	128	100k	$\approx$ 3 min
35 sections, independent	128	100k	9.2 min
35 sections, joint	128	100k	13.6 min

Two comparisons put these numbers in context. Within the framework itself, the step budget, not the section count, dominates: solving the thirty-five sections jointly at $N = 128$, with the coupling mechanisms, costs 13.6 minutes, under five times a single section run of the same step budget, because the joint solve advances all thirty-five sections per step rather than running them one after another. The full continuity machinery itself is cheap, adding less than five minutes over the uncoupled thirty-five-section solve.

Against the alternative the framework replaces: a single finite element crash simulation of a structure of this size runs for several hours (Section 1.2), so the entire redesign, from the partitioned original geometry to a verified minimum perimeter assembly, completes in a fraction of the time of one evaluation of the design it started from. It is this ratio that turns the analytical model from a calculator into a design tool.

6.5 Discussion

6.5.1 Distance to the theoretical bound

The saturation bound of Section 3.4 gives, for each section, the perimeter of a hypothetical boundary fully saturated at the manufacturing limit, Equation (3.12); summed over the stack it floors the structure at 6285 mm. The delivered assembly measures 6347 mm, 0.98% above that floor, and the interior sections individually sit between 0.1% and 2% above their own bounds.

What separates the result from the bound is the closure and total turning requirements, which force part of the arc length to turn the section around rather than accumulate force.

The one outlier is the tip section, at 17% above its force derived bound, and its excess has a different explanation: its target of 31.5 kN falls below the ≈ 33 kN that the smallest admissible closed shape, the circle at the minimum radius, already develops. For that section the binding constraint is the first bound of Section 3.4, not the second, and the force surplus of 12.3% reported in Section 6.1 is the geometric floor showing through.

6.5.2 Limitations

The claims of this chapter carry known limitations, listed here so that the conclusions of Chapter 7 can be weighed against them.

First, the physics. The analytical model is quasi-static and empirical: it assumes stable progressive crushing everywhere and encodes the material response in a fitted stress law. Nothing in the optimiser checks global or local buckling, and a corrugated thin wall is not obviously immune to either; the crash simulation of the generated model remains the necessary test. The verification of Section 6.2 is deliberately circular in the physics, as both sides implement the same crush model; it validates the geometry pipeline and the force accounting, not the model itself.

Second, the manufacturability metric. The inter-cross-section kink is a proxy, calibrated against a single production geometry. Genuine manufacturing constraints, draft, tooling access, ply drape over the corrugation, were not modelled. Also the tip section exceeds its bounding box by 0.14 mm. Both excursions are small, but a production release would need them cleaned up or formally accepted. The internal rib of the original structure was also removed on axial-crush grounds alone: any secondary function it served, such as stiffening the long walls or shaping the response under oblique loading, is not assessed here, and its study is deferred to the future work of Section 7.2.

Third, the validation status. The structure has been verified against the analytical tool and delivered as a simulation ready model, but it has not yet been crushed, in CZone or in a laboratory. Until it is, the 14.1% perimeter saving is a prediction of the analytical model, made twice by independent implementations, rather than a demonstrated property of the structure. The steps that would complete the demonstration are set out in Section 7.2.

Chapter 7

Conclusions and Future Work

This final chapter brings the work together. Section 7.1 returns to the objectives introduced in Section 1.3 and explains what the thesis delivered for each one. Section 7.2 then outlines the main directions in which the work could be continued, ordered by the value that each direction would add.

7.1 Conclusions

This thesis aimed to turn a fast analytical crush model into a practical design tool. The goal was to find, for a composite crash structure, the lightest closed cross-section that could still develop a required crush force, while respecting a manufacturing limit on wall curvature.

The research question was answered through a working design framework. The framework redesigned the complete Side Impact Structure, section by section and as a full assembly, cutting the total load-bearing wall length by 24.7%. Measured on the generated finite element models, the mass falls from 0.479 kg to 0.274 kg, a saving of 0.205 kg, and the specific energy absorption read by the reference tool rises by 25.0%, from 73.2 to 91.5 kJ/kg (Sections 6.1 and 6.2).

The main conclusions, assessed against the objectives of Section 1.3, are as follows.

A differentiable implementation of the analytical crush model makes gradient-based cross-section optimisation practical. The first primary objective was to formulate and solve the minimum perimeter problem using a differentiable implementation of the CrushAnalytica force model. This was achieved through the formulation in Section 3.3, solved by the PyTorch pipeline of Chapter 4 with the Adam optimiser. The computational cost is low enough for early design work. A single section with $N = 128$ segments converges in about three minutes, while the complete thirty-five section stack, with more than four thousand design variables optimised together, converges in 13.6 minutes on a desktop CPU (Section 6.4). Therefore, the full redesign takes much less wall-clock time than one finite element crash simulation of the

original structure.

The minimum perimeter section at a fixed crush force is corrugated. The saturation bound of Section 3.4 predicted that the optimum must alternate the sign of its curvature at the manufacturing limit, and the optimiser confirms the prediction. On the canonical stadium seed the converged section is fully saturated, with the whole boundary at $|\kappa| = 1/R_{\min}$, and measures 215.65 mm against a continuous lower bound of 215.20 mm, an excess of only 0.2% (Section 4.5.1). The excess that remains is the geometric overhead of the closure and total turning requirements, which force part of the arc length to turn the section around rather than accumulate force (Section 6.5). Convergence to this optimum is not an artefact of the starting shape: of eight independent seed families, six reach the same geometry and the same perimeter to within a fraction of a percent, while the two that settle about 2% higher confirm that local minima exist and that the optimiser escapes them from every other seed (Section 4.5.6).

The choice of boundary parameterisation is decisive. The second primary objective asked for a parameterisation capable of representing the true optimum, and the comparison of Section 4.5.2 shows why this mattered. The inherited polar description, even after its force model was raised to the fidelity of the reference tool and its runs made robust (Section 3.7), cannot fold: on the stadium at a matched target it stops at 259.79 mm, still 29% above the continuous bound, because the corrugated optimum lies outside the star-shaped family it can express (Section 3.7.3). Parameterising the signed curvature directly, with the manufacturing bound enforced by construction through the saturating map of Equation (4.1), reaches 207.11 mm from the same seed, within 3.1% of the bound and 20.3% shorter than the best the radial family can produce. The limitation of the inherited optimiser was structural, not a matter of tuning, and removing it is the core methodological contribution of this thesis.

The multi-section design space is multi-basin, and basin family is decided at initialisation. Extending the method to the full structure exposed a difficulty that no single section study could show: a closed curve saturated at the curvature limit can distribute its turning across different lobe counts, and each count is a separate basin of attraction that a section commits to early in a run, while the constraint penalties still dominate the loss (Section 5.2). Independently optimised sections therefore scatter across families, and the resulting stack, although optimal section by section, lofts into a surface with inter-cross-section kinks of up to 29.7° (Section 4.6). Penalties that couple neighbouring sections can smooth within a family but cannot move a section between families; what succeeds is the structural mechanism of single-family warm-starting, which initialises every section inside one chosen family by blending a donor section in curvature space, so that the family is fixed before the optimiser takes its first step (Section 5.3).

The complete redesigned structure is both lighter and continuous. With the warm-start in place, the delivered assembly meets every one of its thirty-five cross-section force targets, and sits 0.98% above the summed theoretical perimeter floor (Sections 6.1 and 6.5). Its maximum inter-cross-section kink falls from the 29.7° of the independent optima to 3.29°

at the tip section, with all other sections below 3° and a mean of 0.90°, close to the 2.43° benchmark set by the original production geometry. Continuity costs essentially nothing: the final assembly is in fact slightly shorter in total perimeter than the stack of independent optima (Table 6.1), so the manufacturability was recovered without giving back the mass saving that motivated the work.

Closing the loop through the reference tool verified the result and improved the tool itself. The supplementary objectives are also met. The optimised curvature description is exported automatically to an Abaqus shell model that meshes the true arcs of the stored curvature (Section 5.5), and CrushAnalytica, re-reading that model with no knowledge of how it was generated, returns the designed forces to within 1.4% in total, and −4.1% in the worst case section (Section 6.2). Reaching this agreement exposed and quantified a genuine weakness of spline based curvature reconstruction: on geometries that deliberately saturate the curvature bound, the smoothing spline rounds the saturated regions and reports radii of 17–30 mm where the true radius is 15 mm, understating the force by roughly 10% until the mesh is refined to millimetre order node spacing and the spline tolerance tightened.

One caveat qualifies these conclusions, stated in Section 6.5 and repeated here deliberately. The verification is independent in implementation but circular in physics: both the optimiser and CrushAnalytica encode the same quasi-static, empirical crush model, which assumes stable progressive crushing throughout. Until the generated model is crushed, in simulation or in the laboratory, the reported savings are a prediction of the analytical model, made twice, rather than a demonstrated property of the structure. That demonstration is the first item of the section that follows.

7.2 Future work

The framework developed in this thesis is complete from finite element model to verified finite element model. However, several directions could extend its use and make its conclusions stronger. These directions are listed below.

- **Numerical and experimental validation.** The generated finite element model should be tested on CZone/Abaqus crush simulation. This would test the corrugated geometry against physics the analytical model does not contain, and it is the definitive check of the predicted force. Beyond simulation, a manufactured demonstrator, crushed in the laboratory alongside a baseline section, would close the remaining gap between prediction and demonstration.
- **Stability and buckling.** The analytical model assumes stable progressive crushing everywhere. The optimiser also does not check global or local buckling of the corrugated thin walls. A buckling assessment of the optimised sections is therefore needed. At first, this should be used as a post check on the final geometry. If buckling becomes a limiting factor, it should then be added as a differentiable constraint inside the loss, so

that stability is designed into the section rather than checked only afterward.

- **Extending the formulation for branched sections.** The present formulation describes a single closed loop, so the internal rib of the branched SIS partitions had to be stripped before fitting (Section 4.6). Extending the curvature description to branched topologies would let ribbed sections be optimised rather than simplified.
- **Oblique crush validation.** Further testing is needed to validate the oblique crush performance of CrushAnalytica. Once this capability has been properly verified, a dedicated study should be carried out to assess the influence of the rib during oblique crushing. A further study should then validate the oblique crush performance of the optimised section presented in this work.
- **Integration.** The optimiser currently runs from the command line. Packaging it as a module of the CrushAnalytica interface, so that a partitioned structure can be optimised, re-verified and exported from within the tool, would complete its transition from research code to the industrial design tool this work set out to build. A further improvement would be to rebuild the workflow around Python, since Python offers greater flexibility, while calling the MATLAB engine from within Python for the CrushAnalytica calculations. This would allow the full toolchain to be controlled from a single environment.

Taken together, the results of this thesis show that the step from fast crush prediction to automatic crush design is not only possible but cheap: once the physics is differentiable and the geometry is described by the quantity the physics actually depends on, its curvature, a complete crash structure can be made a quarter lighter in wall material in less time than a single simulation of the original.

References

[1] Derek Hull. “A unified approach to progressive crushing of fibre-reinforced composite tubes”. In: *Composites Science and Technology* 40.4 (1991), pp. 377–421. DOI: 10.1016/0266-3538(91)90031-J.

[2] A. G. Mamalis, M. Robinson, D. E. Manolakos, G. A. Demosthenous, M. B. Ioannidis, and J. Carruthers. “Crashworthy capability of composite material structures”. In: *Composite Structures* 37.2 (1997), pp. 109–134. DOI: 10.1016/S0263-8223(97)80005-0.

[3] Denis Dalli, Luís F. Varandas, Giuseppe Catalanotti, Stephen Foster, and Brian G. Falzon. “Assessing the current modelling approach for predicting the crashworthiness of Formula One composite structures”. In: *Composites Part B: Engineering* 201 (2020), p. 108242. DOI: 10.1016/j.compositesb.2020.108242.

[4] Fédération Internationale de l’Automobile (FIA). *2026 Formula 1 Technical Regulations*. Tech. rep. Issue 17, 28 April 2026. FIA, 2026. URL: https://www.fia.com/regulation/category/110.

[5] Pedro João Cunha. “Industry-focused development of an analytical tool for rapid performance assessment of composite crash structure designs”. MSc thesis. Faculdade de Engenharia, Universidade do Porto (FEUP), June 2024.

[6] Tiago Ferreira Cardoso. “Extension of an industry-focused analytical tool for rapid performance assessment of hybrid composite crash structure designs”. MSc thesis. Faculdade de Engenharia, Universidade do Porto (FEUP), Sept. 2025.

[7] Mohammad Jawaid, Mohamed Thariq, and Naheed Saba. *Failure Analysis in Biocomposites, Fibre-Reinforced Composites and Hybrid Composites*. 1st. Woodhead Publishing Series in Composites Science and Engineering. Cambridge, UK: Woodhead Publishing, 2019. ISBN: 978-0-08-102293-1. DOI: 10.1016/C2017-0-02928-9.

[8] Jianguang Fang, Guangyong Sun, Na Qiu, Nam H. Kim, and Qing Li. “On design optimization for structural crashworthiness and its state of the art”. In: *Structural and Multidisciplinary Optimization* 55.3 (2017), pp. 1091–1119. DOI: 10.1007/s00158-016-1579-y.

[9] Libo Yan and Nawawi Chouw. “Crashworthiness characteristics of flax fibre reinforced epoxy tubes for energy absorption application”. In: *Materials & Design* 51 (2013), pp. 629–640. DOI: 10.1016/j.matdes.2013.04.014.

[10] Sang-Heon Kim, Hyoungjoon Kim, and Nack J. Kim. “Brittle intermetallic compound makes ultrastrong low-density steel with large ductility”. In: *Nature* 518.7537 (Feb. 2015), pp. 77–79. DOI: 10.1038/nature14144.

[11] Yun Zhou, Fengchong Lan, and Jiqing Chen. “Crashworthiness research on S-shaped front rails made of steel-aluminium hybrid materials”. In: *Thin-Walled Structures* 49.2 (2011), pp. 291–297. DOI: 10.1016/j.tws.2010.10.003.

[12] Guohua Zhu, Guangyong Sun, Hao Yu, Shujuan Li, and Qing Li. “Energy absorption of metal, composite and metal/composite hybrid structures under oblique crushing loading”. In: *International Journal of Mechanical Sciences* 135 (2018), pp. 458–483. DOI: 10.1016/j.ijmecsci.2017.11.017.

[13] Chiara Bisagni, Giorgio Di Pietro, Luca Fraschini, and Davide Terletti. “Progressive crushing of fiber-reinforced composite structural components of a Formula One racing car”. In: *Composite Structures* 68.4 (2005), pp. 491–503. DOI: 10.1016/j.compstruct.2004.04.015.

[14] Gary Savage. “Development of penetration resistance in the survival cell of a Formula 1 racing car”. In: *Engineering Failure Analysis* 17.1 (2010), pp. 116–127. DOI: 10.1016/j.engfailanal.2009.04.002.

[15] U.S. Department of Defense. *MIL-STD-1290A (AV): Light Fixed and Rotary-Wing Aircraft Crash Resistance*. Military Standard MIL-STD-1290A(AV). Washington, DC: Department of Defense, Sept. 1988.

[16] Karen E. Jackson, Edwin L. Fasanella, and Karen H. Lyle. *Crash Certification by Analysis: Are We There Yet?* Tech. rep. NASA, 2006.

[17] Federal Aviation Administration. *Airworthiness Standards: Normal Category Airplanes (14 CFR Part 23)*. Code of Federal Regulations, Title 14, Part 23. 2023. URL: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23.

[18] Federal Aviation Administration. *Airworthiness Standards: Transport Category Airplanes (14 CFR Part 25)*. Code of Federal Regulations, Title 14, Part 25. 2023. URL: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25.

[19] Federal Aviation Administration. *Airworthiness Standards: Normal Category Rotorcraft (14 CFR Part 27)*. Code of Federal Regulations, Title 14, Part 27. 2023. URL: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-27.

[20] European Union Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Normal, Utility, Aerobatic, and Commuter Aeroplanes (CS-23)*. Amendment 6, applicable as of 07 March 2023. 2023.

[21] European Union Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes (CS-25)*. Amendment 28, applicable as of 19 December 2023. 2023.

[22] European Union Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Small Rotorcraft (CS-27)*. Amendment 10, applicable as of 07 February 2023. 2023.

[23] A. B. M. Supian, S. M. Sapuan, M. Y. M. Zuhri, E. S. Zainudin, and H. H. Ya. "Hybrid reinforced thermoset polymer composite in energy absorption tube application: A review". In: *Defence Technology* 14.4 (2018), pp. 291–305. DOI: 10.1016/j.dt.2018.04.004.

[24] George C. Jacob, John F. Fellers, Srdan Simunovic, and J. Michael Starbuck. "Energy absorption in polymer composites for automotive crashworthiness". In: *Journal of Composite Materials* 36.7 (2002), pp. 813–850. DOI: 10.1177/0021998302036007164.

[25] Gary L. Farley and Robert M. Jones. "Crushing characteristics of continuous fiber-reinforced composite tubes". In: *Journal of Composite Materials* 26.1 (1992), pp. 37–50. DOI: 10.1177/002199839202600103.

[26] Jing Xu, Yan Ma, Qianjin Zhang, Toshi Sugahara, Yuqiu Yang, and Hiroyuki Hamada. "Crashworthiness of carbon fiber hybrid composite tubes molded by filament winding". In: *Composite Structures* 139 (2016), pp. 130–140. DOI: 10.1016/j.compstruct.2015.11.053.

[27] Tomasz Wierzbicki and Wlodzimierz Abramowicz. "On the crushing mechanics of thin-walled structures". In: *Journal of Applied Mechanics* 50.4a (1983), pp. 727–734. DOI: 10.1115/1.3167137.

[28] S. Cauchi Savona and P. J. Hogg. "Effect of fracture toughness properties on the crushing of flat composite plates". In: *Composites Science and Technology* 66.13 (Oct. 2006), pp. 2317–2328. DOI: 10.1016/j.compscitech.2005.11.038.

[29] Rafea Dakhil Hussein, Dong Ruan, and Guoxing Lu. "An analytical model of square CFRP tubes subjected to axial compression". In: *Composites Science and Technology* 168 (2018), pp. 170–178. DOI: 10.1016/j.compscitech.2018.09.019.

[30] S. Boria, S. Pettinari, and F. Giannoni. "Theoretical analysis on the collapse mechanisms of thin-walled composite tubes". In: *Composite Structures* 103 (2013), pp. 43–49. DOI: 10.1016/j.compstruct.2013.03.020.

[31] A. G. Mamalis, D. E. Manolakos, G. A. Demosthenous, and M. B. Ioannidis. "Analytical modelling of the static and dynamic axial collapse of thin-walled fibreglass composite conical shells". In: *International Journal of Impact Engineering* 19.5–6 (1997), pp. 477–492. DOI: 10.1016/S0734-743X(97)00007-9.

[32] Paolo Feraboli. "Development of a corrugated test specimen for composite materials energy absorption". In: *Journal of Composite Materials* 42.3 (2008), pp. 229–256. DOI: 10.1177/0021998307086202.

[33] Paolo Feraboli, Bonnie Wade, Francesco Deleo, and Mostafa Rassaian. "Crush energy absorption of composite channel section specimens". In: *Composites Part A: Applied Science and Manufacturing* 40.8 (2009), pp. 1248–1256. DOI: 10.1016/j.compositesa.2009.05.021.

[34] Vasco Milton Lobão. "Development and testing of hybrid composite coupons for crush applications". MSc thesis. Faculdade de Engenharia, Universidade do Porto (FEUP), July 2022.

[35] Diogo Filipe Martins Moreira. "Development of hybrid composite coupons and experimental set-ups for dynamic crash testing". MSc thesis. Faculdade de Engenharia, Universidade do Porto (FEUP), Feb. 2024.

[36] A. G. Mamalis, D. E. Manolakos, M. B. Ioannidis, and D. P. Papapostolou. "On the response of thin-walled CFRP composite tubular components subjected to static and dynamic axial compressive loading: Experimental". In: *Composite Structures* 69.4 (2005), pp. 407–420. DOI: 10.1016/j.compstruct.2004.07.021.

[37] Xing Liu et al. "Experimental investigation of energy absorption behaviour of circular carbon/epoxy composite tubes under quasi-static and dynamic crush loading". In: *Composite Structures* 227 (2019), p. 111266. DOI: 10.1016/j.compstruct.2019.111266.

[38] J. André Lavoie and Sotiris Kellas. "Dynamic crush tests of energy-absorbing laminated composite plates". In: *Composites Part A: Applied Science and Manufacturing* 27.6 (1996), pp. 467–475. DOI: 10.1016/1359-835X(95)00058-A.

[39] Louis N. S. Chiu et al. "Crush responses of composite cylinder under quasi-static and dynamic loading". In: *Composite Structures* 131 (2015), pp. 90–98. DOI: 10.1016/j.compstruct.2015.04.057.

[40] Qiang Liu, Kai Liufu, Xiao Cui, Jie Li, Jianguang Fang, and Qing Li. "Multiobjective optimization of perforated square CFRP tubes for crashworthiness". In: *Thin-Walled Structures* 149 (2020), p. 106628. DOI: 10.1016/j.tws.2020.106628.

[41] Maria Pia Falaschetti, Federico Rondina, Nicola Zavatta, Enrico Troiani, and Lorenzo Donati. "Effective implementation of numerical models for the crashworthiness of composite laminates". In: *Engineering Failure Analysis* 160 (2024), p. 108196. DOI: 10.1016/j.engfailanal.2024.108196.

[42] Maria Rosaria Ricciardi, Ilaria Papa, Valentina Lopresto, Antonio Calzolari, Luca Vigna, and Vincenza Antonucci. "Experimental characterization of the crashworthiness of carbon fiber reinforced epoxy composites". In: *Progress in Aerospace Sciences* 148 (2024), p. 101003. DOI: 10.1016/j.paerosci.2024.101003.

[43] Bo Wang et al. "Optimal selection of composite layup considering the fuselage crashworthiness". In: *Composites Part C: Open Access* 16 (2025), p. 100574. DOI: 10.1016/j.jcomc.2025.100574.

[44] Xiaolin Deng, Yingchun Liu, Weigang Zhang, and Chunyan Feng. “Crashworthiness design with a new optimization criterion for multilevel thin-walled structures”. In: *Coatings* 15.9 (2025), p. 1058. DOI: 10.3390/coatings15091058.

[45] Peng Lian, Wenjie Wang, Yong Zhang, Xiaobing Chen, and Shucai Xu. “Multi-objective optimization of lightweight and crashworthiness of automotive front-end structures based on stacked regression surrogate”. In: *Engineering Applications of Artificial Intelligence* 155 (2025), p. 111138. DOI: 10.1016/j.engappai.2025.111138.

[46] Martin Philip Bendsøe and Noboru Kikuchi. “Generating optimal topologies in structural design using a homogenization method”. In: *Computer Methods in Applied Mechanics and Engineering* 71.2 (1988), pp. 197–224. DOI: 10.1016/0045-7825(88)90086-2.

[47] Fabian Duddeck, Stephan Hunkeler, Pablo Lozano, Erich Wehrle, and Duo Zeng. “Topology optimization for crashworthiness of thin-walled structures under axial impact using hybrid cellular automata”. In: *Structural and Multidisciplinary Optimization* 54.3 (2016), pp. 415–428. DOI: 10.1007/s00158-016-1445-y.

[48] Christian J. G. Silva, Rogério F. F. Lopes, Tiago M. R. M. Domingues, M. P. L. Parente, and Pedro M. G. P. Moreira. “Crashworthiness topology optimisation of a crash box to improve passive safety during a frontal impact”. In: *Structural and Multidisciplinary Optimization* 68.1 (2025), p. 3. DOI: 10.1007/s00158-024-03924-6.

[49] Peng Zhao, Yang Zhang, Guanghui Shi, Guozhong Dong, and Jiao Jia. “Crashworthiness topology optimization of shell–infill structures”. In: *Composite Structures* 377 (2025), p. 119863. DOI: 10.1016/j.compstruct.2025.119863.

[50] Minsung Lee, Hyunjun Kim, and Jaewook Lee. “Multiscale topology optimization of functionally graded lattice structure for energy absorption”. In: *Multiscale Science and Engineering* 7.2 (2025), pp. 135–144. DOI: 10.1007/s42493-025-00135-z.

[51] Xiong Zhang, Yuandong Guo, Xuan Cheng, Zhiyuan Dou, and Sebastian Stichel. “Integrated topology optimization method for crashworthiness of metal–FRP hybrid thin-walled tubes: A review and analysis”. In: *Chinese Journal of Mechanical Engineering* 39 (2025), p. 100046. DOI: 10.1016/j.cjme.2025.100046.

[52] Qihua Ma et al. “Crashworthiness optimization of composite/metal hybrid tubes with triggering holes”. In: *Designs* 10.2 (2025), p. 44. DOI: 10.3390/designs10020044.

[53] Dassault Systèmes Simulia Corp. *CZone for Abaqus: composite crush simulation*. Abaqus documentation; composite crush add-on based on CZone technology from Engenuity Ltd. Dassault Systèmes SIMULIA. Providence, RI, USA, 2020.

[54] Raphael T. Haftka and Zafer Gürdal. *Elements of Structural Optimization*. 3rd edition. Solid Mechanics and Its Applications. Dordrecht: Kluwer Academic Publishers, 1992. DOI: 10.1007/978-94-011-2550-5.

[55] Jamshid A. Samareh. “Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization”. In: *AIAA Journal* 39.5 (2001), pp. 877–884. DOI: 10.2514/2.1391.

[56] Charles T. Zahn and Ralph Z. Roskies. “Fourier descriptors for plane closed curves”. In: *IEEE Transactions on Computers* C-21.3 (1972), pp. 269–281. DOI: 10.1109/TC.1972.5008949.

[57] J. Alan Adams. “The intrinsic method for curve definition”. In: *Computer-Aided Design* 7.4 (1975), pp. 243–249. DOI: 10.1016/0010-4485(75)90069-X.

[58] Kenjiro T. Miura. “A general equation of aesthetic curves and its self-affinity”. In: *Computer-Aided Design and Applications* 3.1–4 (2006), pp. 457–464. DOI: 10.1080/16864360.2006.10738484.

[59] Raph Levien and Carlo H. Séquin. “Interpolating splines: Which is the fairest of them all?” In: *Computer-Aided Design and Applications* 6.1 (2009), pp. 91–102. DOI: 10.3722/cadaps.2009.91-102.

[60] James McCrae and Karan Singh. “Sketching piecewise clothoid curves”. In: *Computers & Graphics* 33.4 (2009), pp. 452–461. DOI: 10.1016/j.cag.2009.05.006.

[61] Diederik P. Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations (ICLR)*. arXiv:1412.6980. 2015.

[62] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic gradient descent with warm restarts”. In: *5th International Conference on Learning Representations (ICLR)*. arXiv:1608.03983. 2017.

[63] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006. DOI: 10.1007/978-0-387-40065-5.

[64] David J. Wales and Jonathan P. K. Doye. “Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms”. In: *The Journal of Physical Chemistry A* 101.28 (1997), pp. 5111–5116. DOI: 10.1021/jp970984n.

[65] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. “Automatic differentiation in machine learning: a survey”. In: *Journal of Machine Learning Research* 18.153 (2018), pp. 1–43.

[66] Adam Paszke et al. “PyTorch: An imperative style, high-performance deep learning library”. In: *Advances in Neural Information Processing Systems 32 (NeurIPS)*. 2019, pp. 8024–8035.

[67] Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. “Neural reparameterization improves structural optimization”. In: *arXiv preprint* (2019). arXiv:1909.04240; NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks.

[68] Tianju Xue et al. “JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science”. In: *Computer Physics Communications* 291 (2023), p. 108802. DOI: 10.1016/j.cpc.2023.108802.

[69] Nils Meyer. *torch-fem: GPU-accelerated differentiable finite elements for solid mechanics with PyTorch*. Version 0.7.0. 2026. DOI: 10.5281/zenodo.20306384. URL: https://github.com/meyer-nils/torch-fem.

[70] Tomas Möller. "A fast triangle-triangle intersection test". In: *Journal of Graphics Tools* 2.2 (1997), pp. 25–30. DOI: 10.1080/10867651.1997.10487472.

[71] Philippe Boisse, Julien Colmars, Nahiene Hamila, Naim Naouar, and Quentin Steer. "Bending and wrinkling of composite fiber preforms and prepregs. A review and new developments in the draping simulations". In: *Composites Part B: Engineering* 141 (2018), pp. 234–249. DOI: 10.1016/j.compositesb.2017.12.061.

[72] Carl de Boor. *A Practical Guide to Splines*. Revised. Vol. 27. Applied Mathematical Sciences. New York: Springer, 2001.

[73] Sam. *McLaren MP4-30*. 2015. URL: https://www.racecar-engineering.com/cars/mclaren-mp4-30/ (visited on 08/02/2025).

[74] Pedro R. S. Antunes and Pedro Freitas. "Numerical optimization of low eigenvalues of the Dirichlet and Neumann Laplacians". In: *Journal of Optimization Theory and Applications* 154.1 (2012), pp. 235–257. DOI: 10.1007/s10957-011-9983-3.

[75] Christian H. Reinsch. "Smoothing by spline functions". In: *Numerische Mathematik* 10.3 (1967), pp. 177–183. DOI: 10.1007/BF02162161.

Appendix A

Optimiser Command-Line Reference

This appendix documents the command-line interface of the curvature optimiser whose mechanisms are developed in Chapters 4 and 5.

A.1 Command-line options

Listing A.1 reproduces the principal options of `rocOptimizer_v8.py` as printed by `-help`. It covers the physical constants of Chapter 3 (`-min-roc`, `-sigma-const`, `-flat-transition`), the penalty weights, and the main mechanism switches of Chapters 4 and 5, each with its default value; a plain invocation with no arguments therefore reproduces the baseline joint configuration. Housekeeping options (input/output paths, run tagging, thread count) and the fine-tuning sub-options of the individual mechanisms are omitted; the complete list is available through `-help`.

```
options:
  --layers LAYERS       1-based layer indices, e.g. '2-36' or '5,10,18'. Empty
                        = all non-empty layers (2..N).
  --N N                 Segments per layer (default 64; must be %4==0 for D2
                        symmetry).
  --steps STEPS
  --lr LR
  --patience PATIENCE
  --tol TOL
  --min-roc MIN_ROC     Minimum radius of curvature [mm]. Default 15.0.
  --sigma-const SIGMA_CONST
                        Stress law constant sigma_const [MPa]. Default 90.0.
  --flat-transition FLAT_TRANSITION
                        Upper RoC clamp in stress law [mm]. Default
                        1221.947188.
  --symmetry {D2,none}
  --lambda-force-attract LAMBDA_FORCE_ATTRACT
                        Attractor coefficient pulling F toward (F_target +
```

```
                        force-margin) after feasibility (F>=F_t) is met.
                        Default 0.001. Set 0 to recover the v4-original
                        feasibility-only behaviour.
  --force-margin FORCE_MARGIN
                        Newtons above F_target where the attractor is
                        centered, so the equilibrium F sits slightly above F_t
                        (safety margin). Default 100 N. Set 0 to attract
                        exactly to F_t.
  --bbox, --no-bbox     Constrain each layer to its seed bounding box (soft
                        v3-style penalty). Default: enabled. Pass --no-bbox to
                        disable.
  --bbox-scale BBOX_SCALE
                        Multiplier on each layer's seed (W, H) before the bbox
                        penalty kicks in. <1 tightens, >1 loosens. Default 1.0
                        (exact seed bbox).
  --lambda-height LAMBDA_HEIGHT
                        Soft penalty weight on the centerline-height shortfall
                        (height_at_x0 < HEIGHT_MIN_MM). Default 1000.0. 0
                        disables.
  --height-min HEIGHT_MIN
                        Minimum acceptable height at x=0 (mm). The boundary's
                        top and bottom crossings of the y-axis must be at
                        least this far apart. Default 4.063 mm (= L3
                        centerline height in the reference v4 run).
  --lambda-couple LAMBDA_COUPLE
                        Initial weight of the inter-layer continuity term:
                        lambda * sum over coupled pairs of mean(((kappa_i -
                        kappa_j) * min_roc)^2). Default 1000. Set 0 to disable
                        (fully independent layers, v5 batched behaviour).
  --lambda-smooth LAMBDA_SMOOTH
                        Weight of the v7 loft-smoothness term: lambda * sum
                        over interior layers of mean_i |P_{l+1} - 2 P_l +
                        P_{l-1}|^2 on placed node positions [mm^2]. Constant
                        (NOT annealed); zero for a linear taper. Default 1.
                        Set 0 to disable (recovers v6 loss, but v7 joint best-
                        state tracking and natural-frame outputs remain).
  --chain-init {fe,broadcast}
                        Adam initialisation: 'fe' = each layer starts from its
                        own FE-extracted seed; 'broadcast' = every layer in a
                        constant-thickness chain starts from the chain-start
                        layer's seed (kappa, L, theta_0), so all chain members
                        begin in the same basin -- the joint analogue of the
                        v5 sequential warm-start. Default broadcast (v7; v6
                        was fe).
  --repair, --no-repair
                        After the joint run, scan for single-layer basin
                        defects (kappa sign-flip count above both same-chain
                        neighbours, h_x0 outlier, or infeasible next to a
                        feasible neighbour) and re-optimise flagged layers
                        warm-started from the nearest healthy neighbour's
                        optimised shape. A repair is kept only if feasible and
                        strictly better. Default on; --no-repair disables.
  --warm-start-mat WARM_START_MAT
                        v8: previous result .mat to warm-start from (default:
                        none = v7 behaviour).
  --warm-mode {family,asis}
                        v8 warm start: 'family' re-bases every layer on
                        --donor's final shape, anisotropically rescaled to
                        that layer's converged extents; 'asis' restores each
```

```
                        layer's own final state from the warm mat.
  --donor DONOR         v8 family mode: 1-based BigArray layer index whose
                        optimised shape seeds every layer. Default 13.
  --extent-corridor EXTENT_CORRIDOR
                        v8 family mode: half-width [mm] of a two-sided soft
                        window pinning each layer's (W, H) extents to the
                        smoothed warm-start schedule, so the optimiser cannot
                        re-create extent steps between layers. 0 = off
                        (default).
  --shape-tie SHAPE_TIE
                        v8: comma-separated 1-based BigArray layer RANGES,
                        e.g. '12-17,33-36'. Within each range, interior layers
                        are pulled toward the kappa-space linear interpolation
                        of the range endpoints, forcing the cross-section to
                        morph gradually across a basin step (extents alone
                        cannot pin the internal lobe pattern). Empty = off.
```

Code Listing A.1: Principal options of `rocOptimizer_v8.py` (`-help` output).

A.2 Reproduction commands

Listings A.2–A.5 collect the commands behind the definitive results of Chapter 6, reconstructed from the run logs archived next to each result file. Options left at their defaults are not repeated.

```
python rocOptimizer_v8.py --N 128 --steps 100000 \
    --warm-start-mat <v7 donor run>.mat \
    --warm-mode family --donor 13 --warm-smooth-window 5 \
    --couple-floor 25 --lambda-smooth 2 --no-repair \
    --extent-corridor 1 --corridor-layers "12-17,33-36" \
    --shape-tie "12-17,33-36" --lambda-tie 5000 \
    --tip-extrapolate 3 --height-min 5 --torch-threads 8
```

Code Listing A.2: Final full-structure run (joint + family warm start).

```
python rocOptimizer_v5.py --N 64 --steps 100000 \
    --seed-chain --bridge-singleton-chains --lambda-height 10000
```

Code Listing A.3: Uncoupled sequential run (Chapter 6 comparison baseline).

```
python ABAQUS-Model/build_inp_from_mat.py --mat <final>.mat \
    --refine 4 --placement placed
```

Code Listing A.4: Finite element model generation from the optimised `.mat`.

```
abaqus cae noGUI=import_inp_to_cae.py -- inp=<model>.inp
```

Code Listing A.5: Scripted import into Abaqus/CAE.

Appendix B

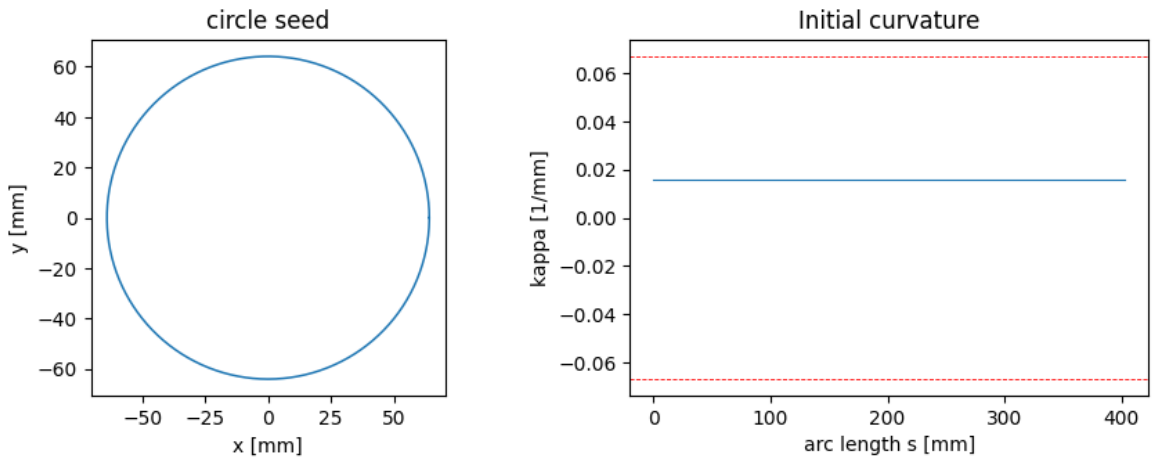
Seed Shape Library

B.1 The seed families

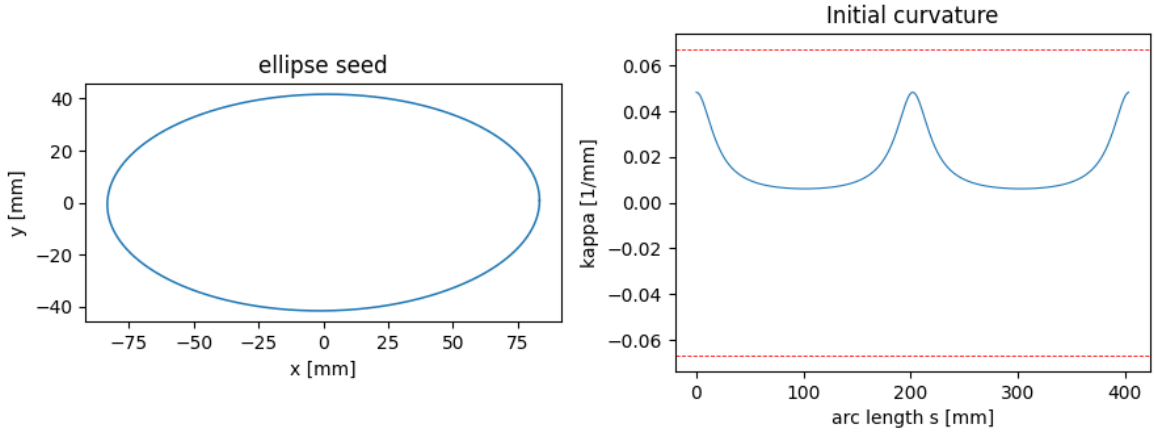
The optimiser is started from a library of eight seed families, generated by `seeds.py` and previewed with `preview_seeds.py`. Each is a closed curve resampled to a common node count (here $N = 800$ for the preview), shown in Figure B.1 together with its initial signed curvature profile $\kappa(s)$.

Every seed returns the three inputs of the optimiser directly: the per segment curvature κ, the perimeter L and the seam heading θ_0. Two constructions are used. The *geometric* families (circle, ellipse, stadium, rounded rectangle, square and blob) are built as an explicit closed curve whose curvature is read off analytically, so they close in the plane from the start. The *curvature domain* families (corrugated and Fourier) are written directly as a curvature profile $\kappa(s)$ whose total turning is 2π; these need not close in the plane at first, and the optimiser pulls them shut in the early iterations. In both cases a single scale parameter is solved by bisection so that the initial integrated crush force matches the target F_t of the run, and the curvature is held inside the manufacturing bound $|\kappa| \leq 1/R_{\min}$.

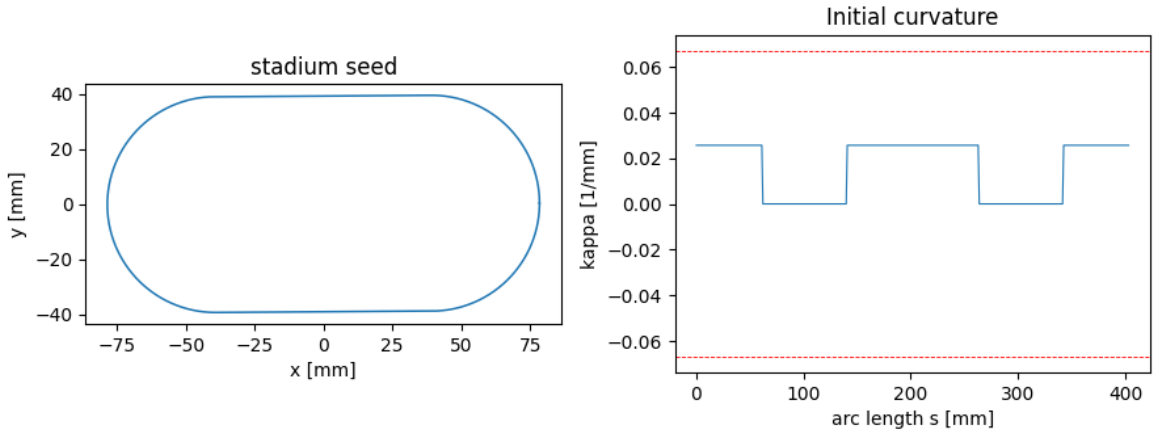
Most families expose a few parameters beyond the scale that are set automatically to match the force. The geometric families take an *aspect* ratio (ellipse, stadium and rounded rectangle), with a corner radius *corner_R* for the rounded rectangle. The families that carry curvature are the most adjustable: the blob, corrugated and Fourier seeds each take an *amplitude* that sets how strongly the curvature swings, and therefore how deep the concave stretches run, together with a feature count, the number of modes *blob_modes* for the blob, the lobe count *n_lobes* for the corrugated seed and the mode count *n_modes* for the Fourier seed. The two random families, the blob and the Fourier seed, additionally take an *rng_seed* that makes a given draw reproducible. The values used for the previews are noted in each panel caption.



(a) Circle

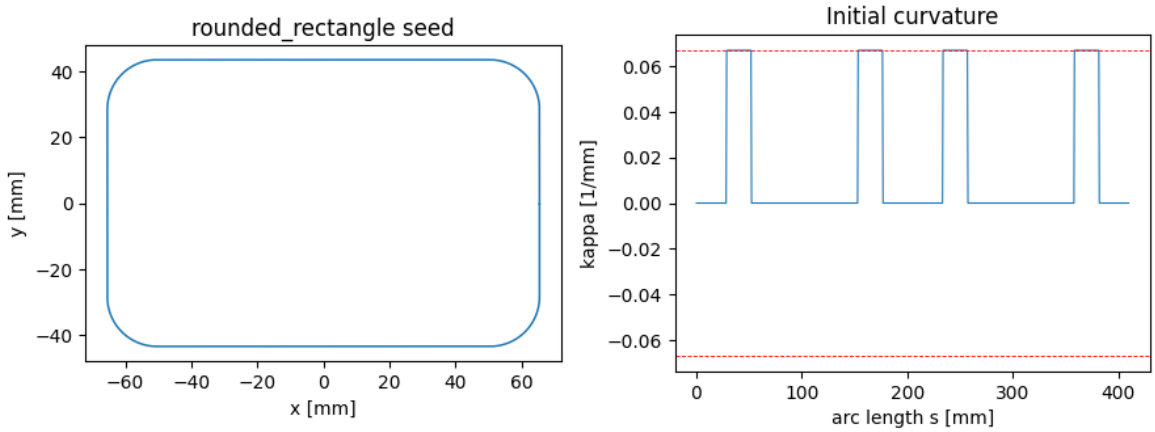


(b) Ellipse

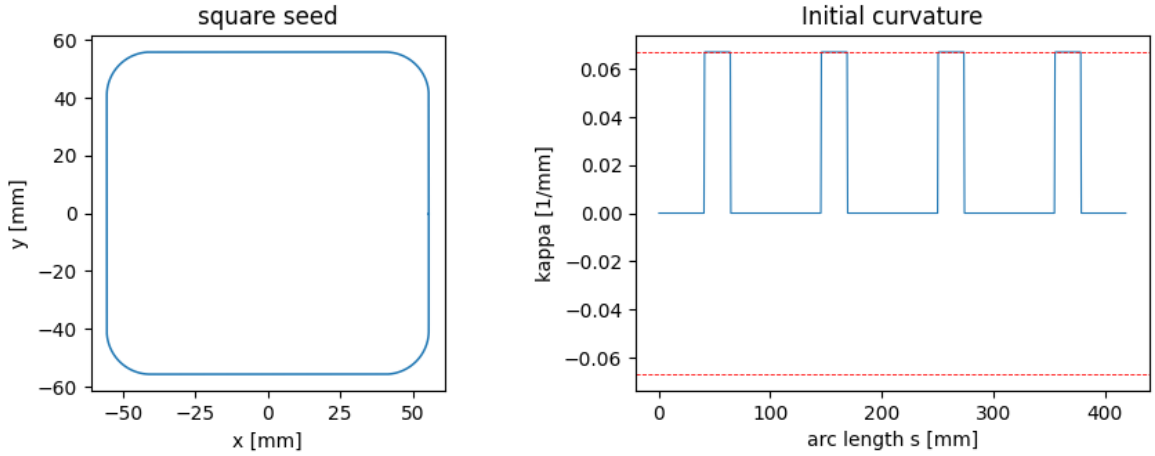


(c) Stadium

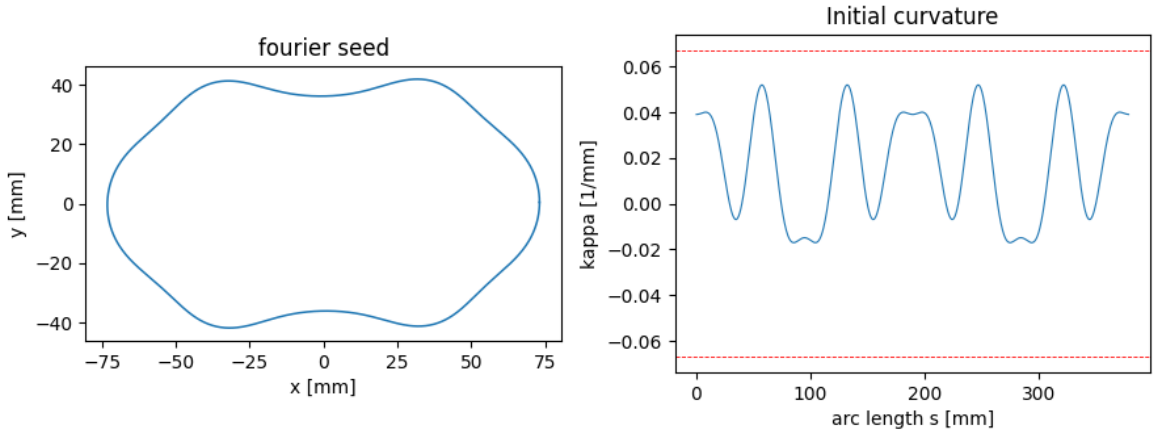
Figure B.1: Seed shape library, each family shown with its boundary (left) and its initial signed curvature profile $\kappa(s)$ (right). The dashed lines on the curvature panels mark the manufacturing bound $\pm 1/R_{\min}$.



(d) Rounded rectangle

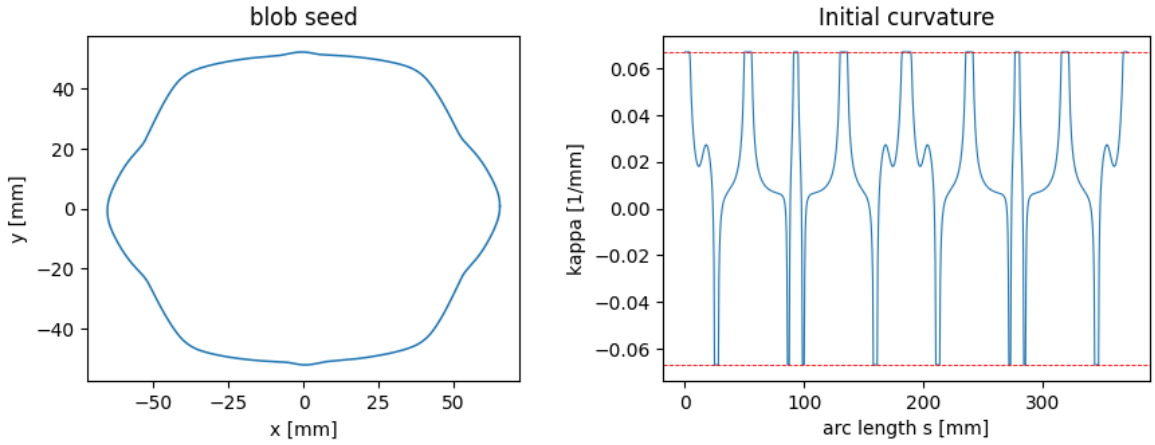


(e) Square

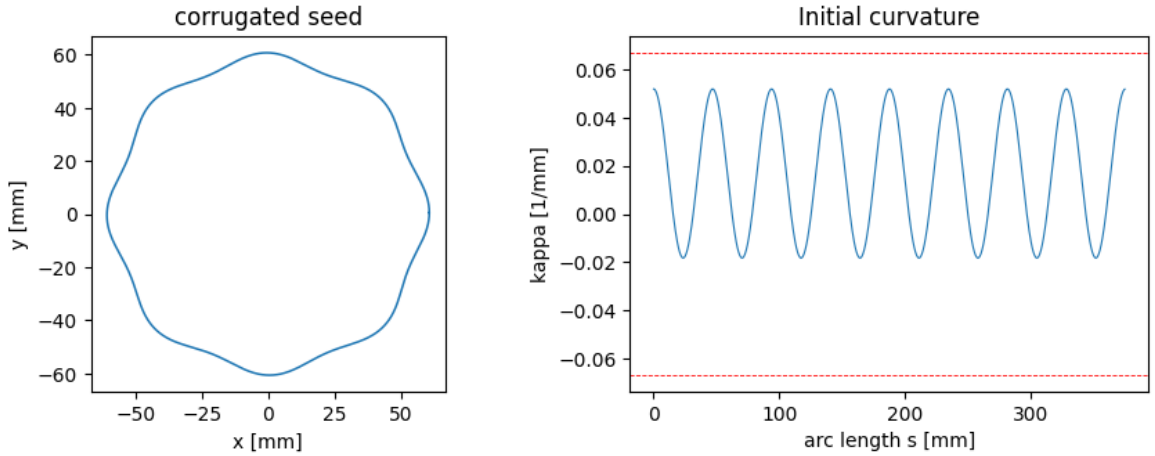


(f) Fourier ($m = 5$, amplitude 0.7)

Figure B.1: Seed shape library (continued).



(g) Blob ($m = 6$, amplitude 0.7)

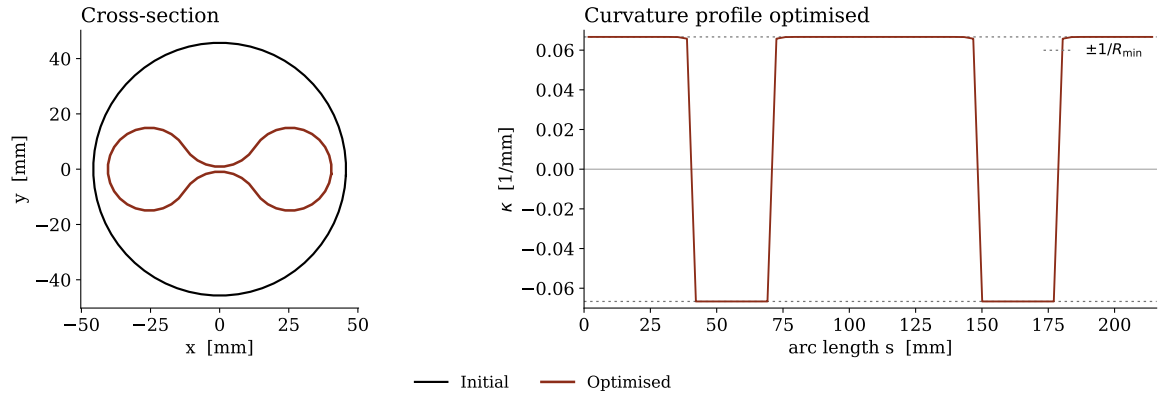


(h) Corrugated (8 lobes, amplitude 0.7)

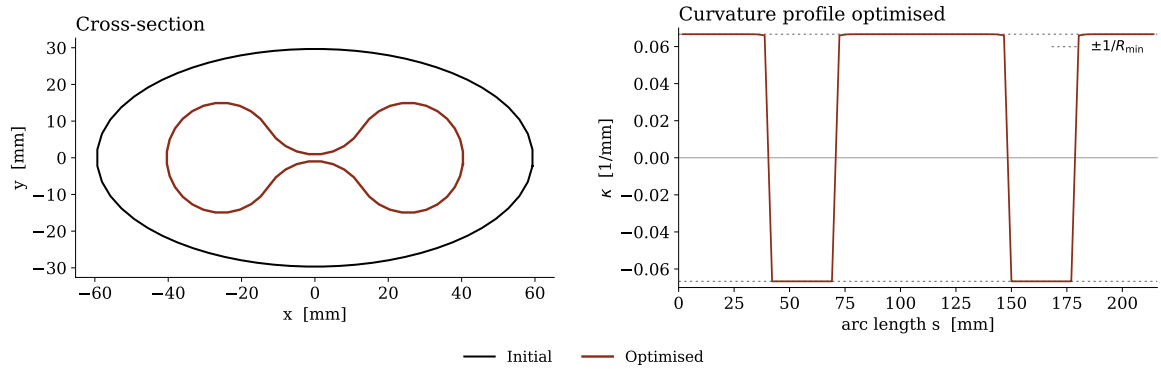
Figure B.1: Seed shape library (continued).

B.2 Per-seed convergence runs

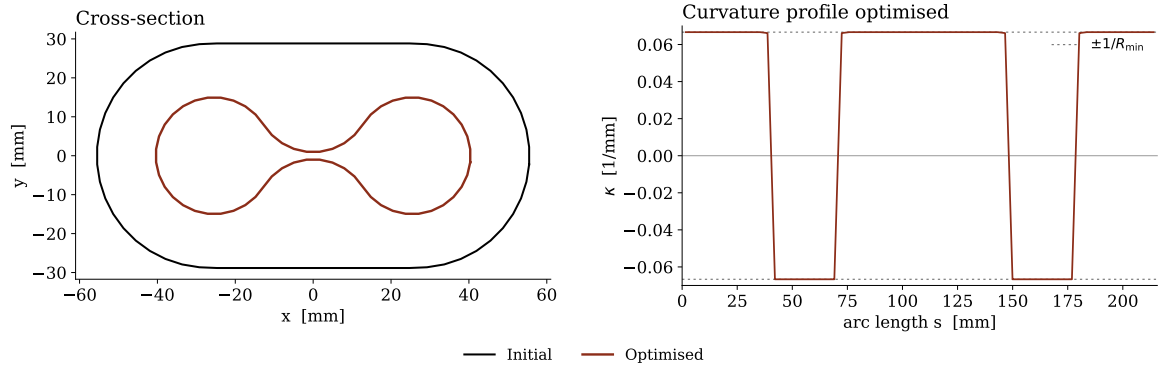
Figure B.2 collects the per-seed output of the seed convergence study ($N = 128$, $F_t = 75\,\mathrm{kN}$, 100 000 steps). Each panel shows the optimised cross-section of one seed against its initial outline, together with the optimised signed curvature profile $\kappa(s)$.



(a) Circle

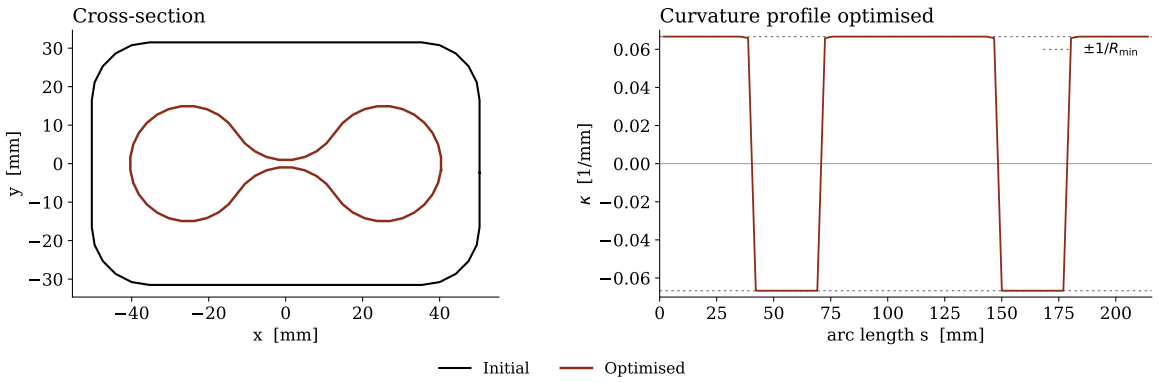


(b) Ellipse

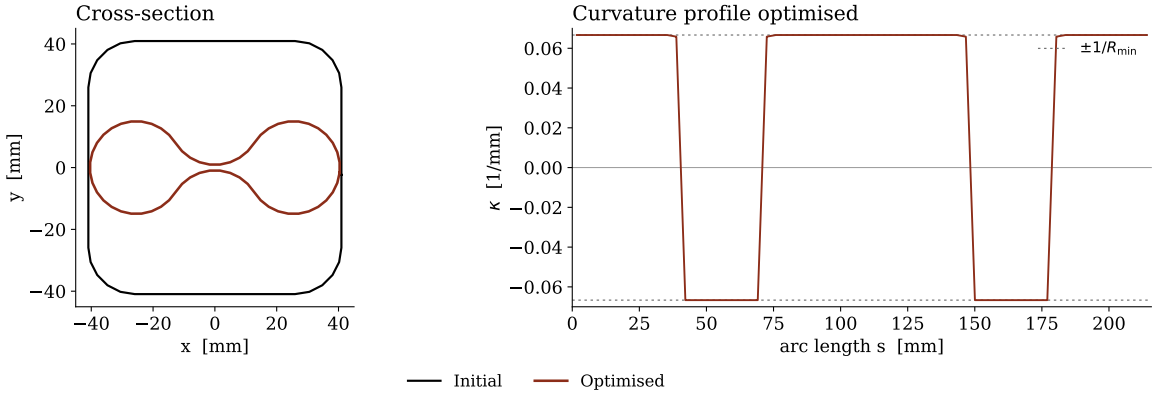


(c) Stadium

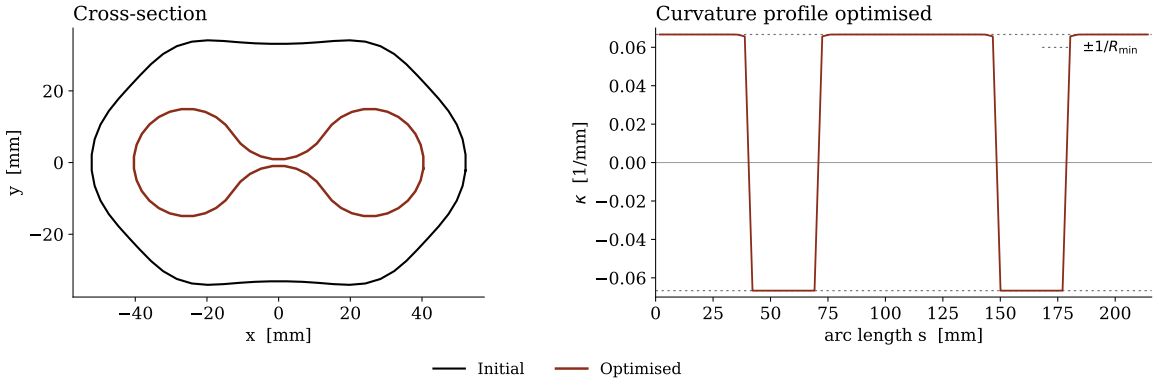
Figure B.2: Per-seed convergence runs: optimised cross-section against the initial outline (left) and optimised signed curvature profile $\kappa(s)$ (right). The dashed lines on the curvature panels mark the manufacturing bound $\pm 1/R_{\min}$.



(d) Rounded rectangle

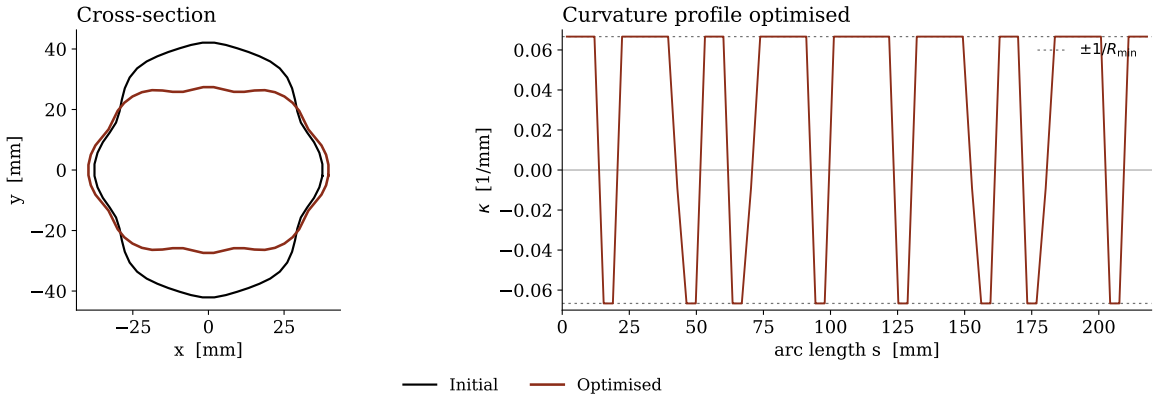


(e) Square

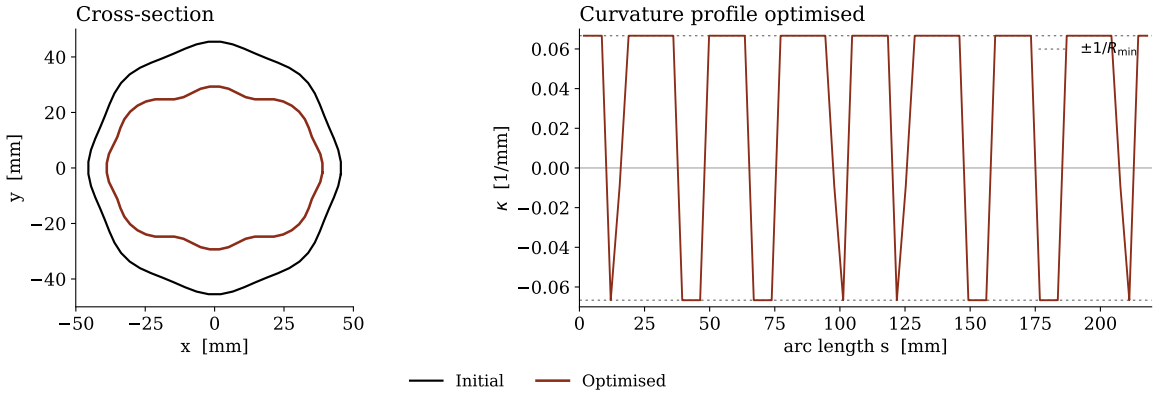


(f) Fourier

Figure B.2: Per-seed convergence runs (continued).



(g) Blob



(h) Corrugated

Figure B.2: Per-seed convergence runs (continued).

Appendix C

Data Interfaces of the Design Loop

The design loop of Section 5.5.1 exchanges its geometry through two MATLAB data files: the partition data that CrushAnalytica exports, and the optimised assembly that the optimiser writes back. This appendix documents only the fields that the next stage of the loop actually reads. Coordinates and lengths are stored in millimetres and forces in newtons.

C.1 Partition data: `BigArray.mat`

The seed extraction of Section 4.6 reads one variable from this file, the struct array `section`, one entry per partition plane from the inbound end to the outbound tip (Table C.1). The first entry is empty, because the first partition plane intersects no material, so the thirty-five real sections occupy entries 2 to 36; the thesis numbers them 1 to 35 from the tip.

Table C.1: Fields of `BigArray.mat` read by the seed extraction. n_k is the number of partition nodes of entry k.

Field	Shape	Meaning
`section`	1×36 struct	one entry per partition plane along the crush axis
`section(k).NodesCoordinates`	$n_k \times 3$	positions of the partition nodes
`section(k).ConnectivityList`	$n_k \times 2$	node pairs forming the closed polyline, including any rib branches
`section(k).Thickness`	$n_k \times 1$	wall thickness at each node
`section(k).ForceTotal`	scalar	crush force of the original section, which becomes the section force target

The file is a saved MATLAB workspace and carries auxiliary variables alongside `section`;

only the fields above form the interface to the optimiser.

C.2 Optimised assembly: the optimiser `.mat`

The optimiser writes its result as a flat `.mat` file. The finite element builder of Section 5.5.2 consumes the first group of fields in Table C.2; the second group is read when the results of Chapter 6 are reported.

Table C.2: Fields of the optimiser output `.mat`. N is the number of boundary segments per section (here $N = 128$).

Field	Shape	Meaning
Read by the finite element builder		
`kappa`	$35 \times N$	signed curvature of every segment of every section; the true-arc mesh is built from these
`nodesCoordsFinal`	$35 \times N \times 3$	optimised boundary nodes, each section centred
`nodesCoordsFinalPlaced`	$35 \times N \times 3$	the same nodes aligned to the reference face (Section 5.4)
`thickness`	1×35	per-section wall thickness
`zCoord`	1×35	station of each section along the crush axis
`feasibleFinal`	1×35	flag marking the sections that meet every constraint
Read when reporting results		
`sectionIndex`	1×35	index of the source entry in `section` (2 to 36)
`forceTarget`, `forceFinal`	1×35	force target and final force of each section
`perimeterInitial`, `perimeterFinal`	1×35	seed and optimised perimeter of each section
`minRoc`, `sigmaConst`, `flatTransition`	scalars	the constants $R_{\min}$, σ_0 and R_{flat} of Section 3.1.1
`Nsegments`	scalar	the discretisation N

The remaining fields echo the command-line options of Appendix A and store per-run convergence diagnostics, so every result file fully documents the run that produced it; they are not part of the interface and are not listed here.

Appendix D

Per-Section Result Tables

This appendix records the full results for each section behind the summary figures and the campaign table of Chapter 6 (Table 6.1). For each of the thirty-five sections of the delivered assembly, Table D.1 lists the section plane height z, the laminate thickness t, the crush force target F_t, the perimeter of the original section P_0, the optimised perimeter P, the final force as a percentage of its target, and the maximum inter-cross-section loft kink measured with the metric of Section 3.2. The original perimeter is that of the rib-stripped outer loop of the source partition section, resampled at the same $N = 128$ discretisation the optimiser uses, so the two perimeter columns compare like with like. Sections are numbered from the outbound end (section 1) to the inbound end (section 35), consistent with the tables of the main text. The kink is undefined at the two end sections of the stack.

The same run is shown graphically in Figures D.1 and D.2: first the optimised cross-section of every section against its extracted seed, then the corresponding signed curvature profiles. Each panel repeats the section plane height z, the thickness t and, for the cross-sections, the perimeter and force outcome of the section. Sections are numbered from the outbound tip, as in Table D.1.

Table D.1: Section results of the final optimised assembly.

Section	z [mm]	t [mm]	F_t [kN]	P_0 [mm]	P [mm]	F/F_t [%]	Kink [°]
1	207.6	2.4	31.5	110.0	105.8	112.3	—
2	197.6	2.4	38.5	116.8	111.5	100.4	3.29
3	187.6	2.4	40.0	123.6	117.1	100.6	2.90
4	177.6	2.4	41.7	130.3	120.8	100.6	1.10
5	167.6	2.4	43.4	137.1	125.5	100.1	1.58
6	157.6	2.4	44.6	143.8	129.0	100.2	2.98
7	147.6	2.4	46.2	150.6	134.6	100.7	1.47
8	137.6	2.4	48.4	157.3	141.1	100.7	1.85
9	127.6	2.4	50.6	164.1	146.8	100.6	1.27
10	117.6	2.4	53.0	170.8	153.8	100.1	1.55
11	107.6	2.4	55.3	177.5	160.2	100.2	0.61
12	97.6	2.4	57.3	184.2	166.7	100.3	2.50
13	87.6	2.4	59.2	190.9	171.8	100.4	1.87
14	77.6	2.4	60.9	197.7	175.8	100.3	1.82
15	67.6	2.4	62.5	204.3	180.6	100.3	1.15
16	57.6	2.4	64.1	211.0	186.4	100.3	1.58
17	47.6	2.4	65.7	217.8	191.0	100.3	1.23
18	37.6	2.4	67.3	224.5	194.9	100.3	1.60
19	27.6	2.4	68.8	231.2	198.8	100.3	2.78
20	17.6	2.4	70.3	237.6	202.6	100.1	2.19
21	7.6	2.4	71.6	243.3	206.4	100.1	1.74
22	-2.4	2.4	72.8	248.4	210.0	100.1	1.44
23	-12.4	2.4	73.8	252.8	213.1	100.1	1.25
24	-22.4	2.4	74.7	256.6	215.7	100.1	0.97
25	-32.4	2.4	75.4	259.7	218.1	100.2	1.62
26	-42.4	2.4	76.0	262.1	219.8	100.3	2.29
27	-52.4	2.4	76.4	263.8	220.9	100.4	1.71
28	-62.4	2.4	76.6	264.8	221.1	100.1	2.92
29	-72.4	3.0	94.6	264.6	218.4	100.3	0.58
30	-82.4	4.2	130.7	264.7	216.0	100.7	1.28
31	-92.4	4.2	130.7	264.7	214.8	100.2	1.21
32	-102.4	4.2	130.7	264.7	214.6	100.2	0.62
33	-112.4	4.2	130.7	264.7	214.6	100.1	0.05
34	-122.4	4.2	130.7	264.7	214.5	100.1	0.07
35	-132.4	4.2	130.7	264.7	214.5	100.1	—
Σ				7385.5	6347.3		max 3.29

The tip section (1) is sized by the geometric height floor rather than by its force requirement, hence its 12.3% force surplus (Section 6.5).

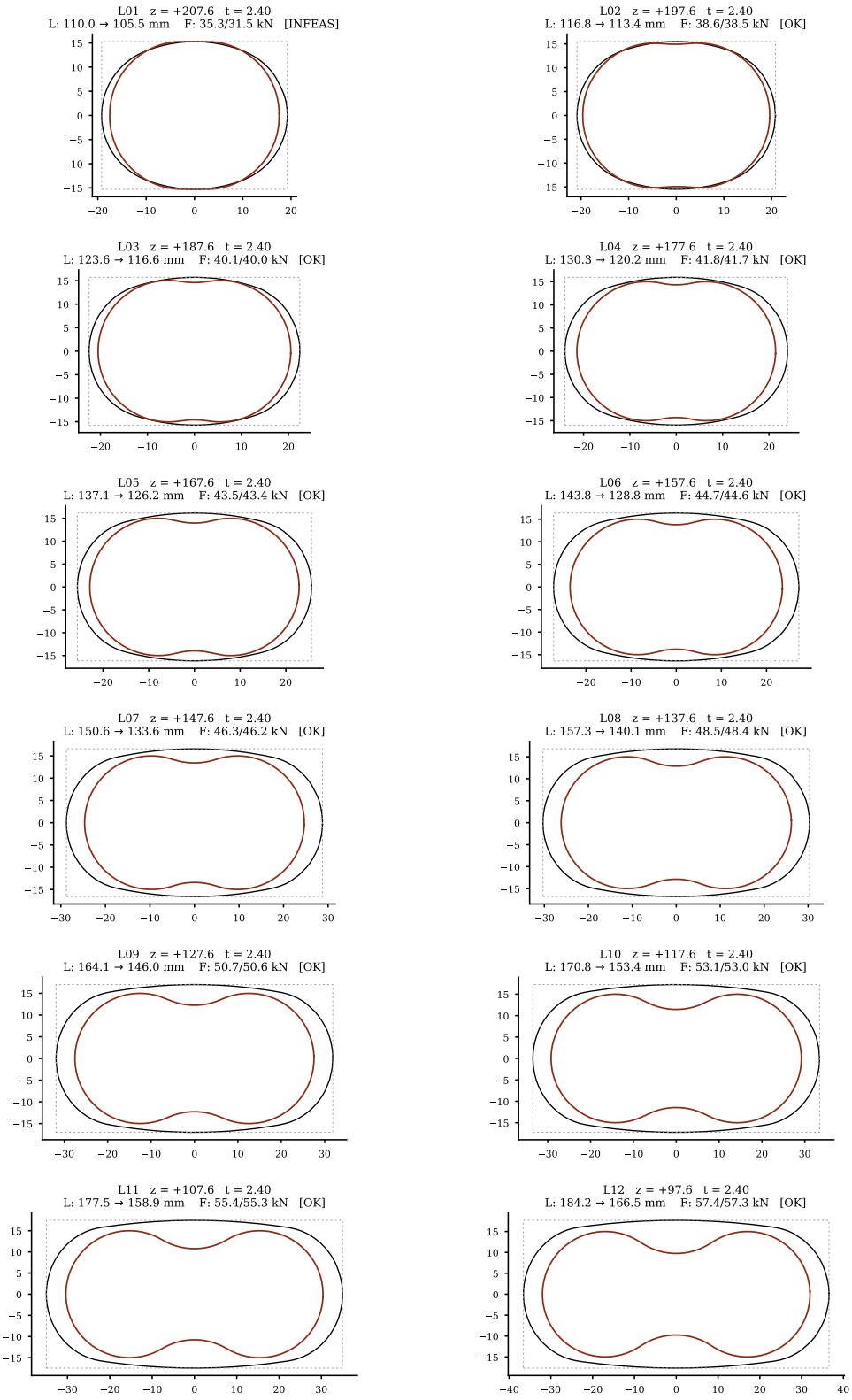


Figure D.1: Per-section optimisation results, sections 1 to 12: extracted seed outline (black) and optimised boundary (dark red); the dashed rectangle marks the seed bounding box.

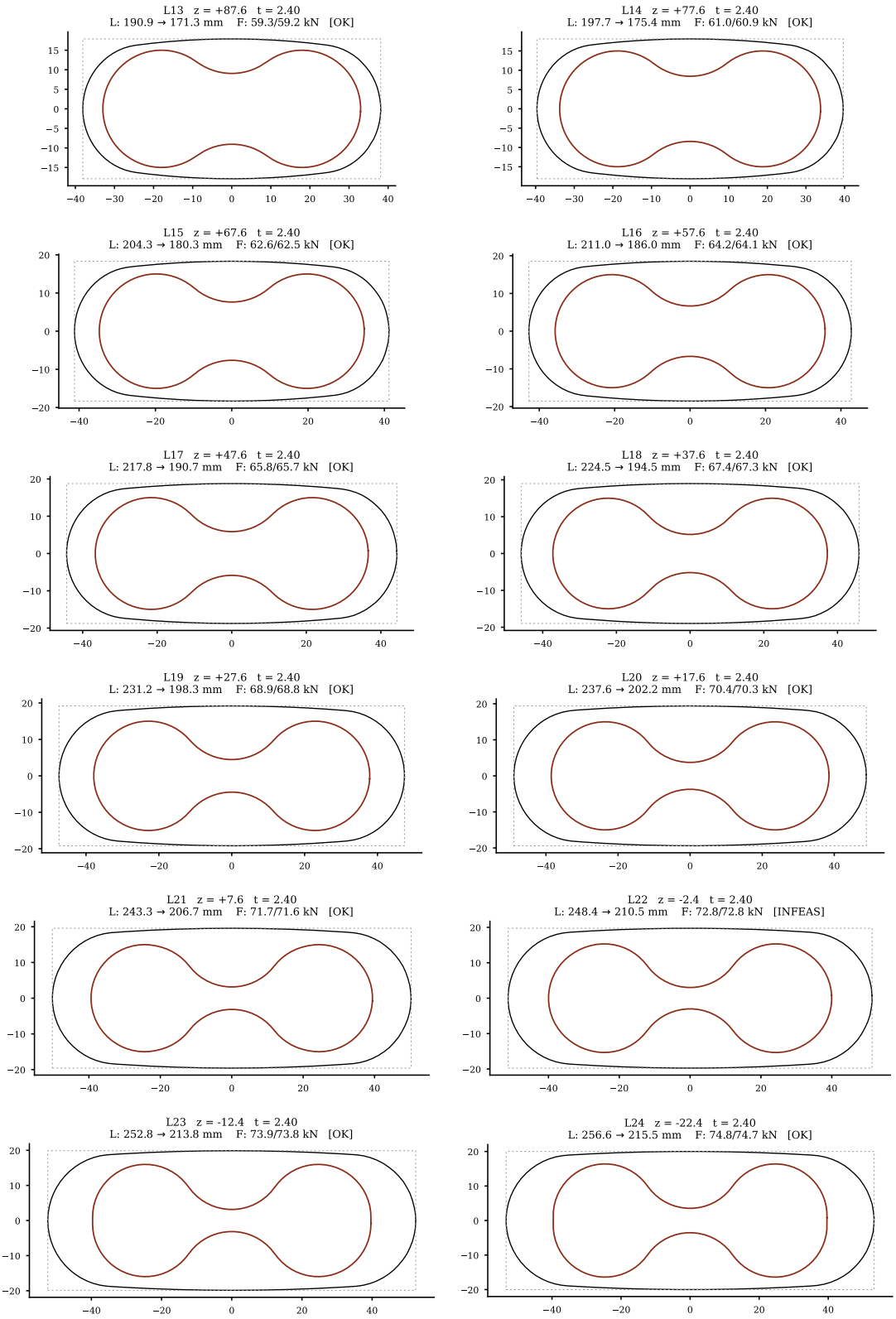


Figure D.1: Per-section optimisation results (continued), sections 13 to 24.

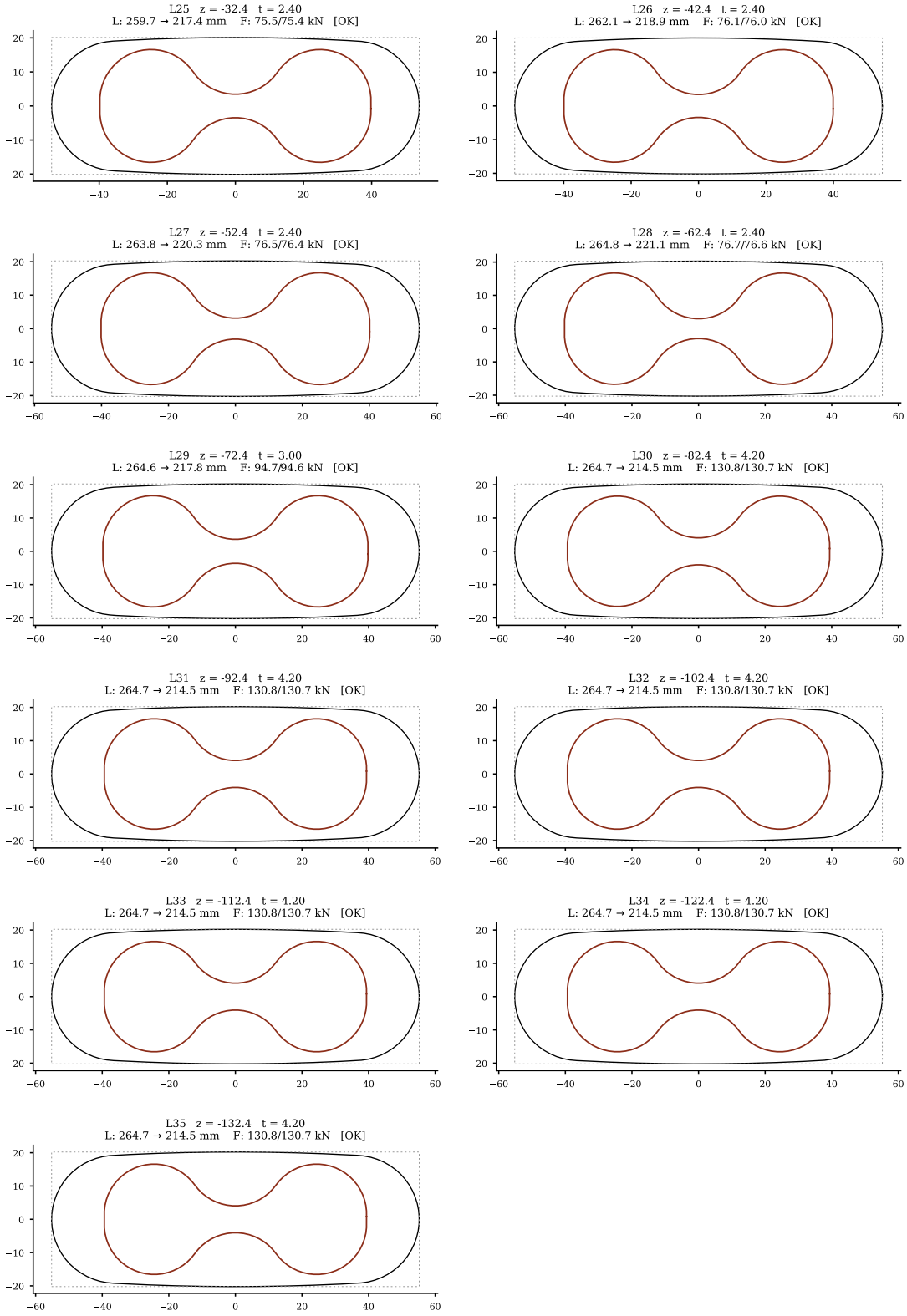


Figure D.1: Per-section optimisation results (continued), sections 25 to 35.

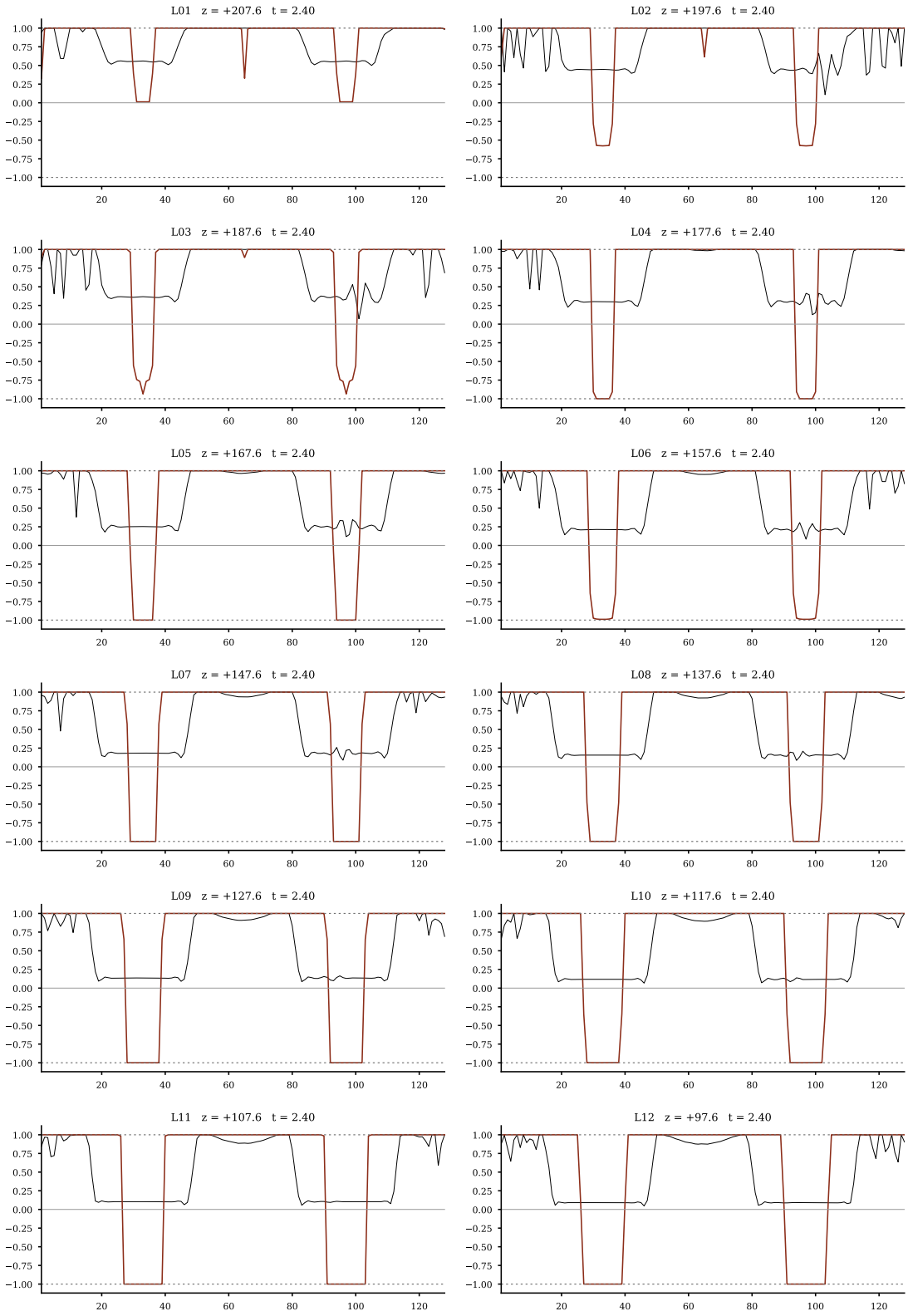


Figure D.2: Per-section curvature profiles, sections 1 to 12: initial seed (black) and optimised section (dark red), as the normalised curvature $\kappa R_{\min}$ against segment index; the dashed lines mark the manufacturing bound ± 1.

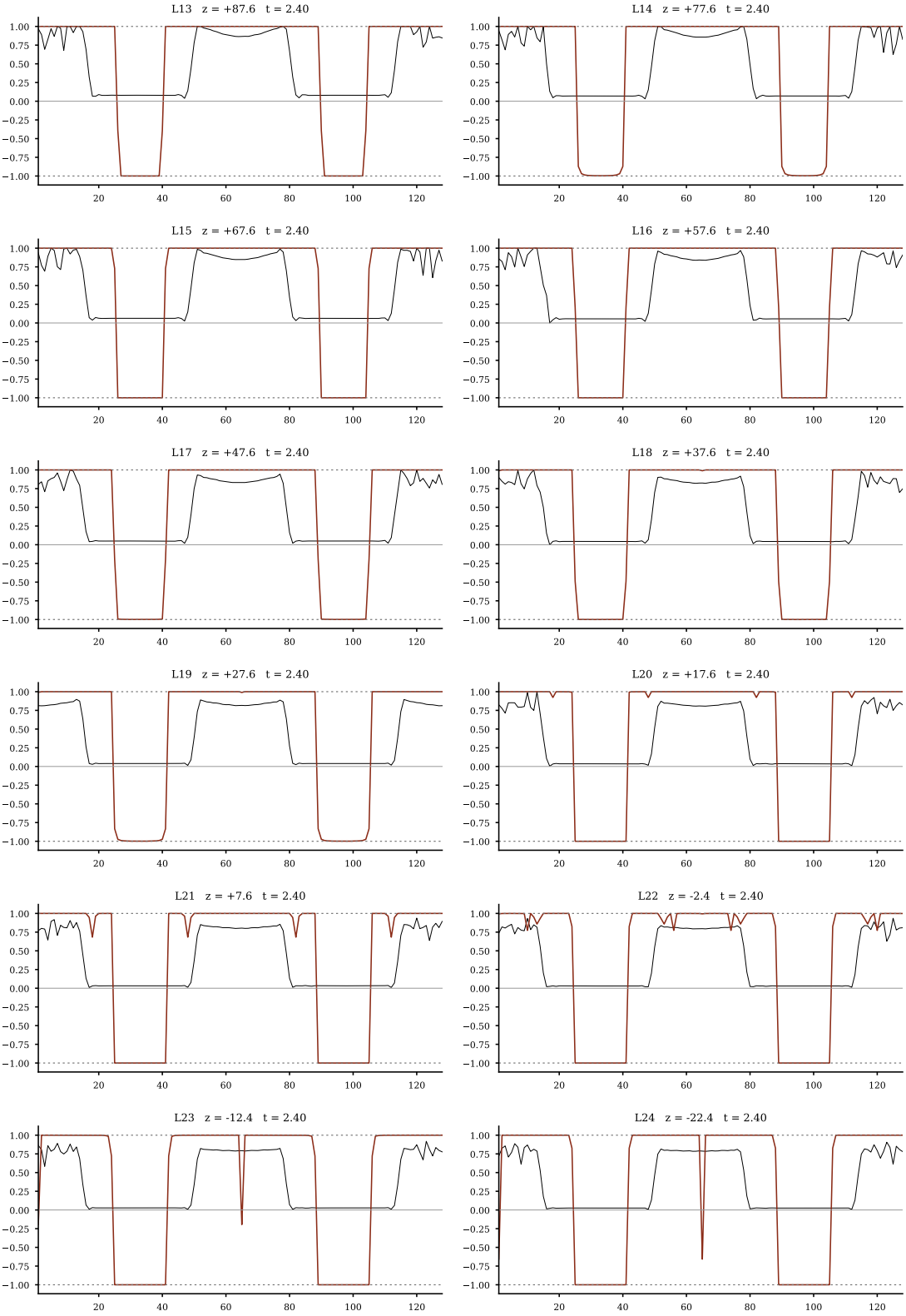


Figure D.2: Per-section curvature profiles (continued), sections 13 to 24.

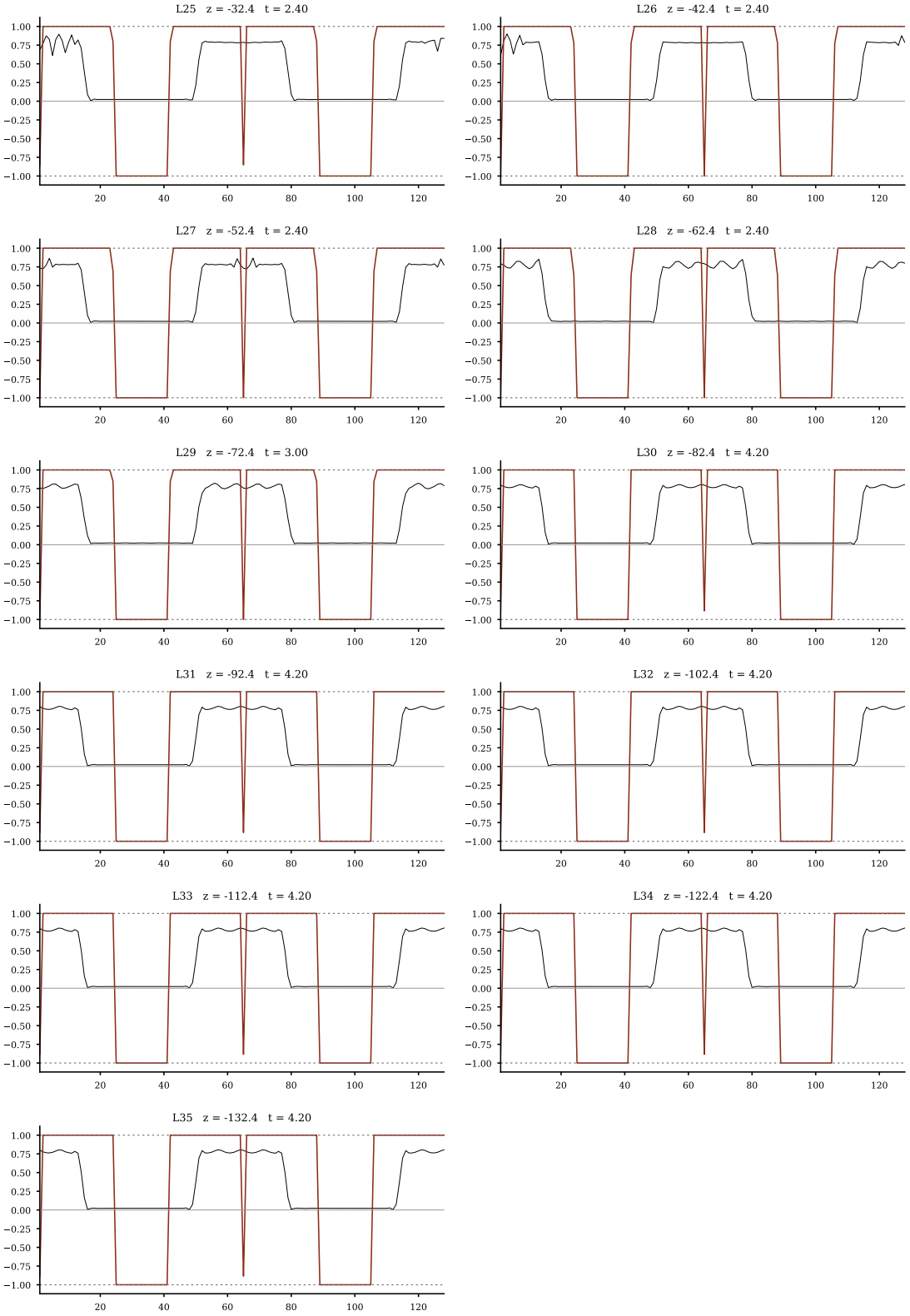


Figure D.2: Per-section curvature profiles (continued), sections 25 to 35.