

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Topology Structural Optimization of Components for the Automotive Industry: Focusing on Hypercar Design and Additive Manufacturing

Miguel Dias de Lima

MASTER IN MECHANICAL ENGINEERING - AERONAUTIC AND
VEHICLE STRUCTURES



Mestrado em Engenharia Mecânica - Estruturas Aeronáuticas e de Veículos

Supervisor: Dr. Ana Isabel Lopes Pais

Co-Supervisor: Dr. Pedro Silva Campos

Supervisor at Adamastor: Eng^o Frederico Tomé Ribeiro

August 6, 2026

Topology Structural Optimization of Components for the Automotive Industry: Focusing on Hypercar Design and Additive Manufacturing

Miguel Dias de Lima

Mestrado em Engenharia Mecânica - Estruturas Aeronáuticas e de
Veículos

August 6, 2026

Abstract

Reducing the mass of a load-bearing structure without compromising its capacity to carry load is a central objective of high-performance vehicle design, and the union of topology optimisation with metal additive manufacturing has made attainable the intricate, organic geometries that conventional fabrication cannot realise. This dissertation develops a bespoke bidirectional evolutionary structural optimisation (BESO) framework, oriented towards selective laser melting (SLM), and proves it on a genuine industrial problem: a suspension component of the Adamastor Furia, reported as the first hypercar to be designed and built in Portugal.

The work comprises three phases. The first is the construction, from first principles, of the optimisation framework, built on a convergent, mesh-independent, soft-kill BESO scheme coupled to a finite-element solver. It carries two engines upon a shared algorithmic core, a solid-void engine for classical topology optimisation and a lattice engine for graded, cellular lightweighting, together with an experimental thermal-proxy feature that biases the optimiser away from geometry left unsupported during the build. The framework is verified against benchmark problems of known solution.

The second phase applies the framework to the Furia’s roll-damper bracket. Driven by the real load cases of the car’s patented decoupled roll-heave suspension mechanism, the optimisation redesigns a heavy steel part, of mass 1.520 kg, as a body of 0.459 kg in SLM AlSi10Mg, a reduction of very nearly 70%. The redesigned part is shown to meet its static strength requirement and, under a representative Spa-Francorchamps duty cycle, to be far from fatigue-limited.

The third phase is exploratory and concerns the cost of the methodology rather than its result. A convolutional encoder-decoder of the U-Net family is trained to reproduce solid-void optima to about 98.5% pixel accuracy, and is then fine-tuned towards the graded output of the framework’s lattice engine. The study finds, in qualified form, that a network pre-trained on the binary task acquires transferable structural knowledge, adapting to the lattice task roughly four and a half times faster than one trained from scratch, with the final accuracy limited not by the initialisation but by the scarce lattice data and by the mismatch between the two optimisers, whose volume fractions differ because the lattice engine takes no prescribed target.

A single principle runs throughout the work: no design produced by any route is accepted on the strength of its method alone, and each is held to independent finite-element validation.

Keywords: *Topology Optimization, BESO, Additive Manufacturing, Selective Laser Melting, Lattice Structures, Machine Learning*

Resumo

Reduzir a massa de uma estrutura sujeita a cargas sem comprometer a sua capacidade de suportar é um objetivo central no projeto de veículos de alto desempenho, e a união da otimização topológica com o fabrico aditivo metálico viabilizou as geometrias complexas e orgânicas que os processos de fabrico convencionais não conseguem realizar. Esta dissertação desenvolve um *framework* próprio de otimização estrutural evolucionária bidirecional (BESO), orientado para a fusão seletiva por laser (SLM), e comprova a sua eficácia num problema industrial real: um componente de suspensão do Adamastor Furia, apresentado como o primeiro hipercarro projetado e construído em Portugal.

O trabalho está estruturado em três fases. A primeira consiste na construção, a partir de princípios fundamentais, do *framework* de otimização, baseado num esquema BESO convergente, independente da malha, e de remoção suave (*soft-kill*), acoplado a um *solver* de elementos finitos. Este integra dois motores sobre um núcleo algorítmico partilhado: um motor *solid-void* para otimização topológica clássica e um motor *lattice* para redução de massa através de estruturas celulares graduadas, em conjunto com uma funcionalidade experimental baseada num modelo térmico, que direciona o otimizador para evitar geometrias não suportadas durante o fabrico. O *framework* é validado face a problemas de referência com solução conhecida.

A segunda fase aplica o *framework* ao suporte do *roll-damper* do Furia. Comandada pelos casos de carga reais do mecanismo de suspensão patenteado e desacoplado em *roll-heave*, a otimização redesenha uma peça pesada de aço, com uma massa de 1,520 kg, convertendo-a num corpo de apenas 0,459 kg em AlSi10Mg processado por SLM, o que representa uma redução de praticamente 70%. Demonstra-se que a peça redesenhada cumpre os requisitos de resistência estática e que, sob um ciclo de carga representativo do circuito de Spa-Francorchamps, está longe de ser limitada pela fadiga.

A terceira fase é de natureza exploratória e foca-se no custo da metodologia em detrimento do seu resultado final. Uma rede codificador-descodificador convolucional da família U-Net é treinada para reproduzir resultados de uma otimização *solid-void* com uma precisão de cerca de 98,5% ao nível do píxel, sendo submetida a um *fine-tuning* direcionado para os resultados graduados do motor *lattice* desenvolvido. O estudo conclui, com as devidas ressalvas, que uma rede pré-treinada na tarefa binária adquire conhecimento estrutural transferível, adaptando-se à tarefa *lattice* cerca de quatro vezes e meia mais rápido do que uma rede treinada do raiz, ficando a precisão final limitada pela dimensão do conjunto de dados e pela incompatibilidade dos dois otimizadores e não pelo ponto de partida da rede.

Um único princípio rege todo o trabalho: nenhum projeto produzido por qualquer via é aceite apenas com base no mérito do seu método, sendo cada um submetido a uma validação independente por elementos finitos.

Palavras-chave: *Otimização Topológica, BESO, Fabrico Aditivo, Fusão Seletiva por Laser, Estruturas em Treliça, Aprendizagem Automática*

Acknowledgements

To my supervisor, Dr. Ana Pais. From having the chance to work with you back in PEC course I always knew that if I'd work in a thesis related to 3D printing, it'd be with you that'd like to do it. Thank you for all the availability you have shown over these past five months, and for always making sure I did not just know the concepts, but truly understood them. I'll be forever grateful for all the time and energy that you have spent helping me.

To my co-supervisor, Dr. Pedro José Silva Campos, for making this partnership with Adamastor possible in the first place, for pushing me to go ahead with the idea and present them my vision for this dissertation. Thank you as well for your feedback at the moments that mattered most, which helped me make some important decisions along the way.

To Adamastor, for handing me the opportunity to work in a real engineering environment and to take on problems in the world of high performance. Thank you for letting me dive deeper into everything automotive engineering has to offer, and for opening your facilities so that I could develop my work.

To Frederico Ribeiro, Engineering Director at Adamastor, who was my right hand in clearing up every question I had and making sure I had all the means to reach my goal, and to Gerardo Menezes, Production Manager at Adamastor, who also helped me source everything my work called for.

To FEUP and the Department of Mechanical Engineering, not only for the facilities that allowed me to carry out this work, but also for being the institution that shaped me into the engineer I am today.

To my friends. It is genuinely crazy to think that we are already closing something that feels like it only just began. I am thankful for every single one of you, and I end this chapter surrounded by the best people I could ask for. I appreciate you deeply, and I sincerely wish you all the best for what comes next.

To my teammates and my coaches, for your patience on the days I had to swap a training session for thesis work, and for all the support and motivation you gave me to see it through. Thank you for all these years together, and for everything still ahead of us - in a matter of weeks, Norway will be ours!!

To Leonor, thank you for being there for me every single day, whether things were going well or not so well, and for motivating me to keep improving. Writing a thesis is not easy, but it sure is a whole lot easier with you!

To my parents, Manuel and Fernanda. Words cannot do justice to the work and sacrifice you have put in to make sure I had every condition I needed and access to the best education, so that I could grow not only as an engineer, but as the person I am. You are my true role models, and every day I am more grateful for everything you have given me!

I hope this work makes you proud.

Miguel

*“Perfection is achieved, not when there is nothing more to add,
but when there is nothing left to take away.”*

Antoine de Saint-Exupéry, *Terre des hommes* (1939)

Contents

1	Introduction	1
1.1	Motivation and Context	1
1.2	The Adamastor Furia	2
1.3	Objectives	3
1.4	Approach	4
1.5	Contributions	4
1.6	Document Structure	5
2	State of Art	7
2.1	Topology Optimisation	7
2.2	Additive Manufacturing and Topology Optimisation	9
2.2.1	Selective Laser Melting	10
2.3	Lattice and Architected Cellular Materials	13
2.3.1	Manufacturability and Real-World Performance	14
2.4	Data-Driven Approaches and Machine Learning	15
2.5	Summary and Positioning	18
3	Theoretical Foundations	19
3.1	Topology Optimisation	19
3.1.1	Rank-Based Selection	21
3.2	Selective Laser Melting	22
3.2.1	Metallic Materials for Selective Laser Melting	27
3.3	Fatigue of Metallic Materials	28
3.3.1	Fatigue of Selectively Laser-Melted AlSi10Mg	30
3.4	Lattice and Architected Cellular Materials	31
3.5	Data-Driven Modelling and Neural Networks	34
4	Development of the BESO Optimisation Framework	37
4.1	BESO Formulation	38
4.2	Software Architecture	41
4.3	Solid-Void Engine	44
4.4	Overhang Mitigation via Thermal Analysis	46
4.4.1	Thermal proxy	46
4.4.2	Sensitivity penalty	47
4.4.3	Directional support promotion	48
4.4.4	Combined update and printability repair	49
4.4.5	Parametric behaviour study	51
4.4.6	Scope of the overhang mitigation feature	56

4.5	Lattice Engine	56
4.5.1	Graded material model	56
4.5.1.1	Property models: Safe and Experimental	59
4.5.2	Utilisation and the fully-stressed target	61
4.5.2.1	Load modes: Scaled and Sized-to-Load	62
4.5.3	The density update and convergence	63
4.5.4	The morphological skin layer	65
4.6	Post-Processing and STL Generation	65
4.6.1	Surface extraction of the solid-void result	66
4.6.2	Geometry generation for the lattice result	67
4.6.2.1	Selection of meshing resolution	68
4.6.2.2	Manufacturability and smoothing	73
4.7	Verification on Benchmark Cases	74
4.7.1	Finite-element and optimisation settings	75
4.7.2	Solid-void benchmarks	76
4.7.2.1	Qualitative overhang-mitigation results	79
4.7.3	Consistency of the lattice engine	82
4.7.3.1	Engine verification through the incremental run sequence	83
4.7.3.2	Robustness to the evolution quota	87
4.7.4	Comparison of the two engines	88
5	Application to the Furia Suspension Component	93
5.1	Suspension Mechanism and Target Component	93
5.1.1	Decoupled roll-heave mechanism	93
5.1.2	Target component	94
5.1.3	Assembly and kinematic modelling	97
5.2	Optimisation Model	97
5.2.1	Design space	97
5.2.2	Finite-element model and boundary conditions	98
5.2.3	Load cases	99
5.3	Mesh Convergence and Sensitivity-Field Rank Study	102
5.3.1	Mesh convergence	102
5.3.2	Sensitivity-field rank preservation	102
5.3.3	Adopted mesh	106
5.4	Optimisation Results	106
5.5	Optimised Geometry	108
5.6	Structural Validation	109
5.6.1	Validation model	109
5.6.2	Static verification	110
5.6.3	Fatigue assessment	112
5.6.3.1	Load history and sign-convention validation	113
5.6.3.2	Stress reconstruction and damage model	113
5.6.3.3	Results	116
5.7	Discussion	119
6	Machine Learning Surrogate for Topology Optimization	121
6.1	Motivation and Problem Framing	122
6.2	Dataset	123
6.3	Network Architecture	124

6.4	Training	125
6.5	Results	127
6.6	Extensions	131
6.6.1	Loss-function and augmentation studies	131
6.6.2	Lattice fine-tuning and the transfer experiment	132
6.6.2.1	The lattice dataset	132
6.6.2.2	Fine-tuned and from-scratch training routes	133
6.6.2.3	Convergence and generalisation results	133
6.6.2.4	Origins of the accuracy ceiling	134
6.7	Discussion and Limitations	135
7	Conclusions and Future Work	139
7.1	Conclusions	139
7.2	Limitations	141
7.3	Future Work	141
A	Supplementary Results for the Machine Learning Surrogate	143
A.1	Full binary run history	143
A.2	Loss-function study	144
A.3	Augmentation study	145
B	Source Code of the Optimisation Pipeline	147
B.1	Overview of the scripts	147
B.2	Availability	148
	Bibliography	149

List of Figures

1.1	The Adamastor Furia	3
2.1	A laser powder-bed-fused titanium acetabular cup with a porous lattice . . .	11
2.2	The additively manufactured chassis of the Czingier 21C	12
2.3	A computationally designed LEAP 71 rocket engine	13
2.4	A taxonomy of machine-learning approaches to topology optimisation: iteration-free design prediction, generative models of the design distribution, learned surrogates for the physics, and neural reparameterisation of the design field.	16
2.5	Generative routes to topology generation: a generative adversarial network, in which a generator is trained against a discriminator, and a denoising diffusion model, which transforms noise into a structure through a sequence of learned denoising steps. Based on the generative approaches of [1] and [2].	16
3.1	Process categories of additive manufacturing (ISO/ASTM 52900).	23
3.2	The Selective Laser Melting process: a scanned, focused laser beam fuses a thin powder layer into a melt pool on the part, while shielding gas sweeps away the vapour plume and spatter; the build platform descends and the powder reservoir rises by one layer thickness per cycle.	24
3.3	Melt-pool behaviour under the laser. At low volumetric energy density the pool is small and fails to fuse, leaving lack-of-fusion voids; at optimal energy density a stable conduction-mode pool fuses into the substrate with Marangoni recirculation and steep heat flow; at high energy density, the metal boils instead of melting, opening deep keyholes, whose walls can become unstable and collapse inward, trapping porosity.	25
3.4	The temperature gradient mechanism of residual-stress formation in SLM, after Mercelis and Kruth [3]: the restrained expansion of the hot top layer yields in compression, and on cooling its contraction leaves tensile residual stress at the surface and an upward distortion.	25
3.5	The staircase effect, in which an inclined surface is approximated by discrete build layers of thickness t ; the deviation from the intended surface grows as the inclination becomes shallower.	26
3.6	Definition of the critical overhang angle θ_c relative to the build plate. Downward-facing surfaces inclined at or above $\theta_c \approx 45^\circ$ are self-supporting, whereas shallower surfaces are unsupported and tend to sag or fail. The value $\theta_c \approx 45^\circ$ is a conventional rule of thumb; the true threshold varies with layer thickness, spot size, energy density, and material.	27

3.7	Sacrificial support structures anchoring a downward-facing overhang to the build plate and providing a thermal path that conducts heat away from the overhanging region.	27
3.8	The two principal families of lattice unit cell.	32
3.9	Bending-dominated versus stretching-dominated unit cells. A cell that forms a mechanism under Maxwell’s criterion, Equation (3.11), deforms by strut bending and is compliant, whereas a statically determinate cell carries load by axial stretching and is far stiffer for the same mass.	32
3.10	Gibson–Ashby scaling of relative stiffness with relative density on logarithmic axes. Stretching-dominated lattices follow a slope of approximately one and bending-dominated foams a slope of approximately two, so the deformation mode, and not merely the amount of material, governs efficiency. . .	33
3.11	Lattice design strategies: a uniform lattice at constant relative density, a functionally graded lattice whose density follows the local stress, and a conformal lattice aligned to the part boundary, optionally enclosed by a solid skin.	34
3.12	The convolutional encoder-decoder (U-Net): a contracting encoder compresses the input to a latent bottleneck, an expanding decoder reconstructs a full-resolution output, and skip connections carry high-resolution detail across the bottleneck.	35
3.13	Supervised training of a topology predictor: the problem specification (domain, loads, supports, and volume fraction) is mapped by the network to a predicted topology, which is compared against the reference optimum through the loss of Equation (3.14).	36
4.1	Architecture of the optimisation framework	43
4.2	The thermal proxy for overhang detection	47
4.3	Directional support promotion	49
4.4	The Trim and Bridge repair strategies	51
4.5	Convergence of the overhang mitigation	52
4.6	The stiffness-overhang trade-off	53
4.7	Effect of the support reach on the trade-off	54
4.8	The Trim and Bridge repair strategies compared	55
4.9	The six lattice morphologies supported by the engine	57
4.10	Gyroid-sheet power-law fit to the polynomial source	59
4.11	Safe and Experimental evaluation of the lattice law	61
4.12	Solid-void surface post-processing	66
4.13	The lattice meshing-resolution law	70
4.14	Cross-section of the I-WP void across the pinch-off	71
4.15	Converged renders at low density	72
4.16	Converged renders at high density	73
4.17	MBB beam benchmark	78
4.18	Short cantilever benchmark	78
4.19	Three-point-bending benchmark	79
4.20	Overhang mitigation showcase on the MBB beam	81
4.21	Lattice run C1: Scaled, Safe, gyroid ligament	84
4.22	Lattice run C2: Sized-to-Load, Safe, gyroid ligament	84
4.23	Lattice run C3: Sized-to-Load, Experimental, gyroid ligament	85
4.24	Lattice run C4: the skinned C3 design	85

4.25	Lattice run C5: Sized-to-Load, Experimental, I-WP sheet	86
4.26	The two engines at 67.0% volume fraction	90
4.27	The two engines at 46.2% volume fraction	91
5.1	The decoupled roll-heave suspension mechanism	94
5.2	The eccentric load path	96
5.3	The original steel bracket	96
5.4	The suspension assembly at the extremes of the rear roll range	97
5.5	The design space	98
5.6	The three load cases	101
5.7	Sensitivity rank correlation against rocker mesh	103
5.8	Agreement of the elements the optimiser acts on	104
5.9	Rank correlation by distance from the pin bore	105
5.10	Sensitivity value scatter, linear rocker elements	105
5.11	Sensitivity value scatter, quadratic rocker elements	106
5.12	Specific-stiffness evolution across the optimisation stages	107
5.13	The optimised geometry before and after smoothing	109
5.14	Static von Mises stress in the three load cases	111
5.15	The pin-bore exceedance is asymmetric	112
5.16	Validation of the roll-spring sign convention.	114
5.17	Fatigue assessment at the hot-spot node.	118
5.18	Fatigue damage and the static concentration coincide	119
6.1	The network architecture used in this chapter	126
6.2	Binary topology predictions on the six evaluation samples	129
6.3	Training history of the final binary network	130
6.4	Training histories of the two lattice routes	137
6.5	Prediction comparison across both tasks and both lattice routes	138

List of Tables

4.1	Gibson–Ashby coefficients for the six lattice morphologies	58
4.2	Benchmark and demonstration runs of the verification study	76
4.3	Absolute stiffness of the solid-void benchmarks	79
4.4	Overhang mitigation showcase results	82
4.5	Lattice consistency runs	87
4.6	Evolution-quota study	88
4.7	Specific stiffness of the two engines at matched volume fractions	89
6.1	Progression of the retained architecture and training refinements	128
6.2	Summary of the loss-function and augmentation extensions	132
6.3	Comparison of the fine-tuned and from-scratch lattice networks	134
A.1	Full binary topology run history	144
A.2	Per-sample results of the loss-function study	145
A.3	Per-sample results of the augmentation study	146
B.1	Source files of the optimisation pipeline	148

Document notation

Abbreviations and Acronyms

AM	Additive Manufacturing
AG	Attention Gate
ASTM	American Society for Testing and Materials
BCC	Body-Centred Cubic
BCE	Binary Cross-Entropy
BESO	Bidirectional Evolutionary Structural Optimization
BN	Batch Normalisation
CTF	Connector total force (components CTF1 to CTF3)
CTM	Connector total moment (components CTM1 to CTM3)
DSC	Dice Similarity Coefficient
ED	Encoder-Decoder
ESO	Evolutionary Structural Optimisation
FCC	Face-Centred Cubic
FEM	Finite Element Method
FKM	Forschungskuratorium Maschinenbau (fatigue assessment guideline)
GAN	Generative Adversarial Network
ISO	International Organization for Standardization
LPBF	Laser Powder Bed Fusion
MAE	Mean Absolute Error
MBB	Messerschmitt-Bölkow-Blohm (topology optimisation benchmark beam)
MSE	Mean Squared Error
NT11	Nodal temperature (Abaqus field output)
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
ReLU	Rectified Linear Unit
sig	Sigmoid (logistic activation function)
RF	Reaction force (Abaqus output)
RP	Reference point (Abaqus)
SCNP	Same-Class Neighbour Penalisation
SIMP	Solid Isotropic Material with Penalisation
SLM	Selective Laser Melting
S-N	Stress-life (fatigue) curve
STL	Standard Tessellation Language (stereolithography mesh format)
TPMS	Triply Periodic Minimal Surfaces
VAE	Variational Autoencoder
VF	Volume fraction

Roman letters

a	Transverse-to-longitudinal conductivity ratio (overhang model)
A	Pixel accuracy
A_k	Top- k set agreement (rank study)
b	Number of struts (Maxwell's criterion)
b_i	Boundary indicator at pixel i
c	Structural compliance
C_1	Gibson–Ashby stiffness coefficient
C_2	Gibson–Ashby strength coefficient
d	Weighted-mean load-case displacement (specific stiffness)
d_i	Rank difference of element i (Spearman)
D	Cumulative fatigue damage (Palmgren–Miner)
E	Young's modulus
E_0	Young's modulus of the fully solid material
E_e	Effective Young's modulus of an element
E_s	Young's modulus of the parent (solid) material
E_v	Volumetric energy density
E^*	Effective (homogenised) Young's modulus of the lattice
ER	Evolutionary ratio (per-iteration void-fraction increment)
f	Prescribed maximum allowable volume fraction
f_θ	Surrogate network with parameters θ
F	External load vector
$F_{G/P/D/N}$	TPMS implicit fields: gyroid, Schwarz primitive, diamond, Neovius (with F_{IWP} for the I-WP)
F_{heave}	Heave-spring force channel
F_{push}	Pushrod force channel
F_{ref}	Reference force for unit-load scaling
$F_{\text{roll}}^\pm$	Roll-spring force channel (right/left corner)
h	Hatch spacing
h_e	Analysis element size (lattice sub-grid)
I_1	First stress invariant
j	Fatigue safety factor
K	Global stiffness matrix
k_e	Element stiffness matrix
L	Number of load cases
$\mathcal{L}$	Combined training loss
ℓ	Unit-cell period (TPMS lattice)
m	TPMS field metric (F for ligaments, $ F $ for sheets)
m_e	Elemental mechanical sensitivity (overhang blend)
M	Maxwell's criterion (kinematic mode)
n	Number of frictionless nodes (Maxwell); sample size (rank study)
n_c	Number of concordant pairs (Kendall)
n_d	Number of discordant pairs (Kendall)
n_E	Gibson–Ashby stiffness exponent
n_i	Applied cycles in stress bin i
n_s	Sub-grid samples per unit cell
n_σ	Gibson–Ashby strength exponent

N	Number of pixels in the prediction grid
N_f	Cycles to failure
N_i	Filter neighbourhood of element i
N_p	Sub-grid points per unit cell
P	Laser power
p	Penalisation exponent
q	Directional-sharpness exponent (overhang casting)
r_{ij}	Distance between the centroids of elements i and j
$r_{\min}$	Filter radius
$r^{(t)}$	Void fraction at iteration t
R	Stress ratio, $\sigma_{\min}/\sigma_{\max}$ (fatigue)
s_L	Load-mode sizing factor (fully-stressed design)
$S_k, \hat{S}_k$	Top- k element sets of two rankings (rank study)
t	Layer thickness (SLM process parameter)
t_{iso}	Iso-value threshold (TPMS level set)
t_w	Overhang trade-off weight
T_e	Elemental thermal-proxy temperature
T_{ref}	Reference temperature (95th percentile over solid elements)
U	Global displacement vector
$\mathbf{u}_e$	Element displacement vector
$\mathcal{U}_e$	Lattice element utilisation (stress over local capacity)
V_0	Total volume of the design domain
v	Laser scan speed
v_e	Individual element volume
w_k	Load-case weight
$w(r_{ij})$	Filter weight function
$\mathbf{x}$	Vector of elemental design variables
x_e	Elemental pseudo-density design variable
$x_{\min}$	Lower bound for element density
$\hat{y}_i$	Predicted value at pixel i
y_i	Target value at pixel i

Greek letters

α_e	Elemental sensitivity number
$\alpha_e^{(k)}$	Elemental sensitivity for load case k
$\tilde{\alpha}_i$	Filtered sensitivity number
$\bar{\alpha}_i$	History-averaged sensitivity number
β	Basquin fatigue-strength exponent (negative)
β_e	Elemental overhang casting sensitivity
γ	Load direction angle
Δ	Move limit (lattice density update)
ζ	Scaled coordinate ωz (TPMS field)
η	Scaled coordinate ωy (TPMS field)
θ	Network parameters (weights)
θ_c	Critical overhang angle
κ	Specific stiffness
λ	Taubin smoothing factor
$\lambda_{\parallel}$	Longitudinal (build-axis) thermal conductivity
$\lambda_{\perp}$	Transverse thermal conductivity
μ	Taubin smoothing parameter (second pass)
ν	Poisson's ratio
ξ	Scaled coordinate ωx (TPMS field)
ρ	Spearman's rank correlation coefficient
$\bar{\rho}$	Relative density
$\bar{\rho}_c$	Cutoff relative density (solid above it)
$\bar{\rho}_e$	Elemental relative density
$\bar{\rho}_e^*$	Target elemental relative density (fully-stressed)
ρ^*	Effective density of the cellular solid
ρ_i	Relative density of element i , in per cent
ρ_s	Density of the parent material
$\sigma(\mathbf{x}, t)$	Reconstructed stress field (superposition)
$\sigma_{\text{rL/rR/p/h}}$	Unit-load stress fields (roll-left, roll-right, pushrod, heave)
σ^*	Effective yield strength
$\bar{\sigma}_e$	Elemental von Mises stress
σ_a	Stress amplitude
σ_{ar}	Equivalent fully-reversed stress amplitude
σ'_f	Fatigue-strength coefficient (Basquin)
σ_g	Gaussian filter standard deviation
σ_{ij}	Cartesian stress components (σ_{11} to σ_{23})
σ_m	Mean stress
σ_u	Ultimate tensile strength
σ_{vM}	Von Mises stress
σ_y	Yield strength (Soderberg limit; local lattice yield $\sigma_y(\bar{\rho}_e)$)
σ_{ys}	Yield strength of the parent material
$\tilde{\sigma}$	Signed von Mises stress
τ	Kendall's rank correlation coefficient
τ_e	Elemental overhang penalty (filtered)
ϕ	Retained mass fraction
ψ	Attention-gate spatial map

ω Cell angular frequency, $2\pi/\ell$

Chapter 1

Introduction

Reducing the mass of a load-bearing structure while preserving its ability to carry load is among the oldest objectives of mechanical design, and it remains a first-order performance driver wherever the cost of carrying weight is high, in aerospace and in high-performance motorsport alike. Two developments have transformed what is achievable in its pursuit. Topology optimisation provides a rational means of deciding where material should be placed within a structure so that performance is maximised for a given mass, and additive manufacturing, and metal powder-bed fusion in particular, provides the means to build the intricate, organic geometries that such an optimisation produces and that conventional fabrication cannot realise. This dissertation is concerned with the union of the two, and with proving it on a genuine engineering problem: it develops a bespoke bidirectional evolutionary optimisation framework, applies it to a real structural component of a hypercar developed in partnership with the Portuguese manufacturer Adamastor, and undertakes an exploratory study of whether a neural network can learn to reproduce the optimiser's output. The work therefore falls into three phases, the construction of the optimisation framework, its application to the component, and the machine-learning study, which the present chapter sets in context before the body of the dissertation develops each in turn.

1.1 Motivation and Context

The motivation for lightweighting is at root mechanical. In any vehicle, mass carried is mass that must be accelerated, braked, and supported, so that its reduction improves acceleration, handling, range and efficiency at once, which is why it is pursued so consistently in the sectors where those measures carry a premium. Historically, the designer's freedom to remove material was bounded not by mechanics but by manufacture: a structure could be lightened only as far as it could still be cast, forged or machined, and the more efficient a geometry became the harder it tended to be to produce. The two technologies at the centre of this work each relax one of those bounds.

Topology optimisation addresses the mechanical side of the problem. Rather than refining a shape the designer has already conceived, it treats the layout of the structure itself

as the unknown, distributing material across a reference domain so that a performance measure is extremised under a constraint on the amount of material used. Its solutions place material along the load paths and leave the rest void, and they are frequently organic and multiply connected in a way no human designer would draw by hand. Additive manufacturing addresses the manufacturing side. By building a part through the successive fusion of material rather than the removal of a billet, it decouples geometric complexity from manufacturing cost to a degree no subtractive process can match, and so allows the geometries that topology optimisation produces to be built rather than simplified back towards manufacturability. Selective laser melting, the metal powder-bed fusion process adopted throughout this work, is the route by which the optimised component of this dissertation is ultimately realised.

It is the coupling of these two, optimisation freed to exploit the full geometric latitude that additive manufacturing provides, that has moved from laboratory demonstration into established engineering practice over the past decade, in fields as varied as orthopaedic implants [4], aero-engine components [5], and the structural nodes of high-performance vehicles [6]. The full survey of that practice, and of the lattice and data-driven methods that accompany it, is deferred to Chapter 2; here it suffices to say that the union is mature enough to justify building upon it.

The methodology is not, however, without cost. Both the optimisation and its validation rest on repeated finite-element analysis, and because an optimiser must evaluate a great many candidate designs, the total computational expense grows unfavourably with the fidelity demanded of each evaluation. This escalation is the practical opening for the data-driven methods that form the third strand of this work, in which a neural network is trained to approximate the expensive mapping from a problem’s specification to its optimised layout. It is against this background, of a mature but costly design methodology, that the present dissertation is set, and it is a concrete industrial problem that gives it its focus.

1.2 The Adamastor Furia

The Adamastor Furia is a two-seat, rear-wheel-drive hypercar, developed by the firm Adamastor near Porto and reported as the first vehicle of its class to be designed and built in Portugal. It is built around a carbon-fibre monocoque and powered by a mid-mounted, twin-turbocharged V6 derived from the Ford GT [7]. Its design is governed by the objectives characteristic of the hypercar class: a low kerb mass, of the order of a thousand kilograms, and the aerodynamic and mechanical performance required of a vehicle intended for both road and track use, with mass reduction pursued consistently across the structure to those ends. Produced in a limited series of sixty units, the Furia represents a substantial engineering undertaking for a low-volume national manufacturer, and it is within this development effort that the present work is situated.



Figure 1.1: The Adamastor Furia, described by its maker as the first Portuguese hypercar. Its low kerb mass, achieved through a carbon-fibre structure and a weight-driven design process, is the priority that motivates the lightweighting study of this dissertation. Image: Adamastor [7].

The partnership with Adamastor gave access to a genuine, performance-critical structural component of the car, and to the loads, geometry and material history behind it, and posed a concrete question: whether a topology-optimisation framework, verified on benchmark problems, could take a heavy, conservatively designed production part and return a substantially lighter one that is manufacturable by selective laser melting and defensible under the car’s real load cases. The component chosen for this is a bracket of the Furia’s suspension, originally a steel part whose mass is heavy for its function. The framework developed in this dissertation is applied to it in Chapter 5, and it is the concrete problem against which the whole of the work is ultimately measured.

1.3 Objectives

The dissertation pursues three objectives, corresponding to its three phases. The first is to develop, from the ground up, a topology-optimisation framework based on the convergent, mesh-independent, soft-kill bidirectional evolutionary scheme, coupled to a finite-element solver and capable of both classical solid-void optimisation and the graded, lattice-based optimisation used for mesoscale lightweighting. The framework is developed with selective laser melting as the intended manufacturing route, and to that end it also provides an experimental, physics-based feature that biases the optimiser away from geometry that would be unsupported during the build.

The second objective is to apply this framework to the Furia’s roll-damper bracket: to optimise the component under its real load cases, to realise the optimised result as a body manufacturable in AlSi10Mg by selective laser melting, and to validate the redesigned part, both by static analysis and against a representative fatigue duty cycle.

The third objective is exploratory, and concerns the cost of the methodology rather than its result. It is to investigate whether a convolutional neural network can learn the mapping from a problem’s physical specification to its optimised topology, and, in particular, to test the hypothesis that a network trained on one class of optimised structures acquires

representations that transfer efficiently to another.

1.4 Approach

The approach taken to each objective is stated briefly here and developed in full in the chapters that follow. The optimisation framework is implemented as a custom driver in Python, which couples the evolutionary update to the Abaqus finite-element solver and manages a complete run from an input model through to printable geometry; its two engines share a single algorithmic core and diverge only in how they represent and update material. The application to the bracket follows a defined workflow: a kinematic model of the patented suspension mechanism establishes the loads the component must carry, these load cases drive the finite-element model and the optimisation, and the optimised result is converted into a manufacturable body and then validated. The machine-learning study trains a convolutional encoder-decoder of the U-Net family, in a supervised manner, to reproduce the optima of a verified public dataset, and then fine-tunes it towards the output of the framework’s own lattice engine in order to test the transfer hypothesis.

A single principle runs through all three phases. No design produced by any route is accepted on the strength of the method alone; each is held to independent finite-element validation, and the manufacturing constraints of the intended process are treated as real limits rather than idealised assumptions.

1.5 Contributions

The principal contribution of this dissertation is the optimisation framework itself, built and developed from first principles rather than assembled from an existing commercial or open-source implementation, and carrying both a solid-void and a lattice engine, together with an experimental thermal-proxy feature for overhang mitigation, upon a shared code base.

Applying this framework to the Furia’s roll-damper bracket yields the second contribution: a redesign of a genuine, performance-critical component that reduces its mass from 1.520 kg in steel to 0.459 kg in selective-laser-melted AlSi10Mg, a reduction of very nearly seventy per cent, and that is shown to meet its static requirement and to be far from fatigue-limited under a representative Spa-Francorchamps duty cycle.

The third contribution is the finding of the machine-learning study: evidence, in qualified form, that a network pre-trained to reproduce solid-void optima acquires transferable structural knowledge, adapting to the distinct problem of lattice-density prediction markedly faster than a network trained from scratch, with the final accuracy bounded not by the initialisation but by the scarcity of the available data and by the structural mismatch between the two optimisers outputs.

1.6 Document Structure

The remainder of the dissertation is organised as follows. Chapter 2 reviews the state of the art across the four fields on which the work draws, topology optimisation, additive manufacturing, lattice structures, and machine learning for structural design, as a narrative of how each has developed and where it stands in engineering practice. Chapter 3 supplies the formal counterpart to that narrative, setting out the definitions, governing equations, and physical mechanisms on which the work rests. Chapter 4 presents the first phase of the work, the development of the optimisation framework, from its shared formulation and software architecture through its two engines and the overhang-mitigation feature, and closes by verifying it against benchmark problems of known solution. Chapter 5 presents the second phase, the application of the framework to the Furia’s roll-damper bracket, following the component from its original steel design through optimisation, manufacture-oriented realisation, and structural validation. Chapter 6 presents the third phase, the machine-learning surrogate and the transfer-learning study. The final chapter draws together the conclusions of the three phases and sets out the directions for future work.

Chapter 2

State of Art

This chapter reviews the four research areas on which the remainder of this dissertation builds, together with the relations between them, considering in each case how the field has developed and where it currently stands in engineering practice. It traces topology optimisation from its analytical origins to the bidirectional evolutionary scheme adopted here (Section 2.1); additive manufacturing, and selective laser melting in particular, as the technology that frees optimised geometry from the constraints of conventional fabrication, together with a survey of the topology-optimised and lattice-based parts already in service (Section 2.2); the establishment of lattice structures into a recognised lightweighting strategy (Section 2.3); and the data-driven and machine-learning methods that promise to accelerate the costly optimisation and evaluation steps (Section 2.4). The concepts introduced in this review are presented qualitatively; their formal definitions, governing equations, and physical mechanisms are developed in the theoretical foundations of Chapter 3, to which the reader is referred at each turn.

2.1 Topology Optimisation

Structural optimisation admits three levels of design freedom, of which topology optimisation is the most general: it fixes neither the shape nor the connectivity of the structure in advance, but decides, across a reference domain, where material should be placed to extremise a performance measure under a constraint on the amount of material used. The formal statement of the problem is given in Section 3.1.

The modern computational era of the field is generally regarded as having begun with the homogenisation method of Bendsøe and Kikuchi [8], which made the otherwise ill-posed material distribution problem tractable by admitting a graded composite microstructure whose properties vary continuously between solid and void. The designs it produced contained large regions of intermediate, porous material that were difficult to interpret and to manufacture. The density approach that followed shortly afterwards, due to Bendsøe [9] and Zhou and Rozvany [10], recovered clearer solid-void layouts by assigning each element a single pseudo-density and penalising intermediate values towards solid or void.

Generalised and given its now standard name, the Solid Isotropic Material with Penalisation (SIMP) model, by Rozvany et al. [11], and consolidated in its material-interpolation foundations by Bendsøe and Sigmund [12], this became the dominant paradigm of the field. Its accessibility owes a great deal to Sigmund [13], whose compact ninety-nine-line implementation made density-based topology optimisation widely reproducible and turned it into a standard teaching and research tool.

Density methods brought their own numerical pathologies, catalogued by Sigmund and Petersson [14]: checkerboard patterns of alternating solid and void elements, mesh dependence in which finer meshes yield qualitatively different designs, and local minima. The accepted remedy, filtering the design over a prescribed radius, is now considered an essential ingredient of any robust implementation. A distinct line of development pursued the boundary itself rather than a density field. In the level-set approach, introduced for structural problems by Wang et al. [15] and, independently, by Allaire et al. [16], the structure is represented implicitly as the zero contour of a higher-dimensional scalar function, which is evolved according to shape-sensitivity information. Level-set methods yield crisp, smooth boundaries free of intermediate grey material, at the cost of a greater tendency to settle in local minima and a reduced ability to nucleate new holes within the classical Hamilton-Jacobi framework. They remain an important parallel paradigm but are not pursued in this work.

The branch most directly relevant to the present thesis is the evolutionary one. The Evolutionary Structural Optimisation (ESO) method, introduced by Xie and Steven [17] and consolidated in Xie and Steven [18], rests on an intuitive premise: a structure tends towards an optimum if the material that is least effectively used is gradually removed. A recognised limitation, noted critically by Zhou and Rozvany [19], is that this hard-kill, removal-only strategy can converge to designs that are demonstrably non-optimal, since an element discarded early can never be recovered. This shortcoming motivated the bidirectional extension. Querin et al. [20] proposed Bidirectional Evolutionary Structural Optimisation (BESO), which allows material to be added back into regions where the sensitivity numbers indicate it is needed, as well as removed where it is not. Early BESO formulations still suffered from the mesh dependence and convergence instabilities inherited from ESO; the decisive advance was made by Huang and Xie [21], whose convergent and mesh-independent framework filters the elemental sensitivities, stabilises the optimisation history, and adopts a soft-kill scheme in which void elements retain a small residual stiffness rather than being purged from the mesh. The comprehensive treatment of this mature method is given by Huang and Xie [22]. It is this convergent, mesh-independent, soft-kill BESO that provides the algorithmic foundation for the optimiser developed in this work; the ingredients that make it convergent are formalised in Section 3.1, and its implementation is presented in detail in Chapter 4.

2.2 Additive Manufacturing and Topology Optimisation

Topology optimisation is, in principle, manufacturing-agnostic: it distributes material to extremise a performance measure and applies equally to parts destined for casting, forging or machining. In practice, however, its design freedom has long outrun what those conventional processes can realise. The organic, multiply-connected geometries that emerge from a compliance minimisation are frequently impossible to mould, draw or mill without violating draft angles, tool access or undercut limitations, so the optimised result has historically had to be simplified back towards manufacturability, eroding much of the gain. Additive manufacturing is the principal technology shift that relaxes this barrier. By building a part through the successive addition of material rather than the removal or forming of a billet, it decouples macro-geometric complexity from manufacturing cost to a degree no subtractive or formative process can match, allowing topology optimisation to be exploited closer to its full theoretical potential. It is in this sense, as the enabler that lets highly complex optimised geometry be built rather than as the only route to it, that additive manufacturing and topology optimisation have grown together. The process on which this work centres, selective laser melting, and the physics and manufacturability constraints that govern it, are set out in Section 3.2; the present section reviews how those freedoms and constraints have shaped the state of the art and the parts it has produced.

Because sacrificial supports are costly and the simple critical-angle rule (Section 3.2) is a blunt instrument, a substantial body of work has sought to integrate overhang constraints directly into the topology optimisation framework, so that the optimiser produces self-supporting, print-ready designs rather than layouts that must afterwards be propped. This effort forms a clear narrative of its own, and it is overwhelmingly geometric in character. The first proposal to fold an overhang criterion into the optimisation came from Brackett et al. [23], whose result was not fully self-supporting. Gaynor and Guest [24] advanced the idea with a wedge-shaped projection filter that enforced the self-supporting condition during optimisation. Langelaar [25, 26] formalised the approach into a now widely used additive manufacturing filter, a layer-by-layer geometric operation that maps an arbitrary density field onto the nearest print-ready one. Qian [27] took a complementary route, controlling undercut and overhang angles through a density-gradient integral measure, and around these sit the explicit and level-set formulations of Allaire et al. [28].

These geometric methods share a common tension that motivates the alternative branch. At one extreme, the bare forty-five-degree rule is trivial to state but crude: it is conservative, ignores the part's mechanical context, and discards designs that would in fact print acceptably. At the other extreme, the more faithful geometric formulations become progressively more intricate to derive, more delicate to tune, and more expensive to evaluate as they pursue accuracy, particularly in three dimensions. Geometric overhang control, in short, tends to be either too coarse to be useful or too heavy to be convenient. A distinct and more recent branch sidesteps the geometry altogether and reasons instead from the process physics: rather than asking whether a surface is geometrically self-supporting,

these methods ask what the thermal and mechanical state of the part will be during the build, and constrain the design accordingly. The foundational step was taken by Allaire and Jakabčín [29], who incorporated the thermal residual stresses arising during the build directly into the topology optimisation problem. The branch has since developed in several directions: Miki and Yamada [30] and Misiun et al. [31] formulated topology optimisation under distortion constraints, and Pellens et al. [32] optimised support layouts while accounting for thermal deformation. At the more elaborate end these formulations couple a transient thermomechanical simulation into the optimisation loop; at the lighter end, the same physical insight can be exploited through far simpler proxies, such as a steady-state thermal analysis. It is at this accessible, proxy-based end of the physics-based branch that the overhang-mitigation approach developed in this thesis is situated; its formulation and rationale are presented in Chapter 4.

2.2.1 Selective Laser Melting

The high per-part cost of metal SLM, set out in Section 3.2.1, confines it to applications where geometric complexity, part consolidation and lightweighting repay the premium. That threshold is crossed today across several high-value sectors, in each of which topology-optimised or lattice-based selective-laser-melted parts have moved from demonstration into service.

The clearest demonstration in service is in the biomedical field, where selective laser melting of titanium has become an established route to load-bearing orthopaedic and spinal implants. The attraction is twofold. An architected porous lattice, formed by the same layerwise process that builds the solid part, can be tuned so that its effective stiffness approaches that of bone rather than that of bulk titanium, which relieves the stress shielding that causes bone to resorb around an over-stiff implant; and the open, interconnected porosity of a surface-based cell promotes osseointegration and bone ingrowth into the device [4]. The Gibson–Ashby scaling of Section 3.4 is used directly to select the cell morphology and relative density that match the mechanical and biological requirements of the host bone [4, 33], and commercial Ti-6Al-4V acetabular cups and spinal cages built by SLM with porous lattice surfaces are now routine clinical hardware (Figure 2.1). This is precisely the functionally graded, ingrowth-oriented use of a printed lattice anticipated by the design strategies of Section 3.4.

In high-performance automotive engineering the same technology is turned to the opposite priority, mass reduction. Two exemplars bracket the field. Bugatti’s titanium brake caliper, first shown in 2018, is the canonical single-part demonstration: printed in Ti-6Al-4V on an SLM machine in a build lasting some forty-five hours, the eight-piston monobloc caliper weighs 2.9 kg against 4.9 kg for its machined aluminium predecessor, a reduction of about forty per cent, while reaching a tensile strength of 1250 N/mm^2 at over 99.7 per cent density [35]. At the other extreme, Divergent Technologies has made additive manufacturing the basis of an entire production system: the Czinger 21C hypercar is built around a



Figure 2.1: A commercially available acetabular cup for total hip replacement (Trident II Tritanium, Stryker), produced from Ti-6Al-4V by laser powder bed fusion. The bone-facing surface carries an integrally printed porous lattice whose interconnected porosity approaches that of trabecular bone, lowering the effective stiffness to limit stress shielding and offering a scaffold for bone ingrowth. Adapted, showing a single design from a larger comparison, under CC BY 4.0 from Dall’Ava et al. [34].

chassis and suspension of topology-optimised, 3D-printed nodes (Figure 2.2) produced and assembled through the Divergent Adaptive Production System, a design-and-print pipeline the company reports can cut a vehicle’s chassis mass by some thirty to forty per cent while collapsing assemblies of many separate parts into a mere handful of printed structures [6]. The same platform now produces the externally visible front suspension wishbones of the McLaren W1 [36], a close industrial parallel to the optimised suspension component pursued in this dissertation. Applications of this kind rest on the coupling of topology optimisation with lattice infill that has been demonstrated for series automotive parts, such as the topology-optimised, lattice-reinforced steering-column support of Mantovani et al. [37].

Aerospace was the sector that first carried a topology-informed, additively manufactured part into high-volume production. The fuel nozzle of the CFM LEAP engine, printed by GE Aerospace, consolidates around twenty separately brazed and welded components into a single part that is some twenty-five per cent lighter and considerably more durable, and it has since been produced in six-figure quantities for engines powering the Airbus A320neo and Boeing 737 MAX [5]. Structural airframe brackets have followed the same route, in which selective laser melting is coupled with topology optimisation specifically to lighten load-bearing fittings [39], and the mechanical behaviour of the two dominant printable lightweight alloys, AlSi10Mg and Ti-6Al-4V, in this setting is now well characterised [40]. In space propulsion the trend is towards printing whole rocket-engine assemblies as near-monolithic parts (Figure 2.3). A recent illustration couples computational design



Figure 2.2: The rolling chassis of the Czinger 21C hypercar. The organically shaped metallic nodes that link the front and rear suspension and the powertrain to the central carbon-fibre tub are topology-optimised and produced by metal additive manufacturing through the Divergent Adaptive Production System; their skeletal, bone-like form is the geometric signature of the underlying compliance-driven optimisation. Image: Czinger Vehicles/Divergent, reproduced from Butcher [38].

with SLM directly: the injector head of LEAP 71’s XRB-2E6 methalox engine, roughly 600 mm in diameter, was generated autonomously by a computational engineering model and printed as a single Inconel 718 component by Nikon SLM Solutions in under four days [41], while earlier engines from the same programme, printed monolithically in a copper alloy, were hot-fired successfully. This union of automated geometry generation and laser powder bed fusion is a direct, industrial-scale echo of the computational-design-and-SLM approach developed in this thesis. Relativity Space is often cited alongside these as the first company to fly a largely 3D-printed rocket, although its primary structures were built by large-scale wire-arc deposition rather than powder-bed fusion, and it has since returned that structure to conventional methods while concentrating additive manufacturing on the engines [42].

In military aviation the public record is more modest but consistent: laser powder bed fusion is used for engine components, such as parts of the Pratt & Whitney F135 that powers the F-35 [44], and for rapid replacement and sustainment brackets on airframes that are already titanium-intensive, rather than, so far as is disclosed, for topology-optimised primary structure. Taken together, these sectors show that the union of topology opti-

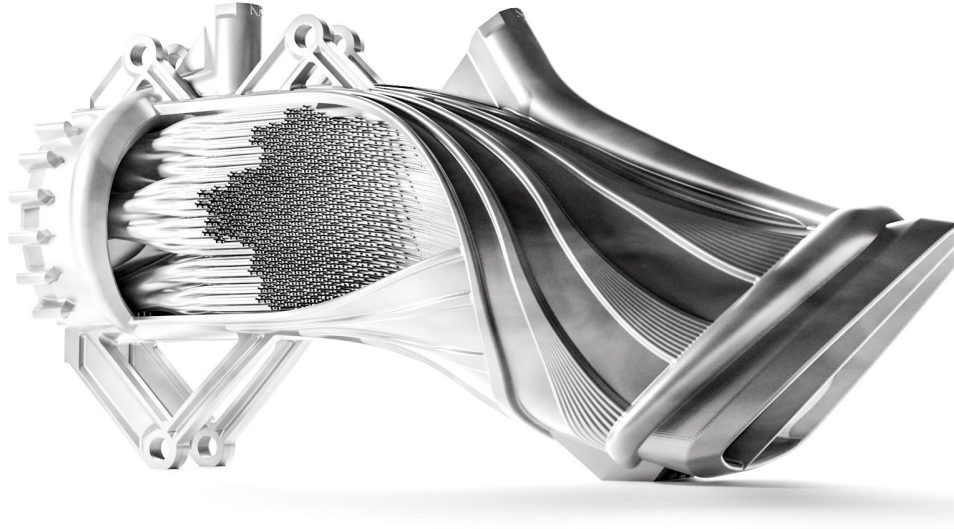


Figure 2.3: A rocket-engine component generated by LEAP 71’s Noyron computational model and produced by metal laser powder bed fusion, shown in cutaway to reveal the injector and the smooth, twisting internal flow passages that a code-driven geometry permits. Parts of this kind are designed algorithmically rather than drawn by hand and printed as near-monolithic bodies, illustrating the union of automated geometry generation and additive manufacturing discussed in the text. Image: LEAP 71, reproduced from VoxelMatters [43].

misation, lattice design and selective laser melting is no longer a laboratory curiosity but an established engineering practice wherever performance justifies its cost, which is the regime in which the present work is set.

2.3 Lattice and Architected Cellular Materials

The geometric freedom of selective laser melting has a structural corollary that is central to lightweight design: it makes *lattice structures* manufacturable at the mesoscopic length scale intermediate between the grain structure of the alloy and the outline of the component. Whereas solid-void topology optimisation decides where material should exist at the scale of the whole component, a lattice distributes a controlled fraction of material according to the localised geometry of a chosen cell; the two ideas are complementary. Additive manufacturing is now the mainstream route to lattice fabrication [45], and over the past decade metallic lattice design has matured from a laboratory curiosity into a recognised structural and lightweighting strategy across the aerospace, automotive, and

biomedical sectors [46], as the applications of Section 2.2.1 attest. It is this maturation that motivated the inclusion of a lattice-generation capability in the present optimiser. The geometric families of unit cell, the distinction between stretching- and bending-dominated behaviour, and the Gibson–Ashby scaling of effective properties with relative density that underlie this field are developed in Section 3.4.

The way in which lattice fields have been coupled to optimisation has evolved considerably, and it is this evolution, rather than the underlying mechanics, that is the concern here. The mathematical coupling between topology optimisation and cellular layouts is today approached through two rigorous methodologies: *direct-infill* optimisation, in which the optimiser is constrained to generate porous, bone-like internal architectures that are robust to localised load variation [47] and, with a dense outer skin, resistant to buckling [48]; and *homogenisation-based multiscale design*, in which each macroscopic point is modelled as an optimal periodic microstructure whose effective properties are computed by numerical homogenisation [8]. The latter is efficient but returns a field of microstructural parameters rather than a printable body; the *de-homogenisation* step that projects that field onto a single, buildable, fully connected lattice [49] was the development that turned lattice-based optimisation from a modelling abstraction into a manufacturable design method, with modern formulations recovering almost the full performance of the idealised continuum while guaranteeing a minimum feature size [50, 51]. These developments form the conceptual basis for the lattice mode of the optimiser used in this work, whose formal apparatus is given in Section 3.4.

2.3.1 Manufacturability and Real-World Performance

The process constraints of selective laser melting reappear, with greater severity, at the length scale of a lattice, where the diameter of a single strut may span only a few melt-pool widths. There is a minimum strut diameter below which a feature cannot be physically resolved, governed jointly by the laser spot size, the melt-pool dimensions, the layer thickness, and the powder particle size. Every strut is itself an overhang, so members inclined below the critical overhang angle of Section 3.2 are inadequately supported; because internal supports cannot be removed from a closed cell array, near-horizontal struts must either be avoided or accepted in a distorted, thermally sagged morphology [52]. Evacuation pathways must also be preserved to allow loose powder to drain from every open cell after the build, which constrains the cell size and internal connectivity.

These manufacturing realities create a systematic gap between the idealised, homogenised properties and those actually measured on printed lattices. The dominant defects, an orientation-dependent deviation of strut diameter, strut waviness, excess material at the nodes, internal porosity, and partially melted powder adhering to the surfaces, mean that the achieved relative density departs from its designed value, the more so for cells of high surface-area-to-volume ratio [52]. Because stiffness and strength scale steeply with relative density, these deviations translate directly into a knockdown of the measured mechanical

properties below prediction. The gap is widest in fatigue: the rough, notched surfaces and the sharp re-entrant geometry of strut nodes act as stress concentrators that nucleate cracks well below the nominal endurance limit [53]. In a comprehensive meta-analysis compiling the mechanical responses of a diverse library of selective-laser-melted cellular structures, Maconachie et al. [45] confirmed that, while the relative-density scaling remains broadly consistent with the Gibson–Ashby model despite considerable scatter, so that the model predicts *trends* reliably, the absolute as-built properties require empirically determined knockdown factors. Surface-based TPMS cells, whose smooth curvature avoids the sharp nodal junctions of strut lattices, tend to print more faithfully and to suffer fewer stress concentrations, which partly explains their growing preference for fatigue-critical and load-bearing parts [52, 54]. The evidence therefore indicates that the performance of a printed lattice falls short of its idealised prediction and must be verified empirically rather than assumed.

A common thread runs through both global solid-void topology optimisation and mesoscopic lattice field design: every candidate design must be evaluated by finite-element analysis, and the computational cost of that evaluation rises sharply as its fidelity increases. Because an optimiser must evaluate a great many such candidates, the total computational cost grows unfavourably with the accuracy one is willing to demand of each evaluation. It is precisely this escalation, common to the solid and lattice routes alike, that motivates the data-driven surrogate models and machine-learning frameworks reviewed in the following Section 2.4.

2.4 Data-Driven Approaches and Machine Learning

The closing observation of the previous section, that the cost of evaluating a candidate design by finite-element analysis grows steeply with fidelity and must be paid at every optimiser iteration, is the practical opening for a different idea: rather than solve every instance from scratch, one may fit a model to a body of already-solved instances and use it to approximate the expensive mapping. This is the premise of *data-driven* design, whose supervised-learning apparatus is stated in Section 3.5. This section reviews how machine learning has been brought to bear on structural optimisation and where the present work sits within that landscape.

The literature, surveyed critically by Woldseth et al. [55] and more broadly by Shin et al. [56], can be grouped into four research strands. The first replaces the optimiser completely with an *iteration-free* predictor: a network maps the problem specifications, namely the design domain, the applied loads and boundary conditions, and the prescribed volume fraction, directly to the optimised topology [57, 58]. The second treats design as a sampling problem and learns the *distribution* of optimal structures with a generative model, whether a variational autoencoder [59], a generative adversarial network [60], or a denoising diffusion model [61]. TopologyGAN conditions a generative adversarial network on physical

fields computed on the unoptimised domain to improve generalisation [62], while TopoDiff applies a conditional diffusion model with a guidance strategy that favours low-compliance, manufacturable structures [2]. The third strand keeps the classical optimiser but replaces its most expensive phase, the finite-element solve, with a learned *surrogate*: networks that predict compliance, stress, or displacement fields, including physics-informed neural networks [63] and neural operators [64]. The fourth reparameterises the design field itself with a network optimised afresh for each problem [65]. Figure 2.4 organises these strands, and Figure 2.5 contrasts the generative variants.

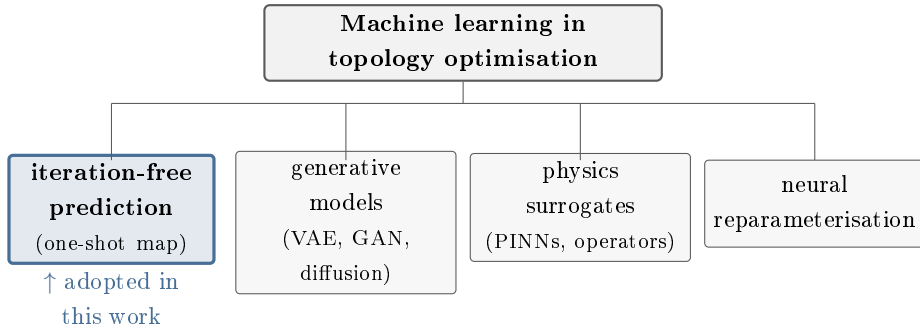


Figure 2.4: A taxonomy of machine-learning approaches to topology optimisation: iteration-free design prediction, generative models of the design distribution, learned surrogates for the physics, and neural reparameterisation of the design field.

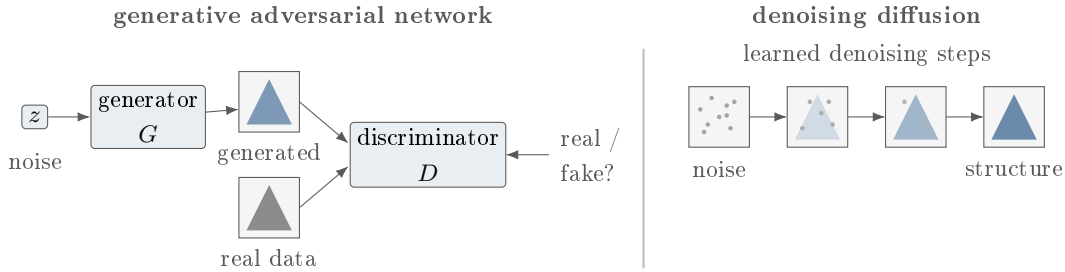


Figure 2.5: Generative routes to topology generation: a generative adversarial network, in which a generator is trained against a discriminator, and a denoising diffusion model, which transforms noise into a structure through a sequence of learned denoising steps. Based on the generative approaches of [1] and [2].

The work in this dissertation belongs to the first strand, using a convolutional encoder-decoder of the U-Net family whose architecture and training loss are set out in Section 3.5. Supervised learning of this kind needs a large number of solved problems, which are themselves costly to produce, so curated open-access datasets have become important research enablers. The datasets released with the TopoDiff framework serve as a representative benchmark: each two-dimensional instance is specified by its boundary conditions, applied loads, and volume fraction, and is paired with the corresponding optimised structure and the physical fields computed on the initial unoptimised domain [2]. A verified resource of this sort makes it possible to train and benchmark a network without first building a

bespoke optimiser to generate the labels, which is what makes it a natural starting point for a first study.

A question of particular interest is whether a network trained on one class of problems acquires representations that generalise to other, distinct structural problems. Deep networks are known to learn generic, reusable spatial features in their earlier layers, with task-specific specialisation concentrated towards the output [66], so a model pretrained on a large, verified dataset can, in principle, be *fine-tuned* to a related task for which little data exist, a process known as *transfer learning*. The idea has been pursued inside topology optimisation: Behzadi and Ilieş [1] coupled conditional generative adversarial networks with transfer learning to adapt a trained model to new loading and boundary-condition scenarios in near-real time. Extending this principle, the hypothesis explored in this dissertation posits that an encoder-decoder trained to reproduce solid-void optima might serve as an effective parametric initialisation for predicting the output of a different optimiser, such as a lattice-based one, since the abstract representations of load paths and stress trajectories learned during solid-void mapping should accelerate adaptation within a lattice-generation framework.

The promise of these deep-learning models is real but qualified, and the critical review of Woldseth et al. [55] is a useful corrective. Three difficulties recur. First, labelled deep-learning models interpolate well within the distribution of their training data but extrapolate poorly when faced with extremal design spaces or novel load configurations outside the training manifold. Second, purely data-driven prediction carries no guarantee of physical validity: it may leave disconnected members, checkerboard patterns, or load paths that cannot carry the load, so an independent finite-element check remains indispensable rather than optional. Third, because training data are computationally expensive, a model only pays for itself once it has been reused enough times to offset the cost of generating those data and training on them, a break-even that many problem-specific networks never reach. Taken together, these caveats position data-driven methods as a complement to verified optimisation rather than a replacement for it, and they are the reason the present work retains explicit finite-element validation throughout.

Within this landscape, the approach adopted here is deliberately modest. A convolutional encoder-decoder is trained, in a supervised and segmentation-like manner, to reproduce the solid-void optima of a verified public dataset. The exercise serves two purposes: it is a controlled introduction to the methods themselves, and it provides a standardised setting in which to test whether a learned representation captures transferable features of structural optimisation. The longer-term aim, to fine-tune that network towards the lattice-based results produced by the framework of this dissertation, is the transfer-learning hypothesis set out above; its implementation and findings are developed in Chapter 6.

2.5 Summary and Positioning

This chapter has reviewed the four research areas on which the remainder of this dissertation builds, and the relations between them. It began with topology optimisation, following the field from its analytical origins to the discrete, bidirectional evolutionary scheme that the present work adopts, the principle running throughout being the redistribution of material within a reference domain to maximise structural performance from a minimum mass budget. Attention then turned to additive manufacturing, and to selective laser melting in particular, as the technology that frees such organic geometries from the constraints of conventional fabrication, together with the range of topology-optimised and lattice-based parts, from titanium implants to hypercar chassis nodes and monolithic rocket-engine injectors, that already justify its cost in industry. The third strand examined the maturation of lattice structures into a recognised mesoscale lightweighting strategy, and concluded that the performance of a printed lattice falls short of its idealised prediction and must be verified empirically. The final strand surveyed data-driven and machine-learning methods, which promise to accelerate the costly optimisation and evaluation steps while carrying their own well-documented caveats, and therefore complement, rather than replace, verified numerical analysis.

Taken together, these threads define both the scope and the posture of this dissertation. The work develops a custom bidirectional evolutionary structural optimisation framework oriented towards selective laser melting, supporting both solid-void and lattice optimisation, and applies it to a structural component of the Furia hypercar developed in partnership with Adamastor. In parallel, it undertakes an exploratory study in which a convolutional encoder-decoder is trained on a verified topology-optimisation dataset, both as an introduction to the methods and as an empirical test of whether the learned spatial features can transfer to the custom evolutionary framework developed herein. A single principle runs through all of it: every design route is held to independent finite-element validation, and the manufacturability limits identified in this chapter are treated as real constraints rather than idealised assumptions. The chapters that follow turn from this contextual literature to the contribution itself, beginning with the theoretical foundations on which the work rests (Chapter 3), and then, in Chapter 4, the development of the optimisation framework at the heart of the dissertation.

Chapter 3

Theoretical Foundations

Chapter 2 reviewed the development of the four fields on which this dissertation draws and the state of their application in engineering practice. This chapter supplies the formal counterpart: the definitions, governing equations, and physical mechanisms that the preceding review invoked but did not state in full. It is organised along the same four strands, so that each formal treatment here can be read against the corresponding section in Chapter 2. Section 3.1 formalises the topology optimisation problem and the evolutionary scheme this work adopts. Section 3.2 sets out the physics of selective laser melting and the manufacturability constraints it imposes. Section 3.4 develops the mechanics of architected cellular solids. Section 3.5 states the supervised-learning apparatus behind the data-driven component of the work. Throughout, the concern is with the canonical formulations of each field; the specific form in which these foundations are implemented in the optimiser developed here is deferred to Chapter 4.

3.1 Topology Optimisation

Structural optimisation is commonly divided into three categories of increasing design freedom: sizing, in which scalar design parameters are tuned on a fixed layout; shape, in which the boundaries and contours of a fixed topology are moved; and topology, in which the connectivity of the structure itself, the number and arrangement of its members and holes, is treated as the unknown. Topology optimisation is therefore the most general of the three, since it does not presume a starting configuration. In the continuum setting the problem is posed as a material distribution problem: a fixed design domain is discretised by finite elements, and the task is to decide, for every point of that domain, whether material should be present or absent so that a performance measure is extremised subject to a constraint on the amount of material used.

Switching material on or off discretely renders the problem ill-posed, and the foundational remedy, introduced with the homogenisation method of Bendsøe and Kikuchi [8], is to relax it by admitting a composite microstructure whose homogenised properties vary continuously between solid and void. To obtain clearer solid-void layouts, the

density approach of Bendsøe [9] and Zhou and Rozvany [10] discretises the domain into finite elements and assigns each element e a single scalar design variable, a pseudo-density $x_e \in [0, 1]$. Assuming an isotropic base material, the material properties are parameterised by interpolating the effective Young's modulus E_e as a power law of that density,

$$E_e(x_e) = x_e^p E_0, \quad (3.1)$$

where E_0 is the modulus of the fully solid material and p is a penalisation exponent. To satisfy the physical Hashin-Shtrikman bounds for a two-phase mixture where one phase is void, a penalisation of $p \geq 3$ is mathematically required for a material with a Poisson's ratio of $1/3$. Because intermediate densities are penalised, they become structurally inefficient and the optimiser is driven towards a near black-and-white design. This scheme is the Solid Isotropic Material with Penalisation (SIMP) model, named by Rozvany et al. [11] and consolidated in its material-interpolation foundations by Bendsøe and Sigmund [12].

The canonical statement of the minimum-compliance problem under the SIMP framework reads

$$\min_{\mathbf{x}} c(\mathbf{x}) = \mathbf{F}^\top \mathbf{U} \quad \text{s.t.} \quad \mathbf{K}(\mathbf{x}) \mathbf{U} = \mathbf{F}, \quad \frac{\sum_e x_e v_e}{V_0} \leq f, \quad x_{\min} \leq x_e \leq 1, \quad (3.2)$$

where c represents the structural compliance, $\mathbf{F}$ is the external load vector, $\mathbf{U}$ is the global displacement vector, and $\mathbf{K}(\mathbf{x})$ is the global stiffness matrix, explicitly parameterised by the vector of elemental design densities $\mathbf{x}$. The scalar parameter v_e denotes individual element volumes, V_0 is the total volume of the design domain, and f is the prescribed maximum allowable volume fraction. The lower bound $x_{\min}$ (typically 10^{-3}) is a small positive value enforced strictly to prevent the global stiffness matrix from becoming singular and computationally unsolvable when elements empty out.

Density methods are not without numerical difficulties. As surveyed by Sigmund and Petersson [14], they are prone to checkerboard patterns of alternating solid and void elements, to mesh dependence in which finer meshes yield qualitatively different designs, and to local minima. The accepted remedy is to regularise the problem with a filter that couples each element to its neighbours over a prescribed radius, which both suppresses checkerboarding and imposes a minimum length scale on the solution. Filtering is now considered an essential ingredient of any robust implementation.

The scheme adopted in this work belongs to the evolutionary family. Evolutionary Structural Optimisation (ESO), introduced by Xie and Steven [17] and consolidated in Xie and Steven [18], quantifies the effectiveness of each element, for global compliance minimisation, by a discrete sensitivity number equal to the elemental strain energy,

$$\alpha_e = \frac{1}{2} \mathbf{u}_e^\top \mathbf{k}_e \mathbf{u}_e, \quad (3.3)$$

where $\mathbf{u}_e$ and $\mathbf{k}_e$ are the element displacement vector and stiffness matrix, respectively.

Elements with the lowest sensitivity are the least effectively used. The bidirectional extension, Bidirectional Evolutionary Structural Optimisation (BESO) [20], allows material to be added back where the sensitivity numbers indicate it is needed as well as removed where it is not. The convergent and mesh-independent form of BESO due to Huang and Xie [21], and given comprehensive treatment in Huang and Xie [22], rests on three ingredients. First, the elemental sensitivity numbers are filtered over a fixed radius, in direct analogy with density filtering. Second, the optimisation history is stabilised by averaging each sensitivity number with its value from the previous iteration. Third, the scheme is soft-kill: void elements are not purged from the domain but retain a small residual stiffness factor (for example $x_{\min}$), which avoids layout singularities in the global finite-element mesh and keeps the sensitivities of such elements computationally meaningful. It is this convergent, mesh-independent, soft-kill BESO that provides the algorithmic foundation for the optimiser of this work; the particular update rule, filter kernels, and history stabilisation used in the implementation are set out in Chapter 4.

3.1.1 Rank-Based Selection

BESO is a comparative algorithm: it uses the sensitivity numbers not as physical quantities to be reproduced but only as the means by which the elements are ordered. At each iteration the elements are sorted by their sensitivity number α_e of Equation (3.3) and a prescribed fraction is removed from the lower end of that order and, in the bidirectional case, restored at the upper end. The design the scheme produces therefore depends on the sensitivity field $\{\alpha_e\}$ only through the ordering it induces on the elements, and not through the magnitudes themselves. A direct consequence is that the scheme is invariant under any transformation that preserves the order of the sensitivities. If every sensitivity α_e is replaced by $g(\alpha_e)$, where g is any strictly increasing function, the elements keep their relative order, and with it every removal and addition decision is unchanged. Uniform rescaling of the whole field, such as that produced by a change in the elastic modulus or in the magnitude of the applied load, is the simplest such transformation, and under it the selection is exactly preserved.

This exact invariance holds only for transformations that preserve the order everywhere. A change to the model that perturbs the sensitivity field non-uniformly, altering the ordering of some elements rather than merely rescaling the field, carries no such guarantee, and whether it leaves the selection intact becomes a question to be answered rather than assumed. Coarsening the mesh of a body that supports the design domain is a perturbation of exactly this kind, neither uniform nor necessarily monotone in its effect on the field, so its consequence for the ordering must be measured. The measures by which two orderings are compared are set out next.

Two rank correlation coefficients quantify how closely one ordering of a set of n elements agrees with another. Spearman's coefficient ρ [67] is the Pearson correlation of the rank

variables; when no ranks are tied it reduces to

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}, \quad (3.4)$$

where d_i is the difference between the rank of element i in the two orderings. Kendall's coefficient τ [68] instead counts pairs of elements: writing n_c for the number of pairs placed in the same order by both rankings (concordant) and n_d for the number placed in opposite orders (discordant),

$$\tau = \frac{n_c - n_d}{\frac{1}{2} n(n-1)}. \quad (3.5)$$

Both coefficients equal +1 for identical orderings, -1 for exactly reversed ones, and zero for unrelated ones, with τ carrying the more direct reading as the net fraction of element pairs placed in the same order.

The two coefficients weigh every element equally, whereas the evolutionary update acts only on the extremes of the ordering, removing the lowest-ranked elements and retaining the highest-ranked ones. A measure aligned with that action compares the sets the two orderings place at one extreme, rather than the orderings as a whole. For a chosen extreme, let S_k and $\hat{S}_k$ be the k elements placed there by the first and the second ranking; the set agreement is the fraction they share,

$$A_k = \frac{|S_k \cap \hat{S}_k|}{k}. \quad (3.6)$$

Taking k as the fraction the update moves in a single iteration makes A_k the agreement of the elements the scheme would actually act on, which is the measure most directly tied to the update rule.

Where the perturbation to the model is confined to a small region, a classical result of elasticity limits how far through the domain its effect on the ordering can reach. Saint-Venant's principle states that if the tractions acting over a small region of an elastic body are replaced by a statically equivalent distribution, one with the same resultant force and moment, the two loadings produce appreciably different stresses only close to that region, within a distance comparable to its size, and give essentially the same stresses everywhere further away [69].

3.2 Selective Laser Melting

The term additive manufacturing is formally defined by the international vocabulary standard ISO/ASTM 52900 [70] as the process of joining material to make parts from three-dimensional model data, usually layer upon layer, as opposed to subtractive and formative manufacturing methodologies. The same standard partitions the field into seven process categories (Figure 3.1): binder jetting, directed energy deposition, material extrusion, material jetting, powder bed fusion, sheet lamination and vat photopolymerisation. Of these,

powder bed fusion is the category most widely used for load-bearing metallic components, and within it the dominant variant for the present work is laser-based powder bed fusion of metals, commonly termed Selective Laser Melting (SLM) or, in the standardised nomenclature, laser powder bed fusion (LPBF). The two names are used interchangeably in the literature and refer to the same physical process.

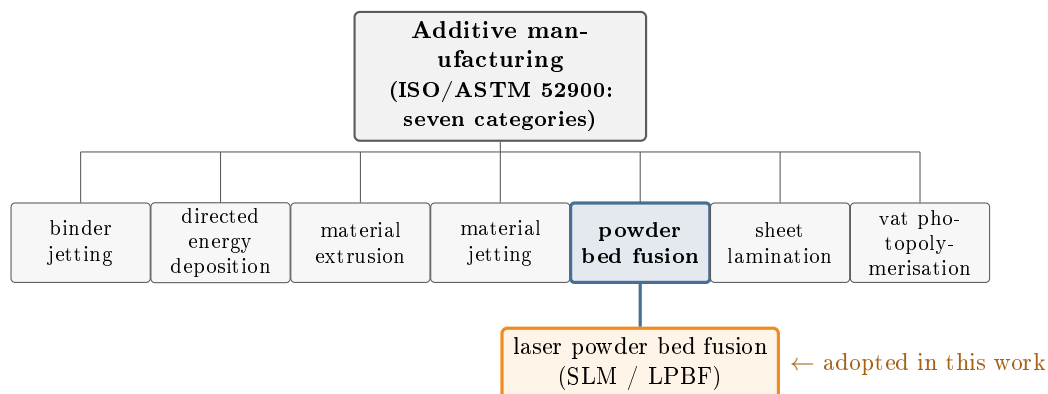


Figure 3.1: The seven process categories of additive manufacturing defined by ISO/ASTM 52900, with powder bed fusion and its laser-based variant (the route adopted in this work) highlighted.

In SLM a thin layer of metallic powder, typically between twenty and sixty micrometres thick, is spread across a build platform by a recoater blade. This layer thickness cannot be reduced below the scale of the powder itself: it must exceed the coarsest particles to spread evenly, so the powder particle size distribution, commonly in the range of fifteen to forty-five micrometres for SLM, sets a practical lower bound on it [71]. A focused laser beam then scans the cross-section of the part for that layer, locally melting the powder and the substrate beneath it into a small, transient pool of liquid metal. Once the layer is fused the platform descends by one layer thickness, fresh powder is spread, and the cycle repeats until the part is complete. The entire process is conducted under an inert atmosphere, usually argon or nitrogen, to limit oxidation of the highly reactive molten metal, and is shown schematically in Figure 3.2.

The physics that governs part quality is concentrated in the melt pool. The laser delivers energy to a region only a few hundred micrometres across, producing extremely steep thermal gradients and cooling rates that can reach 10^5 to 10^7 K/s. The molten metal is driven by Marangoni convection arising from surface-tension gradients across the pool [72], and the balance between the energy absorbed and the energy conducted away into the surrounding solid determines the pool's size, depth and stability. When the local energy input is too low, the powder is incompletely melted and the pool fails to penetrate and fuse with the layer beneath, leaving irregular lack-of-fusion voids. When it is too high, the pool deepens through vaporisation into a narrow, unstable cavity whose periodic collapse traps spherical gas pores, a regime known as keyholing. A dense, defect-free part therefore requires the energy input to be held within an intermediate processing window.

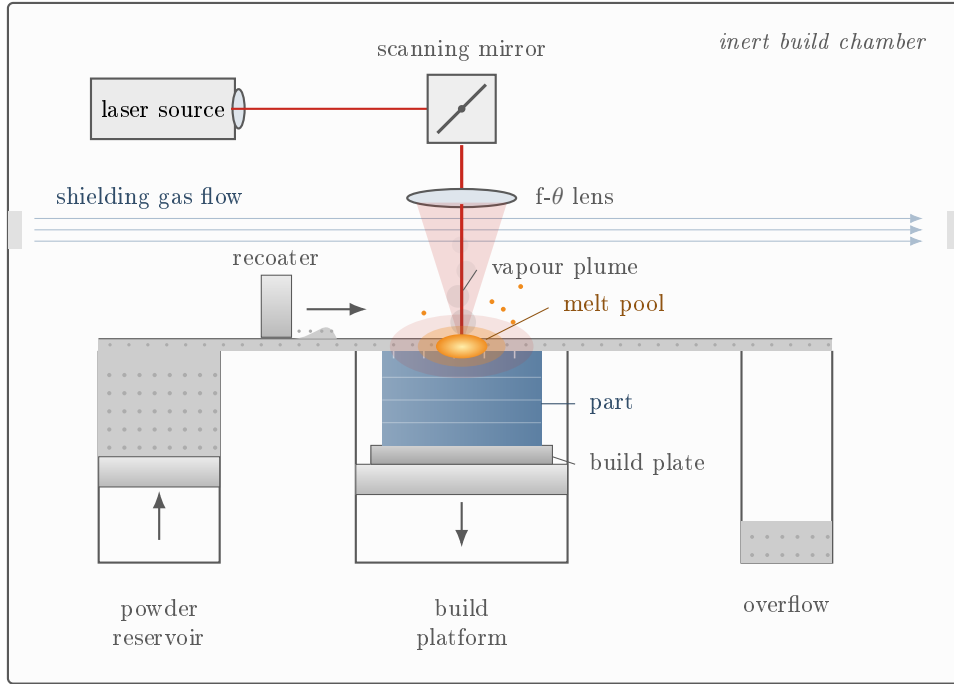


Figure 3.2: The Selective Laser Melting process: a scanned, focused laser beam fuses a thin powder layer into a melt pool on the part, while shielding gas sweeps away the vapour plume and spatter; the build platform descends and the powder reservoir rises by one layer thickness per cycle.

The principal process parameters that set this energy input are conventionally combined into a single volumetric energy density,

$$E_v = \frac{P}{v h t}, \quad (3.7)$$

where P is the laser power (W), v is the laser scan speed (mm/s), h is the hatch spacing between adjacent scan tracks (mm), and t represents the layer thickness (mm). Equation (3.7) is a convenient first-order indicator of whether a parameter set lies in the lack-of-fusion, dense, or keyholing regime as E_v varies (Figure 3.3), and it is widely used to transfer parameters between geometries and machines. Its limitations should be stated plainly, however: because it lumps four parameters into one scalar, very different combinations of power and speed can share the same E_v yet produce markedly different melt-pool depths and defect populations. The volumetric energy density is thus a useful guide to the processing window rather than a complete predictor of it, and the underlying behaviour of the melt-pool dynamics remains the governing physics [72].

The same rapid, localised heating that fuses the powder also leaves the finished part in a state that differs markedly from a conventionally processed metal, and these differences follow directly from the thermal history rather than being incidental. The most consequential is residual stress. Its origin is captured by the temperature gradient mechanism proposed by Mercelis and Kruth [3]: as the laser heats a region, the material there tries to

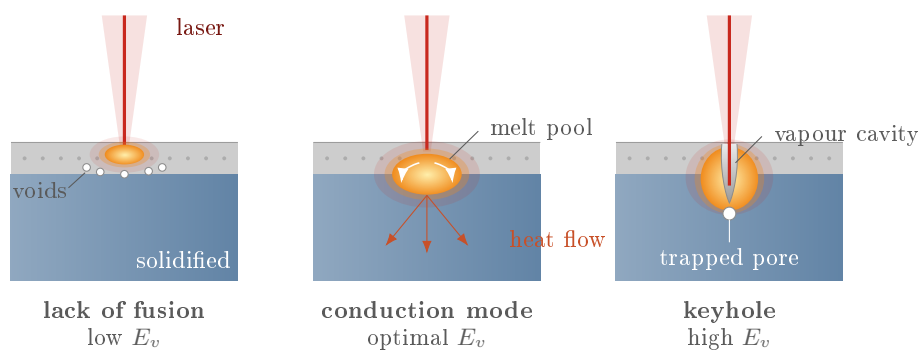


Figure 3.3: Melt-pool behaviour under the laser. At low volumetric energy density the pool is small and fails to fuse, leaving lack-of-fusion voids; at optimal energy density a stable conduction-mode pool fuses into the substrate with Marangoni recirculation and steep heat flow; at high energy density, the metal boils instead of melting, opening deep keyholes, whose walls can become unstable and collapse inward, trapping porosity.

expand but is restrained by the cooler, stiffer solid around it, so it yields in compression while hot; on cooling it then contracts and is again restrained, leaving that region in tension and balancing compression in the bulk, as illustrated in Figure 3.4. With each deposited layer reheating and partially remelting those beneath it, this cycle accumulates, producing large tensile stresses near the surfaces and along the build direction. The practical consequences are part distortion, a tendency to crack or delaminate from the build plate, and a reduction in fatigue life, which is why stress relief is a near-universal post-processing step for SLM parts [73].

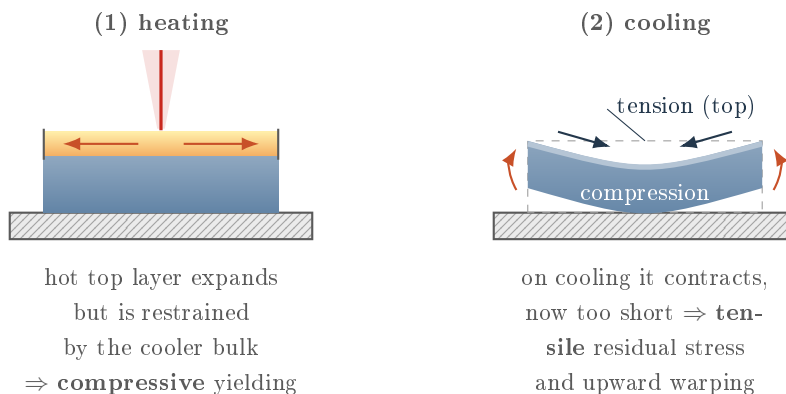


Figure 3.4: The temperature gradient mechanism of residual-stress formation in SLM, after Mercelis and Kruth [3]: the restrained expansion of the hot top layer yields in compression, and on cooling its contraction leaves tensile residual stress at the surface and an upward distortion.

The thermal history also imprints a characteristic microstructure. Because heat is extracted predominantly downward into the previously solidified material and the build plate, grains tend to grow epitaxially along the build direction, frequently as coarse columnar grains spanning several melt-pool layers. This directional texture makes the as-built

mechanical properties anisotropic, with strength and especially ductility differing between the build direction and the plane of each layer. A further geometric consequence of the layerwise nature of the process is the staircase effect, in which curved or inclined surfaces are approximated by discrete layer steps, degrading surface finish and dimensional accuracy in a manner that worsens with layer thickness and with the shallowness of the surface's inclination (Figure 3.5). Porosity from the lack-of-fusion and keyholing regimes described above completes the picture of defect types that an SLM part designer must contend with. Many of these effects, residual stress and anisotropy in particular, can be reduced by post-process heat treatment and hot isostatic pressing, at additional cost [74].

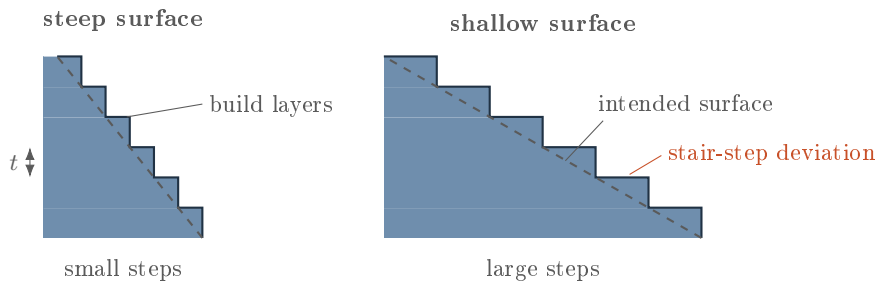


Figure 3.5: The staircase effect, in which an inclined surface is approximated by discrete build layers of thickness t ; the deviation from the intended surface grows as the inclination becomes shallower.

The freedoms of SLM are thus accompanied by constraints, and designing for the process means working within them. Chief among the geometric constraints is the overhang problem. Each new layer must rest, at least partially, on solid material below it; a downward-facing surface inclined below a critical angle to the build plate, conventionally taken as around forty-five degrees for metal powder bed fusion, cannot support itself and will either sag into the loose powder or fail outright (Figure 3.6). Thomas [75] established such critical-angle design rules empirically for SLM, and they remain the simplest manufacturability guideline in use. This forty-five-degree value is a rule of thumb rather than a fixed property of the process: the true threshold depends on the layer thickness, laser spot size, energy density, and material, and with carefully tuned down-skin parameters surfaces well below it, in the limit even horizontal overhangs, have been built support-free [76]. It is nonetheless retained here as a conservative design guideline. Overhanging regions that fall below the critical angle must instead be propped by sacrificial support structures, which anchor the overhang and conduct heat away from it (Figure 3.7). Supports, however, carry real penalties: they consume additional material and build time, they must be removed in post-processing, which is laborious and can be impossible inside enclosed cavities, and their removal leaves witness marks that degrade the local surface. The strategies that have been proposed to relieve the designer of this burden, and the physics-based route adopted in this work, are reviewed in the state of the art (Section 2.2) and formulated in Chapter 4.

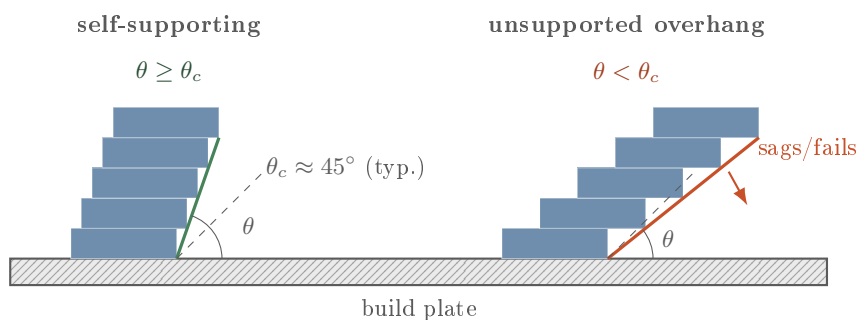


Figure 3.6: Definition of the critical overhang angle θ_c relative to the build plate. Downward-facing surfaces inclined at or above $\theta_c \approx 45^\circ$ are self-supporting, whereas shallower surfaces are unsupported and tend to sag or fail. The value $\theta_c \approx 45^\circ$ is a conventional rule of thumb; the true threshold varies with layer thickness, spot size, energy density, and material.

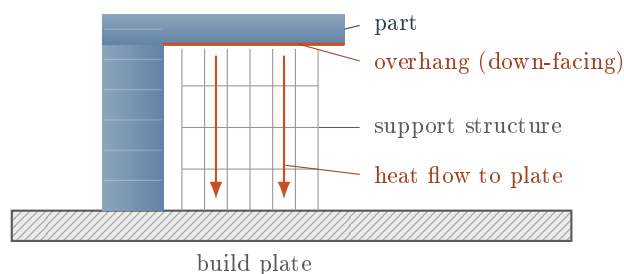


Figure 3.7: Sacrificial support structures anchoring a downward-facing overhang to the build plate and providing a thermal path that conducts heat away from the overhanging region.

3.2.1 Metallic Materials for Selective Laser Melting

The set of metals that can be processed reliably by SLM is far narrower than the catalogue of engineering alloys, because the process effectively subjects the material to repeated, rapid welding, and only alloys that tolerate this without cracking are suitable. Of the several thousand alloys in industrial use, only a handful are routinely printable, the most established being the aluminium alloy AlSi10Mg, the titanium alloy Ti-6Al-4V, cobalt-chromium, and the nickel-based superalloy Inconel 718. Alloys outside this group frequently develop intolerable columnar-grain structures and solidification cracks, so printability is in practice limited to compositions that are known to be readily weldable [74].

Within the printable set, the choice is governed by the specific application. Aluminium alloys, principally AlSi10Mg, are the workhorse for lightweight structural parts: the near-eutectic silicon content gives good castability-like behaviour and resistance to solidification cracking, magnesium enables precipitation strengthening through Mg_2Si , and the low density suits automotive and aerospace components where mass minimisation is the primary concern. Their absolute strength is modest, but their specific strength after heat treatment is competitive, and among printable metals they are comparatively inexpensive [74].

Titanium alloys, chiefly Ti-6Al-4V, offer a far higher specific strength together with excellent corrosion resistance and biocompatibility [77], at a substantially higher material cost and with more demanding processing. Nickel-based superalloys such as Inconel 718 and Inconel 625 retain strength and oxidation resistance at high temperature and are the most prone to residual-stress problems [73, 77]. Stainless and maraging steels, including 316L, 17-4PH and the maraging grades, occupy the lower-cost end of the spectrum, while cobalt-chromium serves dental and orthopaedic applications [77]. The microstructure and mechanical response of the two lightweight workhorses, AlSi10Mg and Ti-6Al-4V, are strongly dependent on the post-build heat treatment applied [40].

A recurring theme across all of these families is the gap between the as-built and the serviceable state. The as-built microstructure carries the residual stress, anisotropy and porosity described earlier, so most structural applications require post-processing, typically stress relief, a solution and ageing heat treatment, and in demanding cases hot isostatic pressing to close internal pores. These steps recover ductility and reduce anisotropy but add processing cost and lead time, and they must be counted as part of the true cost of an SLM part rather than as an afterthought.

Cost, indeed, is the constraint that ultimately bounds the adoption of these technologies. Metal powder feedstock is itself expensive, with broad market prices ranging from the order of tens of euros per kilogramme for aluminium and stainless steels to several times that for titanium and nickel superalloys [78], and the powder must meet stringent sphericity and size-distribution requirements that further raise its price. To the feedstock cost must be added the slow build rate of laser powder bed fusion, the energy and inert gas consumed, the machine's high capital cost amortised over a low throughput, and the extensive post-processing already noted [78]. The result is a high per-part cost that confines metal SLM to applications where its distinctive advantages, geometric complexity at no additional tooling cost, part consolidation, lightweighting and short development cycles, repay the premium. This is precisely the regime of high-performance automotive and aerospace components in which the present work is set, and the range of parts that already justify the premium in industry is surveyed in Section 2.2.

3.3 Fatigue of Metallic Materials

The verification of a load-bearing component against its peak static load establishes that it will not fail on first application, but it does not establish that it will endure. A component may sustain a load many times without incident and yet fail after a large number of repetitions of a load well below the one that would break it in a single application, because each cycle leaves behind a small increment of irreversible damage that accumulates until a crack forms and grows to a critical size. This is the phenomenon of fatigue, and it accounts for the majority of in-service failures of metallic structures [79]. Failure proceeds in two stages: the initiation of a crack at a site of local stress concentration, such as a surface

irregularity, an inclusion or a pore, followed by its stable propagation cycle by cycle until the remaining section can no longer carry the load and separates. Because a suspension component is loaded and unloaded continually as the vehicle travels, a static check alone is insufficient, and a fatigue assessment is required to confirm that its life exceeds its intended service.

The treatment of fatigue adopted in this work is the stress-life, or S-N, approach, appropriate to the high-cycle regime in which the stresses remain nominally elastic and the lives are long. Its central object is the S-N curve, the experimentally determined relationship between the amplitude of a fully reversed stress cycle and the number of cycles the material sustains before failure. Over the high-cycle range this relationship is well described by the power law of Basquin [80],

$$\sigma_{\text{ar}} = \sigma'_f N_f^\beta, \quad (3.8)$$

in which σ_{ar} is the fully reversed stress amplitude, N_f the number of cycles to failure, and σ'_f and β are material constants fitted to the test data, β being the negative fatigue-strength exponent. The law is written here in terms of the number of cycles and a single strength coefficient, rather than in the equivalent classical form expressed in reversals, so as to match the parameterisation of the experimental characterisation drawn upon in Chapter 5. Ferrous alloys typically display a fatigue limit, an amplitude below which the life becomes effectively unbounded, whereas aluminium alloys do not, so their S-N line continues to descend and any assessment that extrapolates it below the tested range does so conservatively.

The S-N curve is measured under fully reversed loading, that is, at a stress ratio $R = -1$ for which the mean stress is zero, whereas a service cycle generally carries a non-zero mean. A tensile mean stress is damaging, lowering the amplitude the material can sustain for a given life, and its effect must be accounted for by correcting each cycle to the equivalent fully reversed amplitude that would produce the same life. This correction relates the amplitude σ_a and mean σ_m of the cycle through a constant-life construction, of which the two used in this work are those of Goodman [81] and Soderberg [82],

$$\sigma_{\text{ar}} = \frac{\sigma_a}{1 - \sigma_m/\sigma_u} \quad (\text{Goodman}), \quad \sigma_{\text{ar}} = \frac{\sigma_a}{1 - \sigma_m/\sigma_y} \quad (\text{Soderberg}), \quad (3.9)$$

where σ_u is the ultimate tensile strength and σ_y the yield strength. The two constructions differ only in the strength to which the mean is referred, Goodman to the ultimate and Soderberg to the lower yield, so that Soderberg penalises the mean more heavily and is the more conservative. Once corrected to σ_{ar} , a cycle can be placed on the fully reversed S-N curve to obtain its life.

A service load history is rarely a sequence of identical cycles but an irregular succession of varying amplitude and mean, and two further steps reduce such a history to a life. The first decomposes it into a set of closed cycles, each with its own amplitude and mean, which

is the purpose of the rainflow counting method [83, 84], standardised in ASTM E1049 [85]; the method extracts from the irregular signal the closed hysteresis loops the material actually experiences. The second combines the damage of these cycles into a single measure through the linear damage rule of Palmgren [86] and Miner [87], which treats the damage of a cycle as the reciprocal of the life it would have in isolation and sums these contributions,

$$D = \sum_i \frac{n_i}{N_{f,i}}, \quad (3.10)$$

where n_i is the number of cycles counted at the i th amplitude and $N_{f,i}$ the corresponding life read from the corrected S-N curve. Failure is predicted when the accumulated damage D reaches unity. The rule disregards the order in which the cycles are applied, which is its principal limitation, but its simplicity and its adequacy for high-cycle assessment have made it the standard basis for cumulative fatigue calculation.

3.3.1 Fatigue of Selectively Laser-Melted AlSi10Mg

The fatigue behaviour of a selectively laser-melted part is governed far more strongly by the defects the process leaves behind than by the intrinsic properties of the alloy, and this follows directly from the process physics of Section 3.2. The pores and lack-of-fusion voids that survive in the as-built material, and above all its rough as-built surface, act as ready crack-initiation sites, so that cracks start earlier and lives are both shorter and more scattered than in wrought material of the same composition [88]. In this respect the printed alloy behaves in a manner closer to a casting than to a wrought product, its fatigue strength being set by the size and location of the largest defect rather than by a smooth-specimen property of the material. The surface condition is the most influential of these features: a machined or polished surface carries a markedly higher fatigue strength than the same material left as-built, so the finish of a fatigue-critical region is a first-order design decision rather than a detail [89].

These characteristics shape how the fatigue of such a part must be assessed. The heat treatment applied to relieve residual stress and recover ductility, and where required the hot isostatic pressing used to close internal porosity, both raise the fatigue strength, so the S-N curve used must correspond to the actual processed condition of the part rather than to a generic value [40]. Because the governing defects are statistical and their severity depends on process control, a fatigue safety factor larger than that adopted for the static check is warranted, in the spirit of the partial-factor treatment that guidelines such as the FKM apply to defect-prone, casting-like materials and to components whose failure carries severe consequences [90]. And because the surface finish is decisive, the critical regions of a fatigue-loaded part are machined to a controlled finish and inspected, so that the assessment at those locations may properly rest on a machined-surface curve. The application of these principles to the optimised bracket, using the experimental characterisation of laser powder-bed AlSi10Mg reported by Sajadi et al. [91], is carried out in Chapter 5.

3.4 Lattice and Architected Cellular Materials

A lattice is a solid whose interior is not fully dense but is instead occupied by a periodic, or near-periodic, repetition of an open unit cell, where material is placed at a mesoscopic length scale intermediate between the microscopic grain structure of the alloy and the macroscopic outline of the component. Whereas the density-based and evolutionary frameworks of Section 3.1 decide where solid material should exist at the scale of the whole component, a lattice distributes a controlled fraction of material according to the localised geometry of a chosen cell.

Lattices are conventionally divided into two families according to how the unit cell is described geometrically. *Strut-based*, or truss, lattices are assemblies of slender beams that meet at nodes, with body-centred cubic (BCC), face-centred cubic (FCC), and octet configurations serving as representative topologies; they are intuitive to parameterise and their mechanical behaviour can be approximated via classical beam theory. *Surface-based* lattices instead define the solid domain as the region bounded by smooth implicit functions, most commonly utilising Triply Periodic Minimal Surfaces (TPMS) such as the gyroid, the Schwarz primitive, or the diamond. These base surfaces partition space into interpenetrating sub-volumes and have, by definition, zero mean curvature everywhere [54]. TPMS cells exist both in *sheet* (matrix-phase) form, in which the solid material forms a thin shell straddling the surface, and in *network* (skeletal) form, where the solid fills one of the two sub-volumes. The absence of sharp junctions gives surface-based cells a highly homogeneous internal stress field, avoiding the severe stress concentrations associated with the re-entrant nodes of strut-based lattices, a distinction that becomes critical when manufacturability and fatigue are considered. Figure 3.8 summarises the two families. Prismatic honeycombs and stochastic foams are the older relatives of these architected lattices, and much of the mechanical framework used to describe lattices was first developed for them.

The mechanical efficiency of a lattice is set less by the volume of material it contains than by the kinematic *mode* in which its members sustain load. A unit cell whose struts deform predominantly by axial stretching is far stiffer and stronger, per unit mass, than one whose struts must bend, because a slender beam is inherently orders of magnitude more compliant in bending than in tension. Which deformation regime prevails is a topological property of the cell, determined by Maxwell’s criterion for the static determinacy of a pin-jointed frame. For a three-dimensional structural frame, this condition is expressed as

$$M = b - 3n + 6, \quad (3.11)$$

where b and n are, respectively, the numbers of struts and frictionless nodes in the cell. When $M < 0$ the frame constitutes a kinematic mechanism, able to deform without stretching its members, and the lattice is therefore *bending-dominated*. When $M \geq 0$ the frame is rigid and carries load by axial stretching or compression, so the lattice is *stretching-dominated*; it is statically determinate when $M = 0$ and statically indeterminate, with

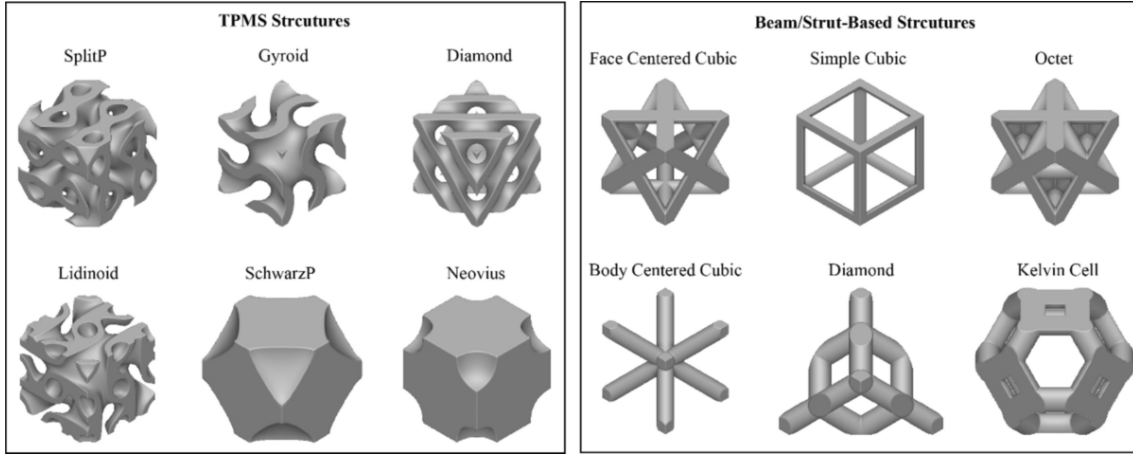


Figure 3.8: The two principal families of architected lattice unit cell: strut-based (truss) cells assembled from slender beams, such as the body-centred cubic, face-centred cubic, and octet, and surface-based cells defined by a triply periodic minimal surface (TPMS), such as the gyroid, the Schwarz primitive, and the diamond. Reproduced from Amirpour et al. [92].

redundant members, when $M > 0$ [93]. Deshpande et al. [93] further demonstrated that, when extended to infinite periodic arrays, an average nodal coordination number (connectivity) of twelve is sufficient for stretch-domination, a condition met by the octet-truss, whose behaviour Deshpande et al. [94] subsequently confirmed analytically, numerically, and experimentally. Figure 3.9 contrasts the two regimes. The practical implication is a trade-off: stretching-dominated cells maximise specific stiffness and strength and are therefore the natural choice for load-bearing structure, but they yield abruptly once their struts buckle or yield, whereas bending-dominated cells are more compliant yet collapse progressively, which is why the latter prevail in cushioning and impact applications.

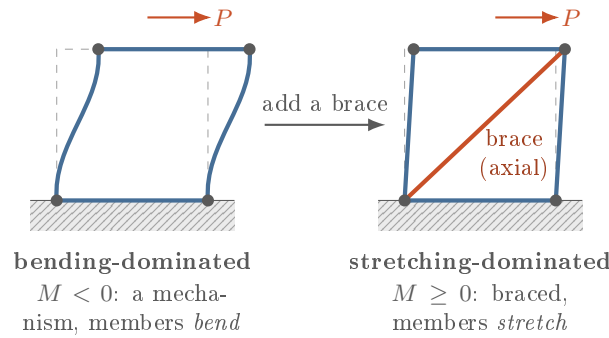


Figure 3.9: Bending-dominated versus stretching-dominated unit cells. A cell that forms a mechanism under Maxwell's criterion, Equation (3.11), deforms by strut bending and is compliant, whereas a statically determinate cell carries load by axial stretching and is far stiffer for the same mass.

The single most useful descriptor of a lattice is its *relative density* $\bar{\rho} = \rho^*/\rho_s$, the ratio of the effective density of the cellular solid to that of its parent material, which is equal

to the volume fraction of solid within the cell. Gibson and Ashby established that the effective properties of a cellular solid scale with relative density through power laws [95], whose exponents are set by the deformation mode of the cell.

$$\frac{E^*}{E_s} = C_1 \bar{\rho}^{n_E}, \quad \frac{\sigma^*}{\sigma_{ys}} = C_2 \bar{\rho}^{n_\sigma}, \quad (3.12)$$

For idealised open-cells, bending-dominated topologies present a stiffness exponent $n_E \approx 2$ and a strength exponent $n_\sigma \approx 1.5$, whereas for a stretching-dominated lattice both tend towards $n_E \approx n_\sigma \approx 1$. The consequence is decisive for lightweight design: at equal relative density, say $\bar{\rho} = 0.1$, reducing the stiffness exponent from two to one raises the relative stiffness by a full order of magnitude. Figure 3.10 plots these laws on logarithmic axes, where the deformation mode appears directly as the slope. A lattice engine within a structural optimiser exploits exactly this relationship, treating relative density, and hence local stiffness, as a field to be distributed across the design domain.

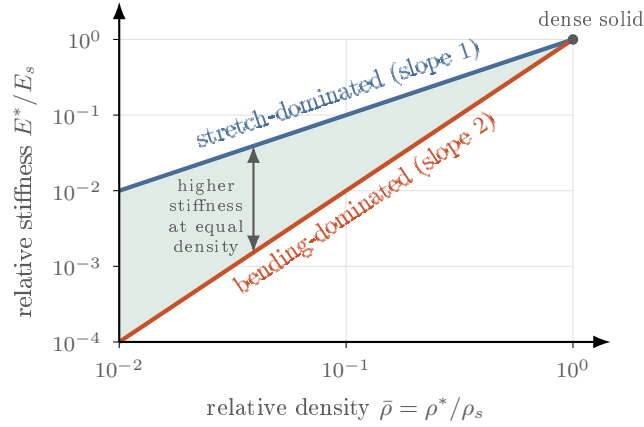


Figure 3.10: Gibson–Ashby scaling of relative stiffness with relative density on logarithmic axes. Stretching-dominated lattices follow a slope of approximately one and bending-dominated foams a slope of approximately two, so the deformation mode, and not merely the amount of material, governs efficiency.

The way in which this field is materialised admits several strategies. The earliest designs were *uniform* lattices, where a single unit cell is tiled at constant relative density throughout a design space; these are simple to generate but inefficient, because they ignore the spatially varying stress state of a loaded part. *Functionally graded* lattices address this by varying the relative density, the cell size, or the cell topology across the domain to match local stress requirements, and *conformal* lattices additionally warp the cellular grid layout so that it aligns with the part boundary or with the principal-stress trajectories [46]. These strategies are illustrated in Figure 3.11.

The mathematical coupling between topology optimisation and cellular layouts is approached through two rigorous methodologies. In *direct-infill* optimisation, the optimiser is explicitly constrained to generate porous topologies by enforcing upper bounds on material

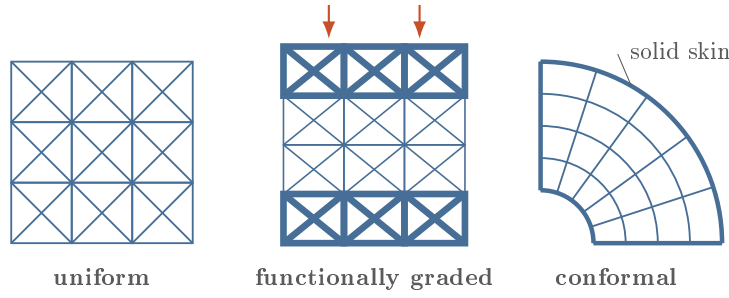


Figure 3.11: Lattice design strategies: a uniform lattice at constant relative density, a functionally graded lattice whose density follows the local stress, and a conformal lattice aligned to the part boundary, optionally enclosed by a solid skin.

fractions, yielding optimised, bone-like internal architectures that exhibit high structural robustness under localised load variations [47], and, when integrated with a fully dense outer skin, markedly enhanced resistance to buckling modes [48]. In *homogenisation-based multiscale design*, each macroscopic point within the reference domain is modelled as an optimal, continuously variable periodic microstructure whose effective properties are computed via numerical homogenisation, the very approach that founded modern topology optimisation [8]. While homogenisation is computationally efficient, as the optimisation runs on a coarse mesh, its output is a field of microstructural parameters rather than a printable body. The missing step, called *de-homogenisation*, projects that field onto a single, high-resolution, fully connected lattice that can actually be built [49]; modern formulations recover almost the full performance of the idealised multiscale continuum while guaranteeing a minimum manufacturable feature size [50, 51]. These concepts form the basis for the lattice mode of the optimiser used in this work, whose place in the development of the field is discussed in Section 2.3 and whose implementation is developed in Chapter 4.

3.5 Data-Driven Modelling and Neural Networks

The cost of evaluating a candidate design by finite-element analysis grows steeply with fidelity and must be paid at every iteration of an optimiser. This is the practical opening for a different idea: rather than solve every instance from scratch, one may fit a model to a body of already-solved instances and use it to approximate the expensive mapping. This is the premise of *data-driven* design. In the supervised learning format most relevant to this work, a model f_θ with parameters (weights) θ is trained on a dataset of N solved problems $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, where $\mathbf{x}_i$ represents the problem's input vector and $\mathbf{y}_i$ is the corresponding optimised reference topology vector, by minimising an empirical loss function over its outputs,

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i), \quad (3.13)$$

after which evaluating f_{θ^*} on an unseen problem configuration is near-instantaneous. The appeal for topology optimisation is direct: the iterative optimiser, with its repeated finite-element solves, is replaced or accelerated by a single forward pass. The taxonomy of approaches that have been brought to bear on this problem is surveyed in Section 2.4; this section states the apparatus of the branch adopted here.

The work in this dissertation employs a convolutional *encoder-decoder* as its primary algorithmic engine. A convolutional neural network [96] processes gridded spatial inputs through a hierarchical sequence of filters that extract increasingly abstract spatial patterns. An encoder-decoder network arranges these layers symmetrically: a contracting encoder successively downsamples the initial input spatial resolution to compress it into a low-dimensional latent bottleneck representation, which an expanding decoder subsequently upsamples, through similar sequential layers, back to a full-resolution output. The U-Net, introduced for biomedical image segmentation, augments this structure with skip connections that carry high-resolution detail directly across the bottleneck to the corresponding decoding layers, allowing it to produce sharp, spatially precise outputs [97]; it has since been applied directly to generate optimised topologies [98]. Its data flow, from problem input to predicted structure, is shown in Figure 3.12.

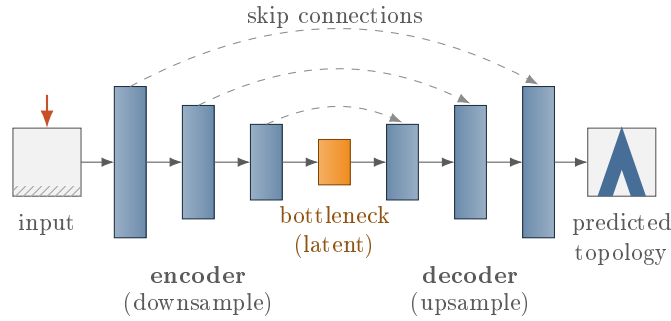


Figure 3.12: The convolutional encoder-decoder (U-Net): a contracting encoder compresses the input to a latent bottleneck, an expanding decoder reconstructs a full-resolution output, and skip connections carry high-resolution detail across the bottleneck.

Casting topology prediction in this image-to-image form treats the output as a dense, pixel-by-pixel classification problem. As networks infer continuous pseudo-density values bounded between solid and void $[0, 1]$ and are trained against this binary solid / void classification target, models are trained not only with a pixelwise regression error but with the overlap losses developed for segmentation. The most common is built on the Dice similarity coefficient between a prediction $\hat{\mathbf{y}}$ and the target $\mathbf{y}$,

$$\text{DSC} = \frac{2|\hat{\mathbf{y}} \cap \mathbf{y}|}{|\hat{\mathbf{y}}| + |\mathbf{y}|}, \quad \mathcal{L}_{\text{Dice}} = 1 - \text{DSC}, \quad (3.14)$$

whose minimisation maximises the overlap of the solid regions and which copes better than standard mean squared error with the imbalance between solid and void zones [99]. The training of such a predictor against a reference optimum is shown in Figure 3.13, and the

concrete architecture, training procedure, and results obtained with it are developed in Chapter 6.

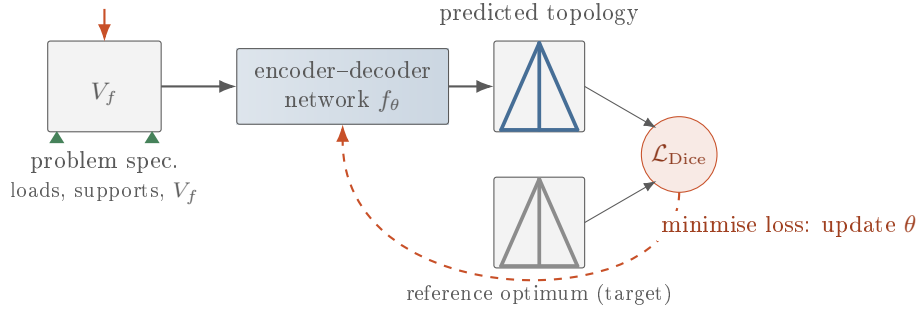


Figure 3.13: Supervised training of a topology predictor: the problem specification (domain, loads, supports, and volume fraction) is mapped by the network to a predicted topology, which is compared against the reference optimum through the loss of Equation (3.14).

Chapter 4

Development of the BESO Optimisation Framework

The state of the art of Chapter 2 and the theoretical foundations of Chapter 3 together established both the method this work adopts and the manufacturing context that motivates it: the convergent, mesh-independent, soft-kill bidirectional evolutionary scheme for topology optimisation, and the freedoms and constraints of selective laser melting that make such optimised geometries worth pursuing. This chapter presents the first of the three phases of the work, the development of the optimisation framework itself. It is the methodological core of the dissertation, in that the optimiser framework is a tool built and developed from the ground up rather than an application of an existing commercial or open-source implementation. The work that follows rests, therefore, upon it, beginning with its application to the Furia’s suspension component in the next chapter.

The framework takes the convergent BESO formulation as its foundation, and it carries two distinct engines upon a shared algorithmic core. The solid-void engine distributes material as a near-binary solid-or-void field, the classical form of topology optimisation. The lattice engine instead distributes a graded relative density, realised as an architected cellular solid, and so exploits the mesoscale lightweighting route surveyed in Section 2.3. Both engines are coupled to the Abaqus finite-element solver through a Python driver and a front-end that manages a complete run from an input model to printable geometry, and both are oriented throughout towards selective laser melting. To that end, the framework also provides an experimental overhang-mitigation feature for the solid-void engine, in which a steady-state thermal analysis acts as an inexpensive surrogate for the thermal conditions produced during the additive manufacturing build, biasing the optimiser away from unsupported geometry without the computational cost of a coupled transient thermomechanical simulation.

The chapter is organised around the line between what the two engines share and where they diverge. Section 4.1 sets out the common formulation: the sensitivity-driven update, the filtering and history-averaging that make it convergent and mesh-independent,

and the criterion by which the best design of a run is selected. Section 4.2 describes the software architecture through which that formulation is coupled to the finite-element solver and managed across a complete optimisation run. Section 4.3 then presents the solid-void engine, covering its binary design representation, the strain-energy sensitivity that drives it, and its volume-targeting update rule. The overhang-mitigation feature that extends the solid-void engine is treated separately in Section 4.4, where its thermal-proxy formulation and its integration into the sensitivity field are developed in full. Section 4.5 presents the lattice engine, which replaces the binary representation with a graded relative-density field and replaces the compliance objective with a fully-stressed-design criterion drawn from the Gibson–Ashby framework. Section 4.6 then describes how the output of either engine is converted into printable geometry. Section 4.7 closes the chapter by verifying the framework against benchmark problems of known solution, establishing its correctness before it is applied to a real component.

4.1 BESO Formulation

Both engines are built on the formulation set out in this section. The framework operates on a fixed design domain discretised into finite elements. Each element e is assigned a scalar design variable x_e that encodes the material state of that element: in the solid-void engine x_e takes a value of either one, for a fully solid element, or a small positive residual $x_{\min}$, for an element treated as void; in the lattice engine x_e is a relative density that varies continuously between $x_{\min}$ and $x_{\max}=1$, controlling the graded stiffness assigned to that element. How x_e maps to an element modulus is specific to each engine and is defined with it in Sections 4.3 and 4.5; what the two engines share is the mechanism by which x_e is updated from one iteration to the next, and that mechanism is what this section describes.

At every iteration the current set of design variables defines a material distribution, the Abaqus solver computes the linear static response of that distribution, and the result is used to evaluate a scalar sensitivity number α_e for each element. The sensitivity quantifies how strongly element e contributes to the objective: a high value indicates that the element is carrying load effectively and should be retained or strengthened, while a low value marks it as a candidate for removal or density reduction. For the solid-void engine α_e is the elemental strain energy of Equation (3.3); for the lattice engine it is a stress-based utilisation measure introduced in Section 4.5. Throughout the iteration loop the scheme is soft-kill: an element whose design variable is set to $x_{\min}$ is not removed from the mesh but retains a small residual stiffness, so that the global stiffness matrix $\mathbf{K}$ remains non-singular and the sensitivity of such elements stays computationally defined. The motivation for this is given in Section 3.1; the residual value $x_{\min}$ is kept small enough that these elements contribute negligibly to structural performance.

When the structure is subjected to more than one load case, a separate sensitivity field $\alpha_e^{(k)}$ is obtained from each solve. Rather than treat the load cases in turn, the framework

reduces them to a single combined field by a weighted sum,

$$\alpha_e = \sum_{k=1}^L w_k \alpha_e^{(k)}, \quad \sum_{k=1}^L w_k = 1, \quad (4.1)$$

where L is the number of load cases and the weights $w_k \geq 0$ are assigned by the user to reflect the relative severity or importance of each case. This combined field is what enters the regularisation steps that follow.

Evolutionary methods are prone to checkerboard patterns and to mesh dependence, as described in Section 3.1, and the accepted remedy is to regularise the sensitivity field with a spatial filter before using it to update the design. The filter replaces the raw sensitivity of each element by a weighted average over the neighbourhood of elements lying within a fixed physical radius $r_{\min}$,

$$\tilde{\alpha}_i = \frac{\sum_{j \in N_i} w(r_{ij}) \alpha_j}{\sum_{j \in N_i} w(r_{ij})}, \quad N_i = \{j : r_{ij} < r_{\min}\}, \quad (4.2)$$

where r_{ij} is the distance between the centroids of elements i and j . Dividing by the sum of the weights ensures that the filtered value $\tilde{\alpha}_i$ is a proper average of its neighbours and therefore always lies within the range of sensitivities present in the neighbourhood, rather than amplifying or attenuating the field. Two weighting functions, referred to here as kernels, are provided; each assigns a weight $w(r_{ij})$ to neighbour j as a function of its distance from element i ,

$$w(r_{ij}) = \begin{cases} r_{\min} - r_{ij}, & \text{linear,} \\ \exp\left(-\frac{r_{ij}^2}{2\sigma_g^2}\right), & \sigma_g = \frac{r_{\min}}{1.25}, \text{ Gaussian.} \end{cases} \quad (4.3)$$

The linear kernel decreases uniformly from $r_{\min}$ at the centre to zero at the boundary of the neighbourhood, so all neighbours within the radius contribute and those closer to element i are weighted more strongly. The Gaussian kernel concentrates weight more sharply around the centre and decays smoothly to near-zero well before the radius, giving a gentler transition at the neighbourhood boundary. Both kernels are positive throughout the neighbourhood and zero outside it.

The filter radius $r_{\min}$ is a physical length that sets the minimum member width the optimiser can resolve; a feature smaller than $r_{\min}$ cannot survive the averaging step. This also imposes mesh-independence on the converged design, since $r_{\min}$ is a fixed spatial scale rather than a count of elements. To keep the cost of Equation (4.2) negligible over the many iterations of a run, the neighbour search is performed once at the outset and the normalised weights are stored as a sparse operator, so that filtering at each iteration reduces

to a single sparse matrix–vector product.

A spatial filter does not, on its own, prevent the design from oscillating between competing layouts as elements are added and removed from iteration to iteration. To stabilise the optimisation history, Huang and Xie [21] proposed averaging each filtered sensitivity with the smoothed value from the preceding iteration, and the present framework follows this directly. The sensitivity that enters the design update at iteration t is

$$\bar{\alpha}_i^{(t)} = \frac{1}{2} \left(\tilde{\alpha}_i^{(t)} + \bar{\alpha}_i^{(t-1)} \right), \quad (4.4)$$

where $\tilde{\alpha}_i^{(t)}$ is the filtered sensitivity of the current iteration and $\bar{\alpha}_i^{(t-1)}$ is the stabilised value from the iteration before it, with the recursion initialised at $t = 1$ by setting $\bar{\alpha}_i^{(0)} = \tilde{\alpha}_i^{(1)}$. This running average damps iteration-to-iteration fluctuations and, together with the filter, is what gives the method its convergent behaviour.

Because the design evolves over many iterations, the framework does not simply accept the final layout but selects, from the entire history of a run, the iteration that performs best against a single figure of merit. That figure of merit is the specific stiffness, the structural stiffness carried per unit of retained mass. The stiffness is taken as the ratio of the total applied load to a representative displacement, and so that it remains consistent with the load-case weighting of Equation (4.1) the displacement is formed as the same weighted average across the load cases,

$$\kappa = \frac{F}{d\phi}, \quad d = \sum_{k=1}^L w_k \max_n \|\mathbf{u}_n^{(k)}\|, \quad (4.5)$$

where F is the total applied load, d is the weighted maximum nodal displacement, $\max_n \|\mathbf{u}_n^{(k)}\|$ denotes the largest nodal displacement magnitude under load case k , ϕ is the fraction of the total domain mass retained in the current design, and κ is the specific stiffness. The design that maximises κ over the run is the one reported, so that the framework returns the most mass-efficient structure it encounters rather than the layout produced at the final iteration.

This common core, a combined sensitivity field, a mesh-independent spatial filter, a stabilised history, and selection by specific stiffness, is shared without alteration by both engines. The engines specialise it in three respects: the physical field that populates $\alpha_e^{(k)}$, the rule by which the stabilised field $\bar{\alpha}_i$ is used to update the design variables x_e , and the criterion that terminates a run. These specialisations are set out in Sections 4.3 and 4.5; the next section first describes the software framework through which the common formulation is executed.

4.2 Software Architecture

The formulation of the previous section requires a finite-element solution at every iteration, and the framework does not produce that solution itself. It drives an external solver instead: a layer of Python orchestrates the optimisation and delegates each structural analysis to Abaqus, then reads the results back to decide the next change to the design. This solver-in-the-loop arrangement is the central architectural choice of the framework, and it is a deliberate one. It places the optimisation on top of a validated industrial solver rather than a hand-written one; it lets the framework accept ordinary Abaqus models, with their established treatment of multi-step load cases, boundary conditions and material definitions; and it scales without modification from the small meshes used in testing to the large three-dimensional models of a real component.

A complete run involves three parts, which act in sequence. The first is the launcher, an HTML-based front-end that serves as the user's workspace: it is where a run is configured, started and monitored, and it also provides a history view in which the specific-stiffness evolution of several runs can be compared against mass fraction or against iteration number. The second is the optimiser driver, the Python program that runs inside the Abaqus environment and carries the iteration loop. The third is the solver, which the driver calls once per iteration to analyse the current design. The remainder of this section follows a run through these parts, from the preparation of the model to the export of a printable result, as set out in Figure 4.1.

Before the first iteration, the driver makes a single preparatory pass over the model. It reads the mesh and records the node coordinates and the element connectivity. It detects the element type and whether the model is two- or three-dimensional, which determines how the structural response is later interpreted. It checks that the mesh is of coherent element size and, where the elements vary, reports their size distribution to the user as a warning, since a non-uniform mesh changes the meaning of the filter radius. It identifies the design domain from a named element set, `NON_DESIGN_SET`, that the user defines in the model: every element placed in that set is held fixed and solid throughout the run, and all remaining elements form the design domain on which the optimiser acts. Finally, it builds the filter neighbourhood map of Section 4.1 once and stores it as the sparse operator used thereafter, so that none of this preparation is repeated inside the loop.

The iteration loop then proceeds as a fixed sequence. At the start of each iteration the driver writes a new Abaqus input file in which the current design is represented by giving every element the material that corresponds to its design variable. The solver analyses this model, and the driver reads the resulting output database for the field that drives the update and for the nodal displacements. From that field it forms the combined, filtered and stabilised sensitivity defined in Section 4.1, updates the design variables x_e accordingly, and evaluates the specific stiffness of the new design, retaining the design if it is the best obtained so far. A convergence test then decides whether to begin another iteration or to stop. This sequence is identical for both engines; they differ only within three of its steps,

marked in Figure 4.1: the field read from the output database, the rule by which the design variables are updated, and the convergence test. The solid-void engine may additionally insert an overhang-mitigation step, which is optional even for that engine and is developed in Section 4.4.

When the loop stops, the framework returns not the final design but the best-scoring one recorded during the run, in keeping with the selection criterion of Section 4.1. This design is exported in the form the post-processing stage consumes: an explicit list of solid elements for the solid-void engine, or a map of relative densities for the lattice engine. The conversion of either into printable geometry is the subject of Section 4.6.

A final property of the architecture is its portability across computing environments. The same driver runs unchanged under the Python 2 interpreter of an older local Abaqus installation and the Python 3 interpreter of the newer version installed on the university supercomputer. Small models are therefore developed and run on a workstation, while large models, such as the Furia component of the following chapter, are dispatched to the cluster, where the additional processor cores and memory make an otherwise intractable run feasible. The framework itself is identical in both settings.

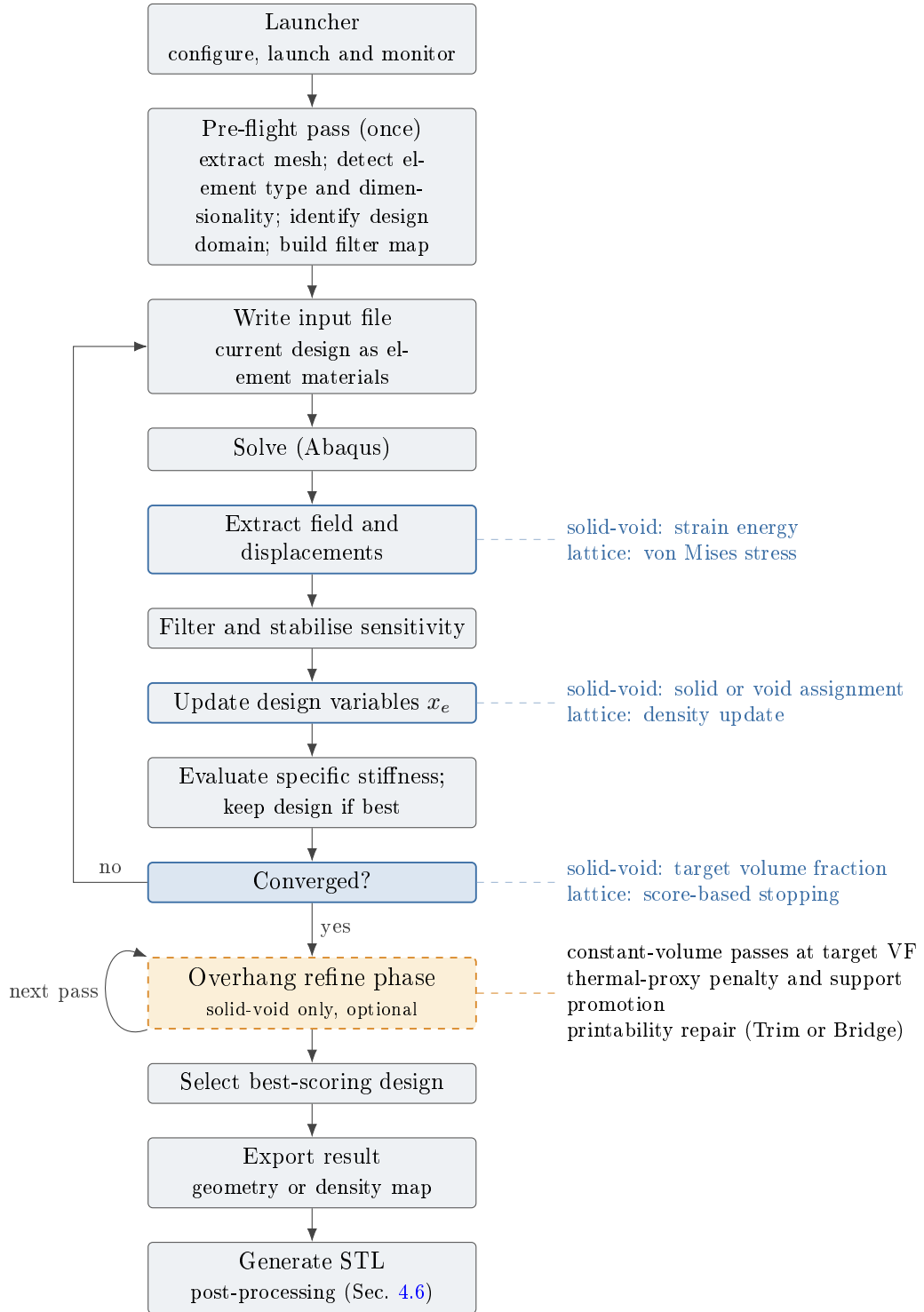


Figure 4.1: Architecture of the optimisation framework. A single pre-flight pass prepares the model and builds the filter map; the driver and the Abaqus solver then execute the iteration loop until convergence, after which the best-scoring design is exported and converted to printable geometry. The design variables updated in the loop are the elemental values x_e of Section 4.1, binary in the solid-void engine and a graded relative density in the lattice engine. The three steps that differ between the engines, namely the extracted field, the design update and the convergence test, are outlined and annotated on the right. The overhang refine phase is optional and available only to the solid-void engine; it holds the volume fraction at its target and is described in Section 4.4.

4.3 Solid-Void Engine

The solid-void engine treats topology optimisation as a discrete, two-state problem: every element of the design domain is either solid or void, and the optimiser decides which. It is the first of the two engines built on the shared core of Section 4.1, and the simpler of the two. This section defines the binary material model that Section 4.1 deferred, and then the three respects in which the engine specialises that core: the field that drives its sensitivity, the rule by which it updates the design, and the criterion by which it stops.

In this engine the design variable x_e of Section 4.1 is restricted to two values, so that every element is either fully solid or treated as void,

$$x_e \in \{x_{\min}, 1\}, \quad E_e = x_e^p E_0, \quad (4.6)$$

where E_0 is the modulus of the fully solid material, p is the penalisation exponent of Equation (3.1), and E_e is the modulus assigned to element e . A solid element, with $x_e = 1$, takes the full modulus E_0 . A void element, with $x_e = x_{\min}$, takes the small residual modulus $x_{\min}^p E_0$, which is the soft-kill stiffness described in Section 4.1: large enough to keep the global stiffness matrix non-singular, but small enough to leave the element structurally negligible. Because x_e never takes an intermediate value here, the penalisation exponent p has no grey transition between solid and void to suppress, as it would in a density method; it serves only to drive the void residual towards zero. The material distribution is therefore genuinely discrete, and the task of the engine is to decide, at a prescribed volume fraction, which elements are solid.

The field that drives the engine is the elastic strain energy of the solved design, which measures how hard each element is working and so how much it contributes to the overall stiffness. The classical evolutionary sensitivity, given in Equation (3.3), is the total strain energy of an element. The present engine uses instead the strain energy *density*, the energy per unit volume,

$$\alpha_e^{(k)} = \frac{1}{v_e} \cdot \frac{1}{2} (\mathbf{u}_e^{(k)})^T \mathbf{k}_e \mathbf{u}_e^{(k)}, \quad (4.7)$$

where $\mathbf{u}_e^{(k)}$ is the displacement of element e under load case k , $\mathbf{k}_e$ is its stiffness matrix, and v_e is its volume. On a uniform mesh, where every element has the same volume, ranking elements by Equation (4.7) and by the total strain energy of Equation (3.3) produces the identical order, so the two measures are interchangeable. On a non-uniform mesh they are not: the total strain energy lets a large element accumulate a high value simply by occupying more volume, whereas the density judges each element on the intensity of its strain energy per unit material, independently of its size. The density is therefore the sounder choice for a general design domain, and it is the quantity the engine reads from the solver. These per-element densities are combined across the load cases by Equation (4.1), then filtered and stabilised by Equations (4.2) and (4.4), exactly as set out for the shared core.

At each iteration the engine ranks every design-domain element by its stabilised sensitivity, the quantity $\bar{\alpha}_i$ of Equation (4.4), from lowest to highest. The lowest-ranked elements, whose material is the least effectively used, are set to void up to the current target void fraction $r^{(t)}$, and the remainder are set solid; the elements of the non-design region of Section 4.2 are held solid throughout and take no part in the ranking. The rule is bidirectional, and this is the property that separates it from the removal-only evolutionary methods discussed in Section 2.1. Because the full ranking is recomputed from the current solution at every iteration, the state of an element is never permanent: an element set to void early in the run can be restored to solid later if, as the surrounding structure changes, its sensitivity climbs in the ranking. Material is thus both removed and added back as the design evolves, which lets the engine recover from premature removals that a one-directional method could never undo.

The target void fraction is not imposed at once but approached gradually, so that the structure can redistribute load between successive solves rather than losing a large amount of material in a single step. Writing $r^{(t)}$ for the fraction of design-domain elements held void at iteration t , the engine advances it by the evolutionary ratio, denoted ER, until the target is reached,

$$r^{(t)} = \min\left(r^{(t-1)} + \text{ER}, 1 - f\right), \quad (4.8)$$

where f is the prescribed target volume fraction, the fraction of the design domain to remain solid, so that $1 - f$ is the final void fraction. The evolutionary ratio may be held fixed for the whole run, or varied dynamically, decreasing from a larger initial value to a smaller final one as the void fraction approaches its target. The dynamic schedule is the more efficient of the two. Early in the run, while much of the domain still carries almost no load, a large ratio removes that ineffective material quickly and cheaply. Later, when the structure is close to its final form and each remaining element matters, a small ratio refines the design in fine steps and captures the higher-stiffness layouts that a coarse step would overshoot, without paying the cost of a small fixed ratio over the entire run.

The engine has no convergence test of its own beyond this schedule: the run is complete when the void fraction reaches $1 - f$, at which point the design has been reduced to the prescribed volume fraction. The engine then reports two designs rather than one. The first is the design at the target volume fraction itself, the structure as it stands when the search ends. The second is the iteration of highest specific stiffness encountered along the way, identified by the selection criterion of Equation (4.5), which the engine evaluates at every iteration of the sweep.

These two designs do not necessarily coincide. Because the search removes material monotonically from the full domain down to the target, the highest specific stiffness can be reached at a fraction fuller than the target, where the trade-off between stiffness and mass is most favourable; for moderate target fractions, however, the two often fall on the same final iteration. Saving both lets the user choose between the leanest design that meets the prescribed volume fraction and the most mass-efficient design found during the search.

The design at the target fraction is the one used when the engine’s output is compared against a benchmark defined at a fixed volume fraction, as in Section 4.7.

This completes the solid-void engine as a pure stiffness optimiser. The next section extends it with one further feature, active only for this engine and only when requested, that biases the result towards designs which can be built without sacrificial supports.

4.4 Overhang Mitigation via Thermal Analysis

Section 2.2 set out the overhang problem and the two classes of remedy that have been proposed within topology optimisation: the geometric methods, which are either too crude, as with the bare critical-angle rule, or too heavy, as with the projection filters and explicit boundary constraints; and the physics-based methods, which infer where supports are needed from a simulation of the manufacturing of the component itself but, at their most faithful, carry the cost of a coupled transient thermomechanical analysis. The feature developed here sits at the accessible end of the physics-based branch, but it is not a pure physics method. It pairs an inexpensive thermal proxy, which identifies where unsupported material sits and how much it matters, with a geometric mechanism that casts supporting material beneath that overhang and keeps the result printable. It is, in short, a hybrid of a thermal indicator and a geometric constructor, and it is best understood in those two parts.

Two properties of the feature should be stated at the outset. First, it is experimental: it is offered as an option rather than as an established part of the method, and the latter half of this section characterises its behaviour rather than asserting a single recommended configuration. Second, it applies only to the solid-void engine, and it is applied as a refinement of an already-optimised structure rather than as a constraint carried through the search. The run is divided into two phases. A *build* phase first carries out the pure mechanical optimisation of Section 4.3, with the mitigation switched off, down to the target volume fraction. A *refine* phase then holds the volume fraction fixed at that target and runs a set number of constant-volume passes with the mitigation switched on, so that the overhang correction reshapes the converged structure without altering the amount of material in it. The thermal analysis is solved throughout both phases, so the temperature field is available the moment refinement begins.

4.4.1 Thermal proxy

The first half of the feature, the thermal proxy, identifies where unsupported material lies by running a simple heat-conduction analysis on the current design and reading off how hot each element becomes, as sketched in Figure 4.2. This thermal analysis is a separate, steady-state step added to the model: every solid element generates heat at a uniform rate, the build plate is held at a fixed temperature and acts as the only heat sink, and the resulting steady temperature of each element is recorded. The conductivity is made

anisotropic and keyed to the build direction: it takes a reference value $\lambda_{\parallel}$ along the build axis and is throttled to a small fraction of that value in the two transverse directions,

$$\lambda_{\perp} = a \lambda_{\parallel}, \quad 0 < a \ll 1, \quad (4.9)$$

where a is the anisotropy ratio, set as a percentage of the build-direction conductivity by the user and equal to 0.05 in this work. Only the ratio matters, not the absolute values, since the steady temperature pattern depends on how heat is allowed to flow rather than on the magnitude of the source.

The anisotropy of Equation (4.9) is what turns temperature into a usable signal for whether an element is supported. Because the transverse conductivity is throttled, heat escapes readily down the build direction toward the plate but only weakly sideways. An element with an unbroken column of solid material beneath it, in the build direction, therefore has a short, low-resistance path to the heat sink and stays cool. An unsupported, downward-facing element has no such path: its heat must detour sideways through the throttled transverse direction, so it runs hot. A high temperature therefore marks precisely the configuration that defines an overhang, and the hotter the element, the more isolated it is from the plate.

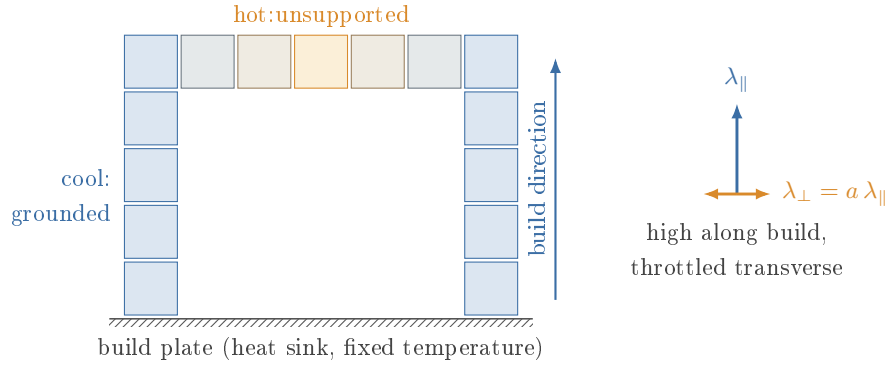


Figure 4.2: The thermal proxy. Every solid element generates heat uniformly and the only escape is conduction to the build plate, held at a fixed temperature. The conductivity is anisotropic, high along the build direction, $\lambda_{\parallel}$, and throttled to a small fraction a of that value transversely, $\lambda_{\perp} = a \lambda_{\parallel}$, with $a = 0.05$ here. The grounded legs have a direct path to the plate and stay cool; the unsupported elements bridging between them must route heat transversely and run progressively hotter towards mid-span. Temperature therefore becomes a proxy for non-supportedness.

4.4.2 Sensitivity penalty

The temperature field is turned into a sensitivity penalty in three steps: a geometric gate, a robust normalisation, and a spatial smear. The gate restricts the penalty to genuine overhangs. An element is an *active violator* if, within a neighbourhood of one and a half element-lengths, it has no solid neighbour directly beneath it in the build direction;

a neighbour at one element-length directly below supports it, and so does a diagonal neighbour at the critical angle, so a violator is an element that is genuinely unsupported. Elements on the build-plate face are exempt, being inherently supported.

The temperature of each violator is then normalised. Rather than scale against the minimum and maximum temperatures, which a single isolated fragment running away to a very high temperature would stretch, crushing every genuine overhang toward zero, the penalty is normalised against a high percentile of the solid-element temperatures and clipped,

$$\tau_e = \begin{cases} \min\left(\frac{T_e}{T_{\text{ref}}}, 1\right), & e \text{ an active violator,} \\ 0, & \text{otherwise,} \end{cases} \quad T_{\text{ref}} = P_{95}(\{T_e : e \text{ solid}\}), \quad (4.10)$$

where T_e is the temperature of element e and T_{ref} is the ninety-fifth percentile of the solid temperatures. An isolated fragment that runs hotter than the reference simply saturates at a penalty of one rather than distorting the scale. The penalty field τ_e is finally smeared by the filter of Section 4.1, so that a violator shares its penalty with its immediate neighbourhood.

4.4.3 Directional support promotion

The penalty discourages unsupported material, but on its own it can only remove it. The constructor half of the feature does the opposite: it actively grows a support beneath an overhang, as sketched in Figure 4.3. This is the more novel mechanism, and it is wholly geometric in character, the temperature serving only to set the strength of the demand.

It depends on the notion of a *grounded* element. The engine identifies the grounded set by a flood fill seeded from the solid elements on the build plate, in which two solid elements are treated as connected if they explicitly touch across a face. Any solid element the fill cannot reach is not connected to the plate by such a chain and is a floating island, which cannot serve as a foundation. For each hot violator, the engine then searches a downward cone of half-angle $\theta_c = 45^\circ$, the critical self-supporting angle established in Section 3.2 and Figure 3.6, for the nearest grounded solid within a chosen distance, the *support reach*, measured in element-lengths. That grounded solid is the foot the violator can lean on. Opening the cone to exactly the critical angle is deliberate: a strut grown from the violator to a foot found inside this cone is itself inclined at or above the self-supporting angle, so the support the engine builds is printable by the same rule it is trying to satisfy.

Support is then cast into the void between the violator and its foot. Each void candidate is given a bonus that is largest along the violator-to-foot line and falls away to either side,

$$\beta_e = \sum_s d_s \cos^q \theta_{s,e}, \quad d_s = \tau_s, \quad (4.11)$$

where the sum runs over the violators s whose cast reaches void element e , d_s is the

thermal demand of violator s (its clipped temperature ratio), $\theta_{s,e}$ is the angle between the cast axis and the line from s to e , and the exponent q is the directional sharpness, which concentrates the demand along the axis. The cast is gated so that the bonus accrues only to void elements that already touch the current solid, which keeps the growth contiguous, and the total amount of material promoted in a pass is capped. The effect is to lean the demand toward the foot and starve the far side, rotating a flat unsupported span toward an inclined, self-supporting strut rather than merely thickening it.

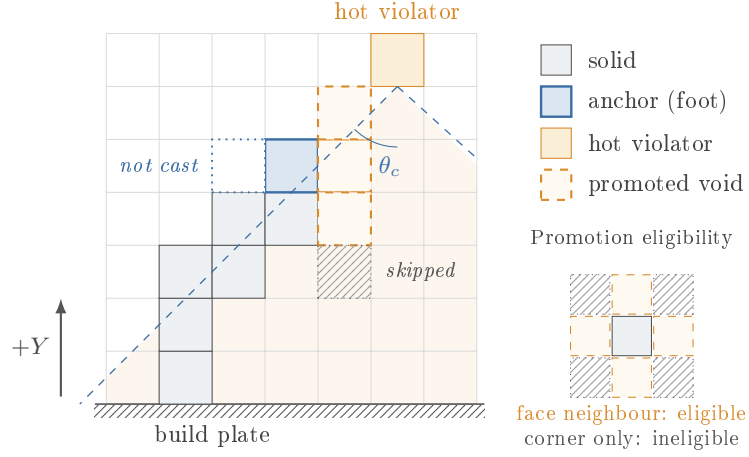


Figure 4.3: Directional support promotion onto face-adjacent void. For each hot violator the engine searches a downward cone of half-angle $\theta_c = 45^\circ$, the critical self-supporting angle of Figure 3.6, and takes the nearest grounded solid within that cone and within the support reach as the foot. Support demand is cast into the void along the violator-to-foot line, each candidate weighted by $\cos^q \theta$ so that the demand leans towards the foot. A void cell is promoted (dashed) only if it lies inside the cone and shares a full face with existing solid, so that the strut grows face by face rather than seeding the single-node contacts the repair step must later remove. The inset states the eligibility rule: face neighbours are eligible, diagonal neighbours are not.

4.4.4 Combined update and printability repair

In a refine pass, the penalty and the promotion enter the design update together. The combined sensitivity of element e is

$$\alpha_e = m_e (1 - t_w \tau_e) + t_w \beta_e, \quad (4.12)$$

where m_e is the mechanical sensitivity of Section 4.3 after it has been normalised to the unit interval, τ_e is the filtered overhang penalty of Equation (4.10), β_e is the support-promotion bonus of Equation (4.11), and $t_w \in [0, 1]$ is the thermal weight, the single knob that sets how strongly the overhang objective competes with stiffness. Each of the two corrections enters in a deliberately different way. The penalty enters multiplicatively, scaling the mechanical sensitivity down rather than subtracting from it, so a hot element is demoted in the ranking but in general not driven to nothing: it retains at least $m_e(1 - t_w)$ of its

original value. At the extreme $t_w = 1$ this residual vanishes, and an element that is both a genuine violator and as hot as the reference ($\tau_e = 1$) loses its mechanical sensitivity entirely, so even a structurally important member can be removed if it is maximally penalised; for the intermediate weights used in practice, however, the multiplicative form weakens such a member without deleting it. The promotion enters additively through β_e , which is non-zero only in the void region the engine is casting support into, so it can call new support into being in places the mechanical sensitivity alone would never favour. When $t_w = 0$ both corrections switch off and the expression collapses to the pure mechanical sensitivity m_e , recovering the engine of Section 4.3 exactly. The combined field then passes through the same stabilisation, Equation (4.4), and the same ranking-based update as in the unmitigated engine.

The promotion can leave the structure with defects that would not print, and the engine resolves three distinct kinds. The first is a genuinely floating island: a cluster of solid the grounded flood fill never reaches, with no connection to the plate at all. The second is a node-only cluster: a group of elements that is internally face-connected and reaches the plate only through a single corner contact, so that it counts as grounded when corner contacts are allowed but not when they are required to be face contacts. The third is the checkerboard contact illustrated on the left of Figure 4.4: two elements of the main structure that meet at a single node, with both of the cells that would complete a face join left empty, where each element is otherwise face-connected to the structure on its own side. The first is irreparable and is simply removed; the second and third are both single-node contacts, a corner join that is mechanically a hinge and too fragile to print, and these are what the two repair strategies heal.

The *Trim* strategy removes the offending corner element and re-establishes soundness by re-flooding the structure at face adjacency after every removal, which tends to reach the higher overhang reduction. The *Bridge* strategy instead promotes one of the two empty shared-face cells, converting the corner contact into a genuine face joint, and holds the volume fraction by a balanced removal of the least useful solid elsewhere; it builds the support up rather than cutting it back, and converges more steadily. Genuine floaters, having nothing to bridge to, are removed under both strategies. The two are compared quantitatively below.

Because the feature now reports a structure that has traded some stiffness for printability, the engine saves a third geometry alongside the two of Section 4.3: the refine pass with the fewest overhang violators, the *most-mitigated* design, which lets the user weigh the most printable result against the stiffest one.

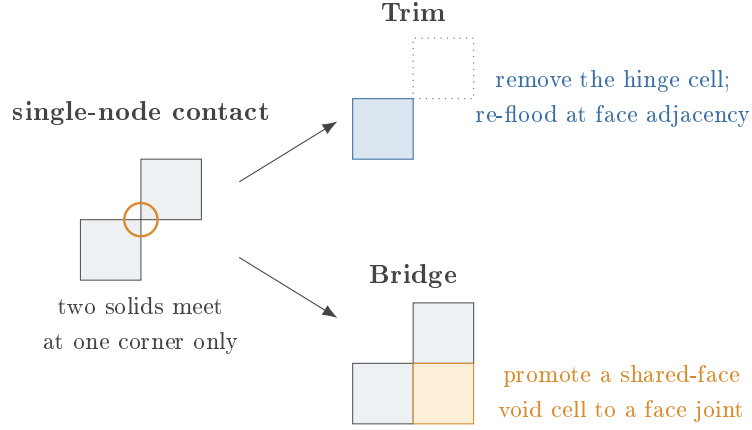


Figure 4.4: The two repair strategies for the single-node contacts left by promotion, where two solid cells touch only at a corner with both shared-face positions empty, a join too fragile to print. Trim removes the corner cell and re-floods the support at face adjacency, favouring peak overhang reduction. Bridge instead promotes one of the empty shared-face cells, converting the corner contact into a printable face joint and holding the volume fraction by a balanced removal elsewhere, which converges more steadily.

4.4.5 Parametric behaviour study

The feature has two principal controls, the thermal weight t_w and the support reach, together with the choice of repair strategy. Their effects were characterised on the benchmark beam of Section 4.7, sweeping t_w from 0.1 to 0.9 and comparing the two reaches and the two strategies. Throughout, overhang reduction is reported as the percentage fall in the violator count relative to the build endpoint, the structure as it leaves the build phase with no mitigation applied, and all summary values are means over the final passes of a run.

Figure 4.5 shows that the refinement converges. Across every penalty weight, the violator count falls sharply over the first few passes and then settles onto a stable plateau whose height decreases as t_w rises. The specific stiffness recovers and holds at essentially its build-phase level throughout; the sharp downward excursions seen in the first few passes of some curves belong to the larger penalty weights, where the more aggressive reshaping briefly shocks the structure before it re-settles, and they disappear as the passes proceed. The feature therefore reaches a steady state rather than churning, and it does so without a lasting stiffness penalty.

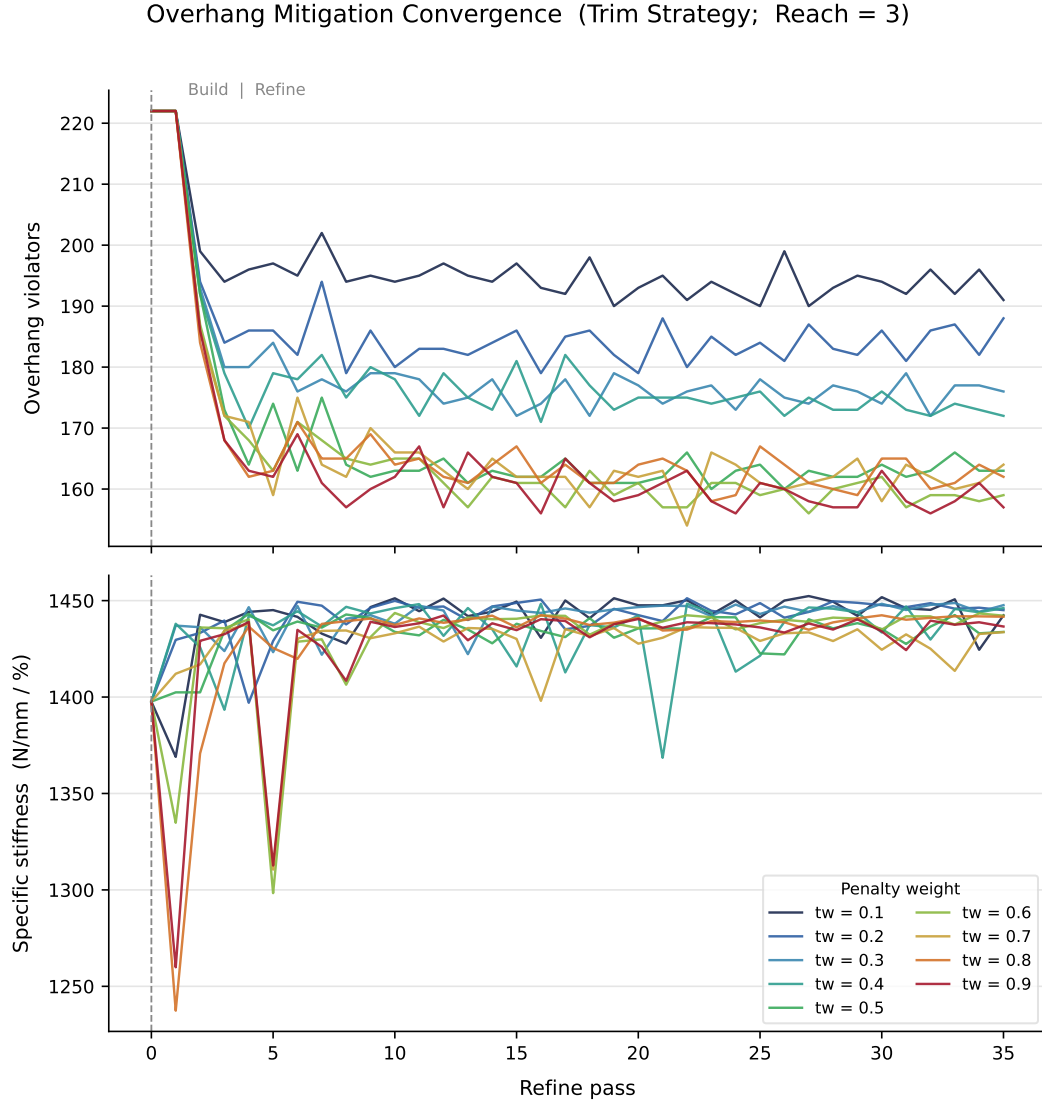


Figure 4.5: Convergence of the overhang mitigation over the refine passes, for the Trim strategy at support reach three and penalty weights t_w from 0.1 to 0.9. Pass zero is the build endpoint, with the mitigation off. Top: the overhang-violator count falls sharply over the first few passes and settles onto a stable plateau whose height lowers as t_w rises. Bottom: the specific stiffness recovers after a brief initial transient for higher t_w values, and all runs then hold at a slight increase from its build-phase level.

The trade-off this buys is shown in Figure 4.6, here for the shorter support reach of three element-lengths that is later adopted as the default. Overhang reduction rises steeply with the thermal weight and then saturates, reaching about 27%. The behaviour of the specific stiffness is more striking than a simple trade-off would predict: rather than merely holding its build-phase value, it rises above it for every penalty weight tested. The reason is that the refine phase does not abandon the mechanical objective when it begins to mitigate overhangs. The mechanical sensitivity remains the dominant, multiplicative term in Equation (4.12), so as the engine reshapes the structure toward printability it continues

to redistribute material toward where it is structurally useful, and it finds configurations that are at once more manufacturable and slightly stiffer per unit mass than the design the build phase left behind. A secondary implication is worth noting: since the constant-volume refine passes can improve specific stiffness on their own, a refine phase run with the mitigation switched off would likely sharpen the build-phase result as well, which is a natural direction for future work. The trade-off plane on the right of the figure summarises the headline outcome: a substantial fraction of the overhangs is removed at no cost in stiffness, and in fact at a small gain.

Stiffness-Overhang Trade-off (Trim Strategy; Reach = 3)

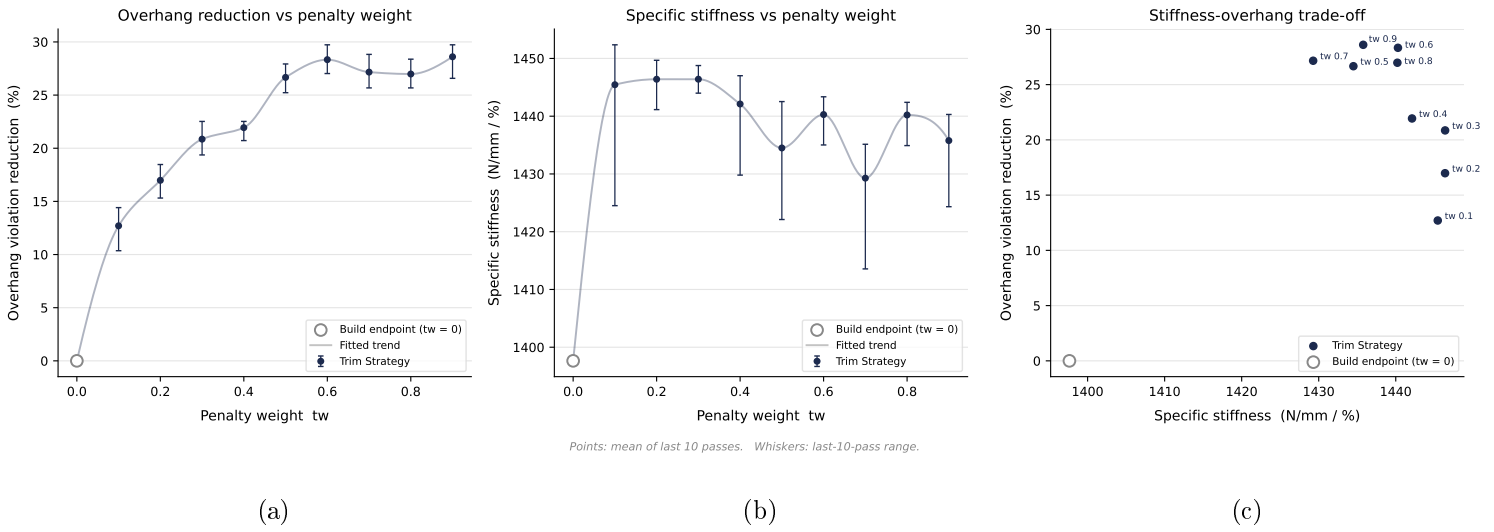


Figure 4.6: The stiffness-overhang trade-off, for the Trim strategy at support reach three. (a) The overhang-violation reduction, measured relative to the build endpoint, rises steeply with the thermal weight t_w and saturates near 27%. (b) The specific stiffness value showcases a significant increase as a result of the refinement phase, which slightly decreases as t_w increases. (c) The two quantities plotted against each other, each point labelled by its t_w . Points are means over the final passes of each run; whiskers span their range.

The support reach governs how far that trade-off can be pushed, as Figure 4.7 shows. A longer reach of six element-lengths casts support across larger gaps and nearly doubles the overhang reduction, to around 40%, but it does so at a real and far more variable cost in stiffness, visible in the long downward whiskers of the middle panel and the lower, more scattered cluster on the right. The shorter reach of three element-lengths keeps the casts local, which tends to strand fewer of the floating fragments a long cast can leave behind, and holds the stiffness tightly. For this reason the shorter reach is taken as the recommended default and the longer reach as an aggressive alternative, to be used where overhang reduction matters more than a small, predictable loss of stiffness.

Finally, Figure 4.8 compares the two repair strategies at the longer reach, where the single-node contacts are most frequent. Trim reaches a higher peak overhang reduction, but at the price of more churn, visible in the larger number of element flips between

consecutive passes in the stability panel. Bridge flips far fewer elements and so settles more smoothly, at a slightly lower peak reduction. The choice between them is thus one of peak performance against convergence stability, and it is left to the user, with Trim the default for its reach.

Effect of Support Cast Reach on the Trade-off (Trim Strategy)

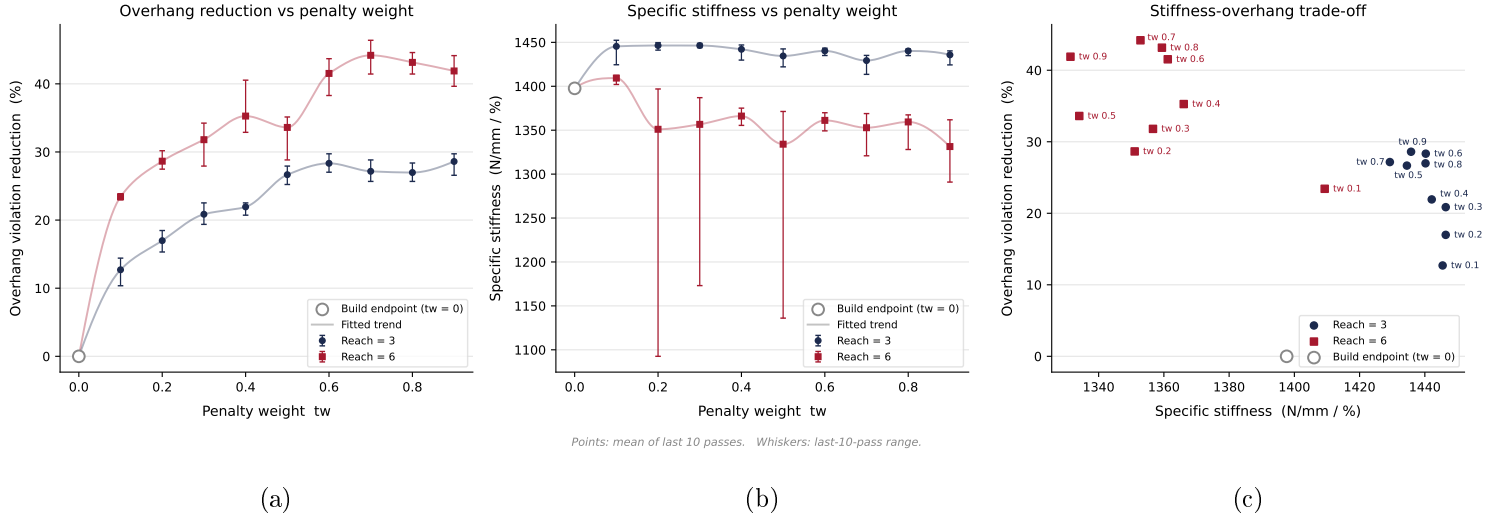
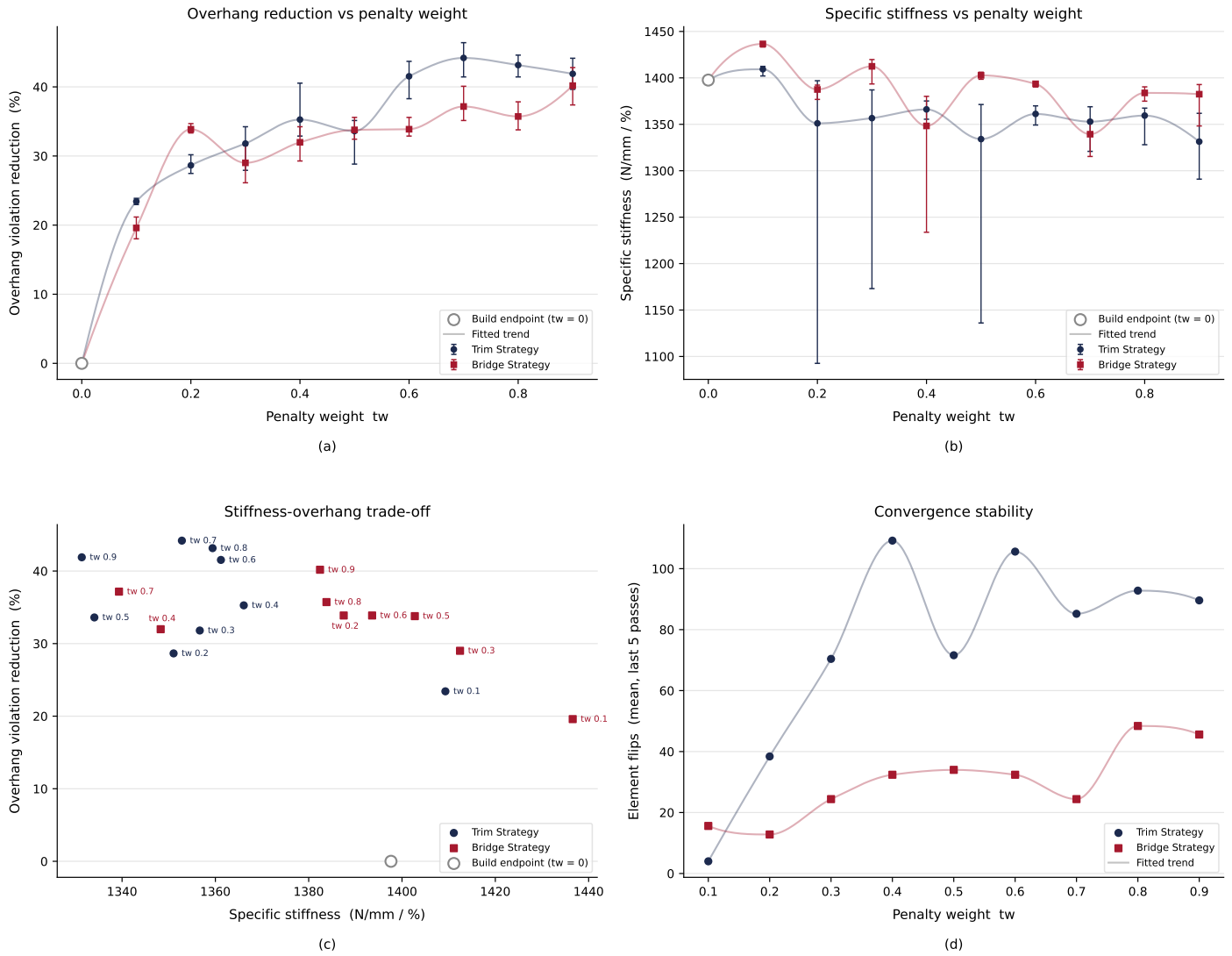


Figure 4.7: Effect of the support reach on the trade-off, for the Trim strategy. A reach of six element-lengths casts support across larger gaps and, as panel (a) shows, nearly doubles the overhang reduction relative to a reach of three, reaching around 40%, but at a real and far more variable cost in stiffness: note the long downward whiskers in panel (b) and the lower specific stiffness values in a more scattered cluster, in panel (c). The shorter reach holds the stiffness tightly and is taken as the recommended default.

Trim vs Bridge Strategy (Reach = 6)



Points: mean of last 10 passes. Whiskers: last-10-pass range.

Figure 4.8: The Trim and Bridge repair strategies compared at support reach six, where the single-node contacts are most frequent. Trim reaches a higher peak overhang reduction, panel (a), but at the price of more churn, visible as the larger number of element flips between consecutive passes in the convergence-stability panel (d). Bridge flips far fewer elements and settles more smoothly, at a slightly lower peak in overhang reduction. The specific stiffness against penalty weight, panel (b), and the stiffness-overhang trade-off, panel (c), showcase, respectively, how the Bridge strategy achieves a slightly higher specific stiffness, at a small cost in overhang reduction, for this larger reach. The choice is one of peak reduction against convergence stability.

4.4.6 Scope of the overhang mitigation feature

The mitigation is restricted to the solid-void engine, for two reasons. The first is that it is experimental, and extending it into the stress-driven, graded-density update of the lattice engine would be a study in its own right with an uncertain return. The second is structural: the geometric half of the feature is defined on a binary solid-void field. The grounded-set flood fill, the single-node-contact repair, and the very notion of a face-connected printable strut all assume that an element is either present or absent, and they have no clean counterpart on a field of continuously graded relative densities. A graded lattice also tends to retain more intermediate-density material throughout the domain and is closed by a manufacturability skin, so the open, unsupported spans the feature targets are in any case less prevalent in a lattice result than in a solid-void one. The lattice engine, which has its own and very different formulation, is the subject of the next section.

4.5 Lattice Engine

The solid-void engine of Section 4.3 distributes material as a binary field and drives it by strain energy. The lattice engine is the second specialisation of the shared core of Section 4.1, and it diverges from the first in every one of the three respects that the core leaves open. Its design variable is not a binary state but a continuous relative density; the field that drives it is not strain energy but von Mises stress; and it does not terminate at a prescribed volume fraction but searches for the iteration of highest specific stiffness. In place of the compliance logic of the solid-void engine it pursues a fully-stressed design, sizing every element so that the material it holds is worked as close to its capacity as the load allows. It does so by treating each element as a homogenised cell of an architected lattice whose stiffness and strength follow the Gibson–Ashby scaling of Section 3.4.

The section develops the engine in four parts. Section 4.5.1 sets out the graded material model, the morphologies the engine supports, and the two ways their property laws may be evaluated. Section 4.5.2 defines the utilisation measure that orders the design and the fully-stressed target the update chases, together with the two load modes that target may serve. Section 4.5.3 describes the density update itself and the score-based criterion that ends a run. Section 4.5.4 closes with the optional solid skin that may be applied to the converged design.

4.5.1 Graded material model

The lattice engine never represents the cellular architecture explicitly. Meshing a graded field of unit cells at their true scale would be intractable for a component-sized domain, and it is unnecessary for the purpose of distributing density: what the optimiser needs from each element is not its internal geometry but its effective stiffness and strength. The engine therefore treats every element of the design domain as a homogenised continuum whose effective properties are functions of a single scalar, the relative density $\bar{\rho}$ of Section 3.4,

the fraction of the cell volume occupied by solid. The design variable x_e of the shared core is, for this engine, exactly that relative density, $x_e \equiv \bar{\rho}_e$, varying continuously between the residual lower bound and unity. The mesh is fixed for the whole run, and only the property assigned to each element changes as its relative density evolves.

The effective properties of a cellular solid depend not only on how much material a cell contains but on the morphology of the cell itself, as established in Section 3.4. The engine offers six morphologies based on triply periodic minimal surfaces, drawn from the two structural families distinguished there:

1. *Ligament*, or skeletal, morphologies, in which the solid occupies a connected network of struts thickened about the minimal surface. The gyroid and the diamond are provided. Their members carry a significant share of their load in bending, so their stiffness exponents lie close to two, the bending-dominated value of Section 3.4, and they are correspondingly less efficient per unit mass than the sheet family.
2. *Sheet* morphologies, in which the solid forms a thin double wall offset symmetrically about the minimal surface. The gyroid, the Schwarz primitive, the I-WP and the Neovius are provided. The continuous, closed topology of a sheet lattice deforms in a more stretching-dominated manner, with stiffness exponents nearer one, so for a given relative density it is generally stiffer and stronger than its ligament counterpart, as the deformation-mode argument of Section 3.4 predicts.

The six morphologies are shown together in Figure 4.9.

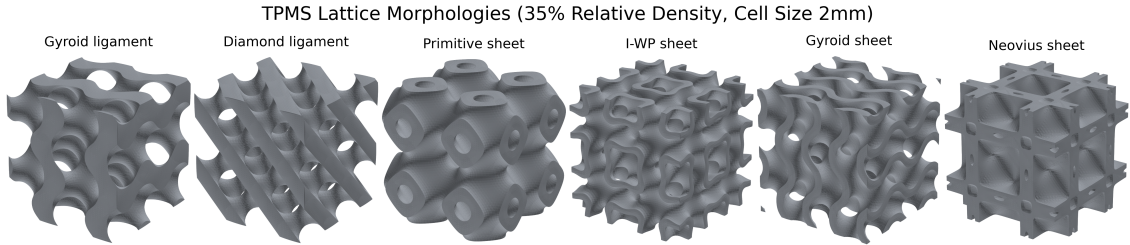


Figure 4.9: The six triply periodic minimal surface morphologies supported by the lattice engine, each rendered at a representative relative density. The ligament (skeletal) family comprises the gyroid and diamond; the sheet family comprises the gyroid, Schwarz primitive, I-WP and Neovius. Each is characterised by a single row of Gibson–Ashby coefficients in Table 4.1.

Within a morphology, the effective stiffness and yield strength scale with relative density through the Gibson–Ashby power laws of Equation (3.12). For this engine the stiffness coefficient C_1 and exponent n_E , and the strength coefficient C_2 and exponent n_σ , take a distinct set of values for each morphology, while E_s and σ_{ys} remain the Young’s modulus and yield strength of the fully dense parent material. The four coefficients are dimensionless and, being descriptors of the cell geometry rather than of the material that fills it, are independent of that material: a single row describes the morphology in any parent material

once multiplied by its E_s and σ_{ys} . Each row constitutes the sole definition of the lattice law within the engine: the same coefficients set the modulus assigned to every density bin, the strength used to size the elements, and the utilisation field reported for inspection, so that no part of the engine can apply a different law from another.

The coefficient set adopted in this work is given in Table 4.1. These coefficients are most reliably obtained by numerical homogenisation, in which a representative unit cell is solved under periodic boundary conditions on a normalised elastic–perfectly-plastic base; because the result is a ratio between the effective and parent properties, it is dimensionless and applies to any parent material without re-fitting. The gyroid and diamond ligament rows [100] and the I-WP sheet row [101] are single-source power-law fits of exactly this kind and are the cleanest of the six. The Schwarz primitive sheet row [102] is taken from the high relative-density branch of that study’s fit, valid above a relative density of one tenth, which coincides with the lowest density the engine assigns by default, so the working range never leaves the validated band. Two rows depart from this ideal and are flagged accordingly.

Table 4.1: Gibson–Ashby stiffness (C_1 , n_E) and strength (C_2 , n_σ) coefficients adopted for the six morphologies, with the provenance of each row. The values are dimensionless and shared, unchanged, by the optimiser, the launcher and the post-processor.

Morphology	C_1	n_E	C_2	n_σ	Source
Gyroid ligament	0.9515	2.174	0.6182	1.746	Guo et al. [100]
Diamond ligament	0.6438	2.026	0.6802	1.614	Guo et al. [100]
Schwarz primitive sheet	0.61	1.57	0.794	1.36	Lee et al. [102]
I-WP sheet	0.699	1.217	0.738	1.151	Abu Al-Rub et al. [101]
Gyroid sheet ^a	0.5909	1.3454	0.7298	1.2066	Pais et al. [33]
Neovius sheet ^b	0.705	1.236	0.721	1.5	[103, 104]

^a Validated power-law approximation of a polynomial source (see Figure 4.10).

^b Mixed-source row: stiffness from [103], strength from [104]; treated as indicative.

No material-independent study reports the gyroid *sheet* response directly as a Gibson–Ashby power law. The closest source, the numerical homogenisation of Pais et al. [33], gives the effective stiffness and yield as cubic polynomials in relative density rather than as power laws. To bring this morphology into the engine’s single power-law form, a power law was regressed onto each polynomial over the band from the void threshold to the density cap defined below, which in the Safe model addressed in the Subsection 4.5.1.1, is the only range over which a graded gyroid-sheet density is assigned. The fitted law reproduces the polynomial source to within about 2.5% root-mean-square error across that band, with the largest deviations, around 6.5%, confined to the two band edges where graded material is least often placed; the comparison is shown in Figure 4.10. The gyroid sheet row is therefore a validated power-law approximation of a polynomial source, not a directly reported power

law, and is the one morphology whose functional form had to be re-fitted.

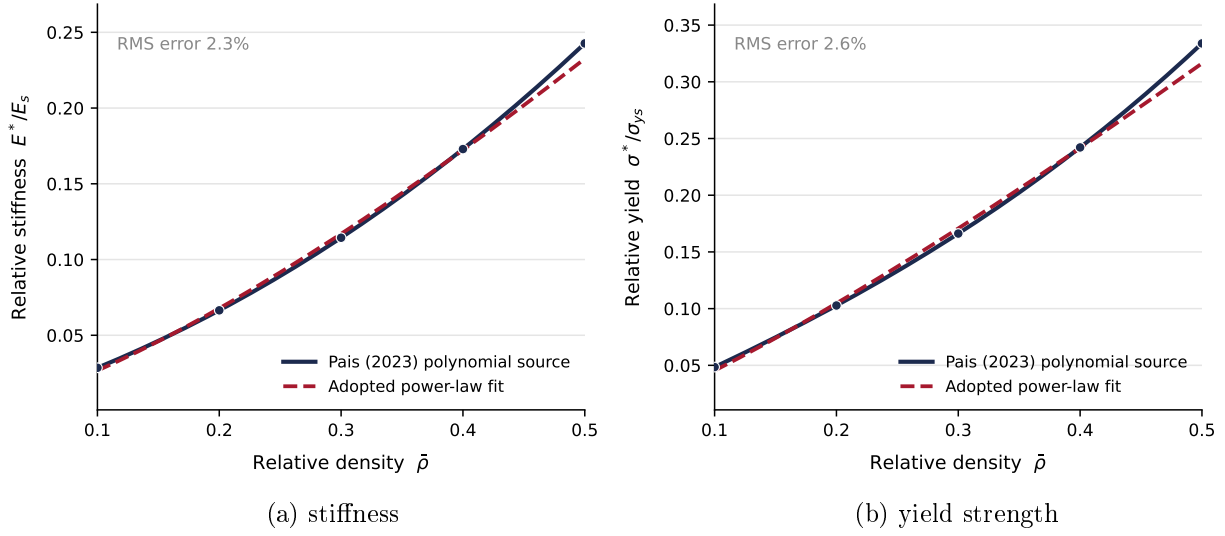


Figure 4.10: The gyroid-sheet coefficients are obtained by regressing a single Gibson–Ashby power law (dashed) onto the cubic-polynomial homogenisation source of Pais et al. [33] (solid) over the working density band. The fit reproduces the source to within about 2.5% root-mean-square error for both the relative stiffness (a) and the relative yield (b), with the largest error at the band edges. Markers denote the sampled densities.

The Neovius row is assembled from two studies rather than one, because no single homogenisation study found in the literature reports both its stiffness and its strength: the stiffness coefficients are taken from Abueidda et al. [103] and the strength coefficients from Ravichander et al. [104]. Since the two halves of the row do not share a common provenance, it is the least internally consistent of the six and is treated as indicative rather than settled. The gyroid sheet and the Neovius rows are both surfaced to the user with an explicit caveat, so that the choice of a re-fitted or a mixed-source morphology is never made unknowingly.

4.5.1.1 Property models: Safe and Experimental

A Gibson–Ashby power law is an empirical fit, and the way it is used away from the regime in which it was fitted carries real consequences for the result. The exponents are obtained over a bounded, low-to-moderate density range, and the law is not constrained to recover the parent solid at full density: extrapolated to $\bar{\rho} = 1$ a fitted power law generally does not return unity. Recent experimental and numerical work on TPMS lattices across a wide density range confirms both points, showing that the classical scaling neither captures the modulus and yield response over the full range nor extrapolates correctly to the fully dense solid [105]. The physical cause is a transition from structure-dominated behaviour at low density, where the cellular architecture governs and the power law holds, to material-dominated behaviour at high density, where the cell increasingly resembles the bulk and the true properties climb toward the solid faster than the power law predicts.

The engine accommodates this through two selectable property models, which apply the same coefficient row differently as relative density rises: they agree closely at low density and diverge toward full density.

In the *Safe* model the power law is trusted only below a density cap $\bar{\rho}_c$, set to one half, near the upper bound of the low-to-moderate density range over which the power-law fits are reported and beyond which the scaling is known to depart from the true response [105]; any element denser than the cap is treated as fully solid:

$$\frac{E^*}{E_s} = \begin{cases} C_1 \bar{\rho}^{n_E}, & \bar{\rho} < \bar{\rho}_c, \\ 1, & \bar{\rho} \geq \bar{\rho}_c, \end{cases} \quad \frac{\sigma^*}{\sigma_{ys}} = \begin{cases} C_2 \bar{\rho}^{n_\sigma}, & \bar{\rho} < \bar{\rho}_c, \\ 1, & \bar{\rho} \geq \bar{\rho}_c, \end{cases} \quad (4.13)$$

where $\bar{\rho}_c$ is the density cap. The graded lattice band in this model runs from the void threshold up to just below the cap, and any element the optimiser would drive denser than the cap is promoted to solid rather than represented by an extrapolated lattice law. The Safe model guarantees that no reported property rests on the power law outside the range in which it was fitted, which makes it the defensible default for quantitative results; its cost is that the mid-to-high density band is unavailable as graded lattice, those regions becoming solid instead.

The *Experimental* model extends the graded band to full density by blending the power law into the exact solid at $\bar{\rho} = 1$. Each coefficient is raised to the power $(1 - \bar{\rho})$,

$$\frac{E^*}{E_s} = C_1^{1-\bar{\rho}} \bar{\rho}^{n_E}, \quad \frac{\sigma^*}{\sigma_{ys}} = C_2^{1-\bar{\rho}} \bar{\rho}^{n_\sigma}, \quad (4.14)$$

so that at the low-density end the exponent $(1 - \bar{\rho})$ approaches one and the expression stays close to the Gibson–Ashby fit, while as density rises the factor $C^{1-\bar{\rho}}$ lifts it progressively above the bare power law, until at $\bar{\rho} = 1$ it becomes $C^0 \cdot 1 = 1$ for both properties and recovers the parent solid exactly. The discontinuity that a bare power law would leave at full density is thereby removed, the upper end of the graded band moves from the cap to full density, and the optimiser can grade material all the way to solid rather than saturating at the cap. The two models are compared for a ligament and a sheet morphology in Figure 4.11: the Safe law follows the fitted curve to the cap and then steps to the solid, whereas the Experimental law blends smoothly to the solid at unit density.

It should be stated plainly that the Experimental blend is a smooth, endpoint-correcting interpolation and not a micromechanically derived constitutive law, so it is not the most rigorous route to the high-density response. The rigorous treatments are the bound- and mean-field homogenisation models, such as those anchored on the Hashin–Shtrikman variational bounds [106] or built from Mori–Tanaka mean-field theory [107], the latter resting on Eshelby’s inclusion solution [108]; all of these approach the bulk at full density from first principles. The Experimental model is offered as a fast and defensible estimate of the true capacity of a fully graded design, bypassing the loss of validity of the Gibson–Ashby

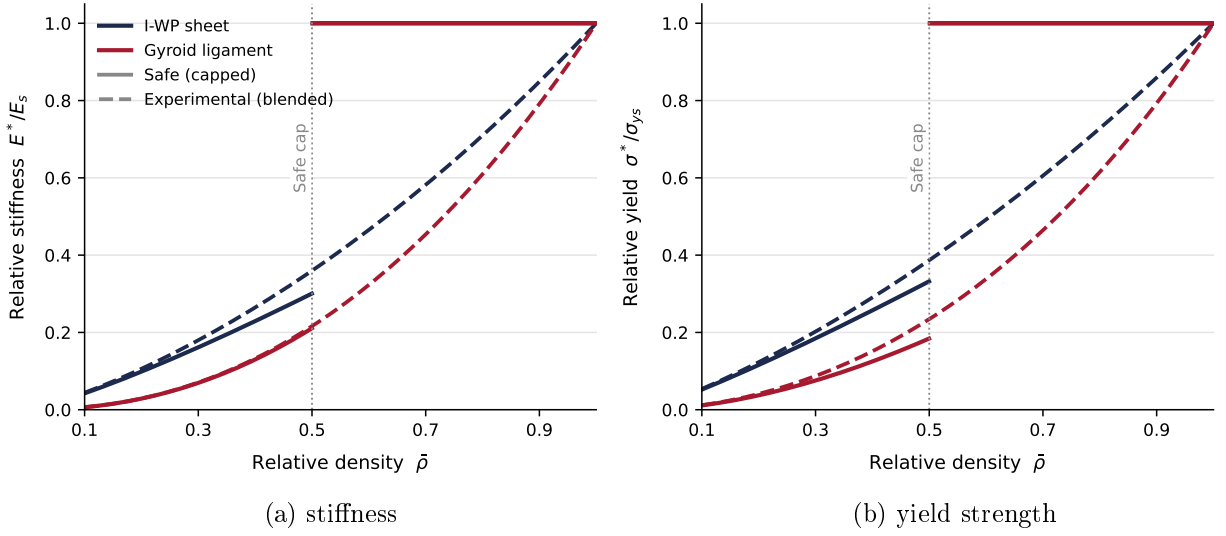


Figure 4.11: The Safe (solid) and Experimental (dashed) property models, shown for the I-WP sheet and the gyroid ligament. The Safe model trusts the Gibson–Ashby law only below the density cap and treats denser material as solid, leaving a step at the cap; the Experimental model blends the law so that it recovers the parent solid exactly at full density. Both the relative stiffness (a) and the relative yield (b) are shown.

law beyond roughly half density, rather than as a substitute for those methods. Any result obtained in the Experimental model is labelled as such throughout this work, and a result that also relies on one of the two flagged morphology rows combines two of the engine’s weaker assumptions and should be read with corresponding caution.

4.5.2 Utilisation and the fully-stressed target

The solid-void engine reads strain energy from each solved design; the lattice engine reads stress. At every iteration the field that drives the update is the von Mises stress of each design element, taken as the largest value across the element’s integration points. When more than one load case is present, the per-case stresses are combined into a single field by the weighted sum of Equation (4.1), and that field is then regularised by the same spatial filter and history average as the shared core, Equations (4.2) and (4.4), so that the stress driving the lattice update is mesh-independent and stable from iteration to iteration exactly as the solid-void sensitivity is. The filtered and stabilised stress of element e is written $\bar{\sigma}_e$.

What the engine does with this stress is the point of divergence. Rather than rank elements by how much they contribute to stiffness, it asks how hard each element is working relative to the strength its current density provides. The local yield strength of an element is the effective yield of its morphology evaluated at its current relative density, under the active property model,

$$\sigma_y(\bar{\rho}_e) = \sigma_{ys} C_2 \bar{\rho}_e^{n_\sigma}, \quad (4.15)$$

where $\sigma_y(\bar{\rho}_e)$ is the local yield strength of element e at relative density $\bar{\rho}_e$, written here for the Safe model; in the Experimental model the capped law is replaced by the blend of Equation (4.14). The *utilisation* of the element is then the stress it carries divided by that local capacity,

$$\mathcal{U}_e = \frac{\bar{\sigma}_e}{\sigma_y(\bar{\rho}_e)}, \quad (4.16)$$

where $\mathcal{U}_e$ is the utilisation. An element with $\mathcal{U}_e$ close to one is loaded near its capacity and efficiently used; an element with $\mathcal{U}_e$ well below one is over-built, carrying far less stress than its density could support; an element with $\mathcal{U}_e$ above one is overloaded. Utilisation, not strain energy, is the measure that orders the design domain for this engine.

The aim of the update is a fully-stressed design, a classical sizing principle in which each member is proportioned so that it just reaches its allowable stress under load [109]. For the graded lattice this means driving each element toward the relative density at which its local yield would exactly equal the stress it carries, so that its utilisation would be one. Inverting the strength law of Equation (4.15) gives this fully-stressed target density,

$$\bar{\rho}_e^* = \left(\frac{s_L \bar{\sigma}_e}{\sigma_{ys} C_2} \right)^{1/n_\sigma}, \quad (4.17)$$

where $\bar{\rho}_e^*$ is the target relative density of element e and s_L is a sizing factor set by the load mode of Section 4.5.2.1. Equation (4.17) is the closed-form inverse of the Safe strength law; in the Experimental model the blended law of Equation (4.14) is not closed-form invertible, and the target is found numerically, the law being monotonic in density. An element more highly stressed than its current density warrants has a target above its present value and should thicken; an element carrying little stress has a target below its present value and should thin or be removed.

One property of the fully-stressed target should be stated, since it shapes how the converged result is read. Each element is sized against the stress field of the current design, as though its own change of density left that field untouched. In reality, changing an element's density redistributes load, because a stiffer element draws more of the surrounding stress into itself, so the field the next iteration solves is not the one each element sized against. Each element is therefore sized against a stress field that its own change of density then alters, so the sizing always trails the equilibrium it targets, and the most highly stressed elements settle at a utilisation slightly above unity rather than exactly at it. The engine reports the utilisation field of the converged design directly (Section 4.7), so this small and characteristic excess over unity is shown rather than hidden.

4.5.2.1 Load modes: Scaled and Sized-to-Load

The sizing factor s_L of Equation (4.17) is where the engine answers one of two distinct engineering questions, selected by the load mode.

In the *Scaled* mode the stress field is normalised so that the single most-stressed element

sits exactly at yield, by setting $s_L = \sigma_{ys}/\sigma_{\max}$, where $\sigma_{\max}$ is the peak design stress, with no further margin. Every other element is then sized relative to that peak. The result is the *shape* of the optimal load path, the best distribution of relative density for the given load pattern, independent of the absolute magnitude of the load. Because the field is rescaled to the material's yield, a Scaled result is agnostic both to the magnitude of the applied load and to the absolute strength of the material: it answers where material should go, not how much load the part can bear. It is the natural choice when the load pattern is known but its absolute level is uncertain.

In the *Sized-to-Load* mode the elements are sized against the real stress magnitudes returned by the analysis, with no rescaling of the field; here s_L is a safety factor, which is unity by default and may be raised to size in reserve. Each element then targets the density that genuinely carries its actual stress multiplied by the chosen margin, so the converged lattice is not merely a shape but a physically dimensioned component, graded to survive the real load at the prescribed factor of safety in the chosen material. This is the mode a user reaches for when sizing a real part, since the result is aware both of the load it must carry and of the strength of the material it is made from. In this mode the engine additionally writes a feasibility check on the converged design, classifying it from the stress it carries under the real load as feasible if no material yields, locally exceeded if only a small boundary fraction passes yield, which is typically an artefact of a load or support pad rather than a structural deficiency, or overloaded if yielding is widespread.

4.5.3 The density update and convergence

The engine begins from the full solid block, every design element at unit relative density, and progressively resolves it into a graded field. At each iteration it ranks the design elements by their utilisation, Equation (4.16), from the lowest, the safest and most over-built, to the highest, the closest to its capacity. The non-design elements of Section 4.2 are excluded from the ranking and held solid throughout. This ranking is what the lattice engine substitutes for the strain-energy re-ranking of the solid-void engine, and as there the order is recomputed from the current solution at every iteration, so that no element's density is ever permanent.

The block is not allowed to rearrange itself all at once. Only a fixed fraction of the design elements, the evolution quota, may change state in a single iteration: the lowest-utilisation elements that are not already at the void threshold are the candidates to thin, and the highest-utilisation elements that are not already solid are the candidates to thicken. Drawing the candidates only from elements that can actually move in the intended direction stops the search both from collapsing the structure in a single step and from stalling, and it is what lets the solid block come apart gradually and stably as load redistributes between successive solves.

Each element in the quota then steps from its current density toward its fully-stressed

target of Equation (4.17), but by no more than a move limit Δ in a single iteration,

$$\bar{\rho}_e^{(t)} = \begin{cases} \max(\bar{\rho}_e^*, \bar{\rho}_e^{(t-1)} - \Delta), & \text{thinning,} \\ \min(\bar{\rho}_e^*, \bar{\rho}_e^{(t-1)} + \Delta), & \text{thickening,} \\ \bar{\rho}_e^{(t-1)}, & \text{frozen this iteration,} \end{cases} \quad (4.18)$$

where $\bar{\rho}_e^{(t)}$ is the updated relative density of element e , $\bar{\rho}_e^{(t-1)}$ its value at the previous iteration, $\bar{\rho}_e^*$ its fully-stressed target, and Δ the move limit. Over many iterations each element thus chases its own fully-stressed state, over-built regions thinning toward lattice or void and under-built regions thickening toward solid, and the design self-organises into a graded density field that mirrors the load path. A smaller move limit carves the same path in finer steps; a smaller quota carves fewer elements at each step.

Two thresholds close the update. An element whose density falls to or below the void threshold, one tenth in this work, is removed and assigned a near-zero modulus, so that it carries no load but keeps the mesh conformal and the global stiffness matrix non-singular, the same soft-kill device used by the shared core in Section 4.1. An element at or above the solid threshold, the density cap in the Safe model or unity in the Experimental model, is treated as fully solid. Everything between is graded lattice, its relative density quantised into one-per-cent density bins, with the lowest bin pinned 1% above the void threshold so that the thinnest lattice is always cleanly separated from removed material. Each bin is assigned the effective modulus its density implies, $E_s C_1 \bar{\rho}^{n_E}$ under the active property model, and the analysis of the next iteration is run with these per-bin moduli. Because the mesh never changes and only material properties are reassigned, each iteration costs a single linear solve on a fixed model, which is what makes the graded search affordable.

The solid-void engine stops when its prescribed volume fraction is reached. The lattice engine has no such target: it is a search for the most mass-efficient graded design, and the amount of material it retains is an outcome of that search rather than an input to it. After a warm-up period it is therefore halted by whichever of three conditions is met first. The first is convergence, when the specific stiffness of Equation (4.5) has changed by less than a set tolerance over the last several iterations and the design has settled. The second is disintegration, when the specific stiffness has fallen for several consecutive iterations, a sign that the search has passed its optimum and is beginning to destroy the structure. The third is stagnation, when the search has gone a set number of iterations without improving on its best design. By default the design carried forward is not the final iteration but the one of highest specific stiffness recorded over the run, so that any post-peak degradation does not contaminate the result. The retained mass fraction ϕ that enters Equation (4.5) is, for a graded design, the density-weighted element count: each solid element contributes one, each void none, and each graded element its own relative density.

4.5.4 The morphological skin layer

The free surface of the converged lattice is porous, which is a poor surface both to manufacture and to load in fatigue. The engine can therefore wrap the design in an optional solid skin. When it is requested, the skin is applied once, as a post-process after the search has finished rather than at every iteration; sealing the surface only at the end lets the shell wrap exactly the material the search carried forward, and keeps the surface treatment from interfering with the search itself. The skin hardens to solid every lattice element on an exposed surface and then grows inward through the material for the chosen number of element layers, producing a uniform solid shell of that thickness around the optimised core. Which surfaces count as exposed depends on the dimensionality of the model, which the engine detects automatically: in three dimensions the outer boundary and the walls of any air-connected cavity are sealed, while internal voids fully enclosed by lattice are left unsealed, both because they are not exposed surfaces and because sealing them would enclose unfused powder with no route for its removal, adding mass to no structural benefit; in a planar model, where an internal void is physically a through-thickness hole, every void wall is treated as an external surface and sealed.

The skin gives the part a solid, manufacturable, fatigue-resistant outer surface in place of a stress-raising porous one, and because it wraps only the optimised geometry it adds the least material needed to close that surface, which keeps the specific stiffness of the finished part high. This closed shell around a graded lattice core is the realisation, at the resolution of the analysis model, of the skinned functionally graded design anticipated in Section 3.4 (Figure 3.11). When the skin is applied, a single final analysis is run on the skinned design, so that its true specific stiffness and mass fraction are reported alongside the pre-skin optimum rather than estimated, and that skinned design becomes the one passed forward. The realisation of this density field as manufacturable geometry is the subject of Section 4.6.

4.6 Post-Processing and STL Generation

Both engines return an abstract result rather than a manufacturable body. The solid-void engine returns an explicit set of retained elements; the lattice engine returns a field of relative densities. Neither is a surface, and the two are different enough in kind that they are realised by two separate programs, each consuming the output its engine produces through the export stage of Section 4.2. This section describes both. The solid-void path (Section 4.6.1) extracts and selectively smooths the boundary of the retained solid; the lattice path (Section 4.6.2) realises the density field as explicit triply periodic minimal surface (TPMS) geometry, the de-homogenisation step anticipated in Section 3.4, carried here in the density dimension alone.

4.6.1 Surface extraction of the solid-void result

The solid-void engine exports its result as a list of retained solid elements, together with the mesh nodes and the non-design set of Section 4.2. The boundary of that solid set, the faces shared between a retained element and either a removed element or the exterior, is the surface of the part. As a set of element faces, however, it is a staircased voxel surface (Figure 4.12a): every face inclined or curved relative to the mesh axes is reproduced as a flight of steps. Smoothing this staircase into a print-ready surface is the generator’s principal post-processing option, applied at the user’s choice rather than automatically. When it is applied, it must take off these steps without disturbing the surfaces that have to stay exactly where the engine left them.

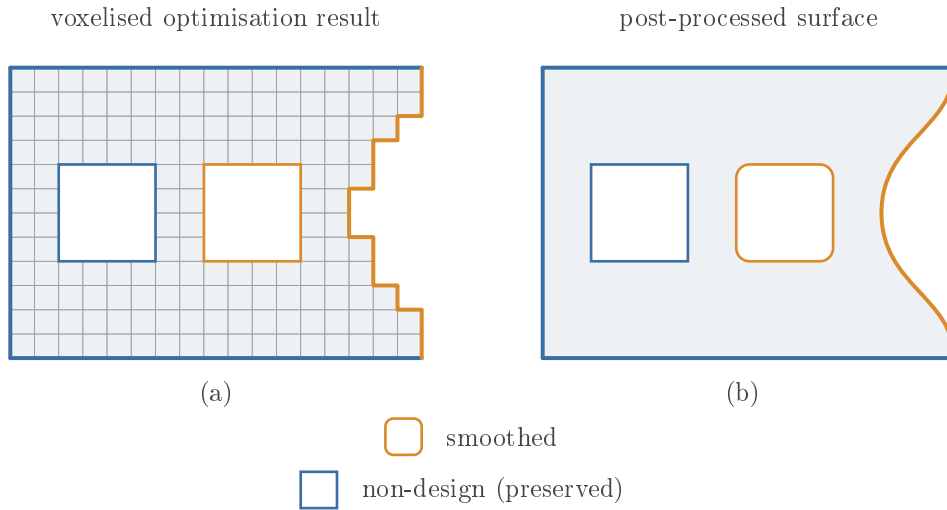


Figure 4.12: Post-processing of the solid-void boundary, shown on a cross-section. (a) The optimisation result discretised on the element grid: the optimisation-created cut is staircased and the square holes are cell-aligned. (b) When smoothing is applied (yellow), the cut is smoothed and the corners of the unflagged square hole are rounded, whereas every surface in the non-design set (navy: the three straight outer edges and the flagged square hole) is preserved exactly, keeping its sharp corners and flat faces.

The distinction the generator draws is between surfaces that the optimisation created and surfaces that were always part of the component. Only the latter must be preserved, since they carry the dimensions and interfaces the finished part must respect, whatever the part may be. This preservation is gated entirely on the non-design set, the regions the user marked as protected. A boundary face is eligible to be locked only if the element that owns it belongs to that set; a face owned by a design element is always smoothed. Among the non-design faces, the generator distinguishes, by default, those that are original exterior surfaces, with no removed design element behind them, which are locked, from those that became exposed only because the design material behind them was carved away, which are treated as cut surfaces and smoothed with the rest. Because this rule depends on the non-design set being supplied, the preservation is conditional on that input: if it is absent

the generator locks nothing and smooths the entire surface, and it warns accordingly.

Every surface the rule does not lock is then smoothed by a shrink-free band-pass filter operating on the surface signal [110], which removes the voxel staircase without the systematic volume loss that plain Laplacian smoothing would cause. The locked surfaces are not simply frozen in place, since a hard freeze would leave the staircased seam where a locked face meets a smoothed one, and would preserve any voxel bumps on the locked face itself. Instead, the generator classifies each locked vertex by the orientations of the locked faces that meet at it. A vertex whose surrounding locked faces are nearly coplanar lies on a single flat patch, and may slide within that plane, to straighten the serrated lock boundary, and move a capped distance normal to it, to flatten residual voxel bumps, while staying on the plane. A vertex whose surrounding locked faces point in noticeably different directions does not lie on a single flat patch, because it sits on an edge between differently oriented faces or on a curved surface, and is frozen exactly: allowing it to move would round that edge or distort that curve, losing the feature the lock exists to keep. The result keeps real mating faces flat and bores round while still cleaning the boundary between preserved and smoothed regions. A dimensional report closes the stage, comparing the smoothed surface against the original so that any drift of the preserved features is reported rather than assumed absent.

4.6.2 Geometry generation for the lattice result

The lattice engine exports a density map: for every element, its centre and a relative density between zero and one. This is a homogenised field, the effective solid fraction the optimiser chose for each region, and it carries no geometry. Turning it into an explicit, buildable microstructure is the de-homogenisation step described in Section 3.4, which projects a homogenised property field onto a connected, manufacturable lattice [49–51]. The realisation here is a density-graded one: it grades a chosen TPMS by solid fraction, element by element, while leaving the cell orientation fixed to the global frame rather than aligning it to the local principal directions.

Each morphology is the level set of an implicit scalar field assembled from the trigonometric nodal-surface approximations of the minimal surfaces [111],

$$\begin{aligned}
 F_G &= \sin \xi \cos \eta + \sin \eta \cos \zeta + \sin \zeta \cos \xi, \\
 F_P &= \cos \xi + \cos \eta + \cos \zeta, \\
 F_D &= \sin \xi \sin \eta \sin \zeta + \sin \xi \cos \eta \cos \zeta + \cos \xi \sin \eta \cos \zeta + \cos \xi \cos \eta \sin \zeta, \\
 F_N &= 3(\cos \xi + \cos \eta + \cos \zeta) + 4 \cos \xi \cos \eta \cos \zeta,
 \end{aligned} \tag{4.19}$$

where $\xi = \omega x$, $\eta = \omega y$, $\zeta = \omega z$, and $\omega = 2\pi/\ell$ is the cell angular frequency for a unit-cell period ℓ . The gyroid field F_G serves both the gyroid ligament and the gyroid sheet; F_P , F_D , F_{IWP} and F_N are the Schwarz primitive, diamond, I-WP and Neovius fields. The two structural families of Section 3.4 are distinguished only by which metric the threshold acts

on: the solid phase is the sub-level set $\{F < t_{\text{iso}}\}$ for the ligament morphologies, giving a connected network of struts, and the shell $\{|F| < t_{\text{iso}}\}$ for the sheet morphologies, giving a thin double wall about the zero level set. In both cases a single scalar threshold t_{iso} sets the solid fraction, and a larger t_{iso} thickens the struts or walls.

The map from relative density to threshold is where geometric fidelity is won or lost, and it is made exact rather than fitted. The volume fraction produced by a threshold t_{iso} is, by construction, the fraction of the cell over which the field metric satisfies the solid condition, that is the cumulative distribution of the metric m (F for ligaments, $|F|$ for sheets). The threshold that realises a target relative density $\bar{\rho}$ is therefore simply its quantile,

$$t_{\text{iso}}(\bar{\rho}) = Q_m(\bar{\rho}), \quad (4.20)$$

where Q_m is the quantile (inverse cumulative distribution) of the metric. Because the metric distribution depends only on the morphology and not on the period or the grid, it is sampled once per morphology over a single unit cell, and Equation (4.20) then applies element by element. The mapping carries no tunable constants; its only error is the small discretisation error of estimating the metric's distribution on a finite grid rather than continuously. This is why the realised volume fraction of the finished mesh, reported at the end as a self-check, matches the requested mean density to within a fraction of a percent.

With the threshold field fixed, the TPMS field is evaluated on a grid fine enough to resolve it, the two are combined into a signed implicit field whose zero level set is the lattice surface, and the surface is extracted by marching cubes [112]. To keep peak memory bounded for a component-sized part at fine resolution, the field is evaluated and meshed in overlapping slabs along one axis, with a one-slice overlap so that the surface is continuous across slab boundaries. The extracted mesh is optionally decimated, checked for manifoldness, and written as a binary STL. Unlike the solid-void generator, the lattice generator has no surface-preservation mechanism, no equivalent of the non-design locking of Section 4.6.1. Regions the engine held solid, the non-design set of Section 4.2, are carried at unit density and so are realised as solid rather than as lattice, and are therefore not turned into porous structure; what the generator does not do is hold any designated surface exactly, since every surface, solid or lattice, is reconstructed by the same isosurface meshing. Adding surface-aware preservation to the lattice path is left as future work.

4.6.2.1 Selection of meshing resolution

How finely the implicit field is sampled is the central decision of the lattice realisation, and a fixed value will not serve. Marching cubes reconstructs the surface from samples on a regular grid: too coarse a grid returns a faceted or, worse, topologically wrong surface, while too fine a grid returns millions of redundant triangles at a steep cost in memory and file size. The faithful resolution is not constant. It depends on the morphology and on the local density, because what must be resolved is the thinnest feature present, and that

feature changes across the density range: a slender strut or wall at low density, and the thin void between solids at high density. A single global setting therefore either under-resolves the thin regions or wastes effort on the thick ones. The resolution is instead chosen from a law, derived once in a convergence study and built into the exporter, that gives the required sampling density for each morphology at each density.

The primary measure of fidelity is how far the reconstructed surface departs from the exact one. For a candidate mesh, the perpendicular distance from each facet to the analytic surface is estimated as $|m - t|/\|\nabla m\|$, and the 95th percentile of that distance, expressed as a percentage of the cell size, is the deviation score; normalising to the cell makes it independent of the physical period. The tolerance was set by validation rather than assumption. An initial criterion of one percent of the cell still left the exported lattices visibly polygonal, so the tolerance was tightened, by direct inspection of the rendered geometry, to 0.30%, the value at which the surfaces first reached a satisfactory smoothness and agreed with a fully resolved reference to within a few microns at a representative period.

A deviation criterion alone is necessary but not sufficient, because it is blind to connectivity. A thin sheet wall can be pierced by many sub-resolution holes while every facet that survives still lies close to the analytic surface, so the deviation score stays low and falsely reports convergence on a wall that is in fact perforated. This was observed on the Schwarz primitive sheet at low density and confirmed with the Euler characteristic of the mesh, a topological invariant: at the deviation-converged resolution the Euler number sits far from its true value, indicating hundreds of false holes, and only at a higher resolution does it settle onto its correct, stable plateau and the holes disappear. A second requirement was therefore added, that the resolution be high enough for the Euler number to reach that plateau, and the requirement adopted at each density is the larger of the deviation and topology values. The topology floor binds only for the sheet morphologies at low density, where the walls are thinnest, and is largest for the Neovius sheet; the ligament lattices are topologically stable already at their deviation-converged resolution and are unchanged.

The combined law is shown in Figure 4.13. The relationship the exporter applies is tabulated rather than analytical: the convergence study provides, for each morphology, the required resolution at a discrete set of densities, and at run time the exporter obtains the value for an arbitrary density by linear interpolation between the neighbouring tabulated points. In Figure 4.13 the markers are these tabulated values and the line is the interpolation between them, so the line passes through every marker by construction; it is an interpolation of the law, not a regression fitted to scattered data. The requirement is U-shaped for every morphology, with the minimum near the middle of the range for the ligament lattices and shifted towards lower density for the sheets, since the binding feature is the thin solid at low density and the thin void at high density.

The form of this tabulated law was validated against the mechanics of cellular solids. For the ligament lattices the binding feature size is expected to scale with the square

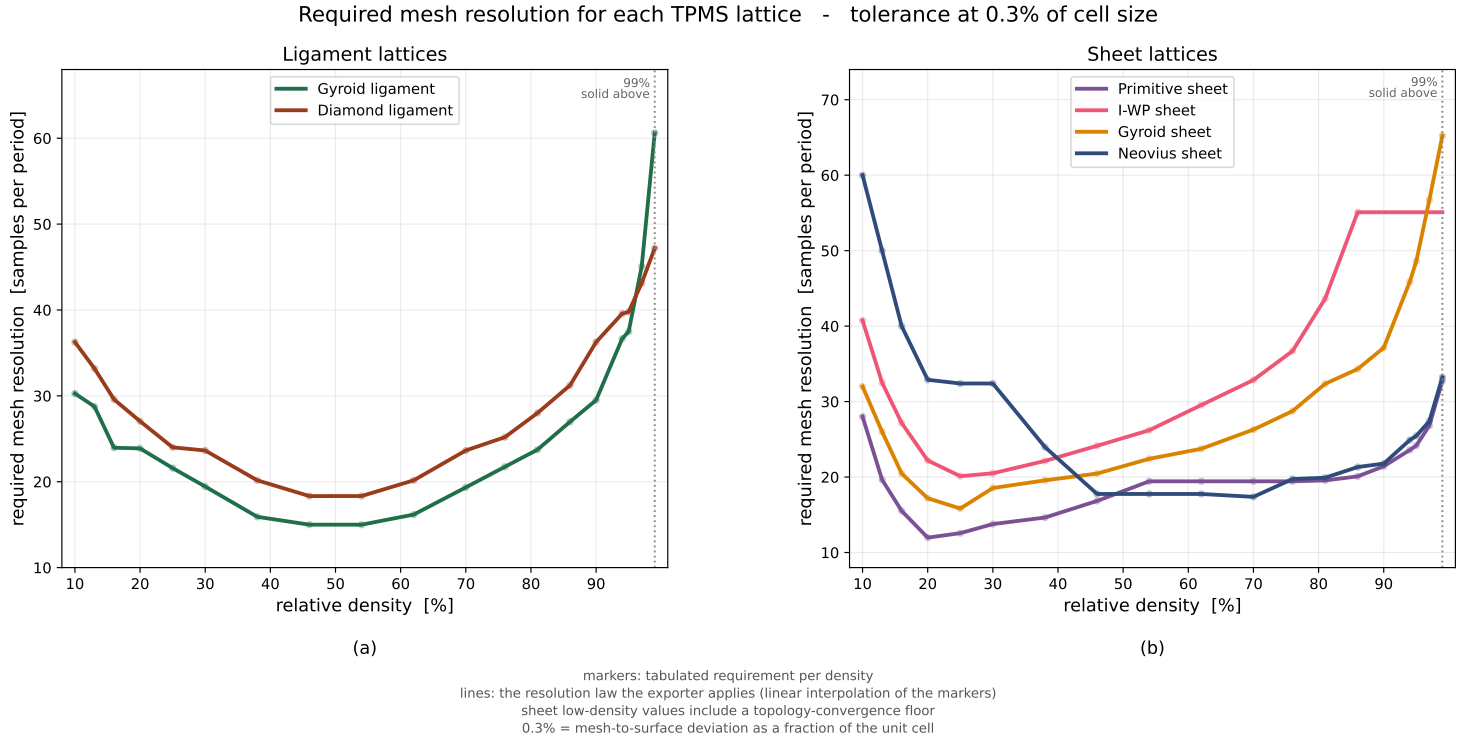


Figure 4.13: The marching-cubes resolution required to mesh each morphology faithfully, in samples per period, against relative density, for the ligament lattices (a) and the sheet lattices (b). Markers are the tabulated requirement from the convergence study at each studied density, taken as the larger of the deviation and topology criteria; the line is the piecewise-linear interpolation between those tabulated values that the exporter evaluates at run time, and it therefore passes through every marker by construction rather than fitting them. The requirement is U-shaped: it falls as the binding solid feature thickens and rises again as the binding void feature thins. The elevated low-density values of the sheet curves are the topology floor; the I-WP curve peaks near 86%, where its void pinches off into isolated pockets, and is held at that peak value for denser cells.

root of relative density, as Gibson and Ashby describe [95]; fitting a two-sided square-root model to the measured ligament requirement returns a coefficient of determination near 0.93. This agreement confirms that the requirement tracks genuine feature physics rather than a numerical artefact, and so justifies the shape of the law. The square-root model is used only as this physical check, however; it is not the relationship the exporter evaluates, which remains the tabulated requirement interpolated linearly as described above.

One sheet morphology, I-WP, behaves distinctively at high density. Around 86% relative density its void undergoes a topological transition: the connecting necks of the void channels collapse to a few tens of microns and then pinch off, so that the single connected void network breaks into isolated pockets, with a corresponding change in the Euler characteristic of the solid. Figure 4.14 shows this in cross-section: the void necks thin between 82% and 85% and have disconnected by 87%. These near-tangent necks are the finest feature anywhere in the study, so the resolution requirement peaks there; on either side

of the transition the requirement is significantly lower, and above the pinch the smooth isolated pockets resolve readily. The requirement is therefore a narrow peak, not an upper bound on resolvability. The tabulated law does not reproduce this narrow spike: its values are constrained to vary monotonically along each arm of the U-shaped curve, which by design removes single-density spikes, so the high-density value is held at an elevated level rather than dropping back above the pinch. This over-resolves the simple pockets above the pinch, which is why the I-WP cell in the 90% validation render (Figure 4.16) is reproduced cleanly.

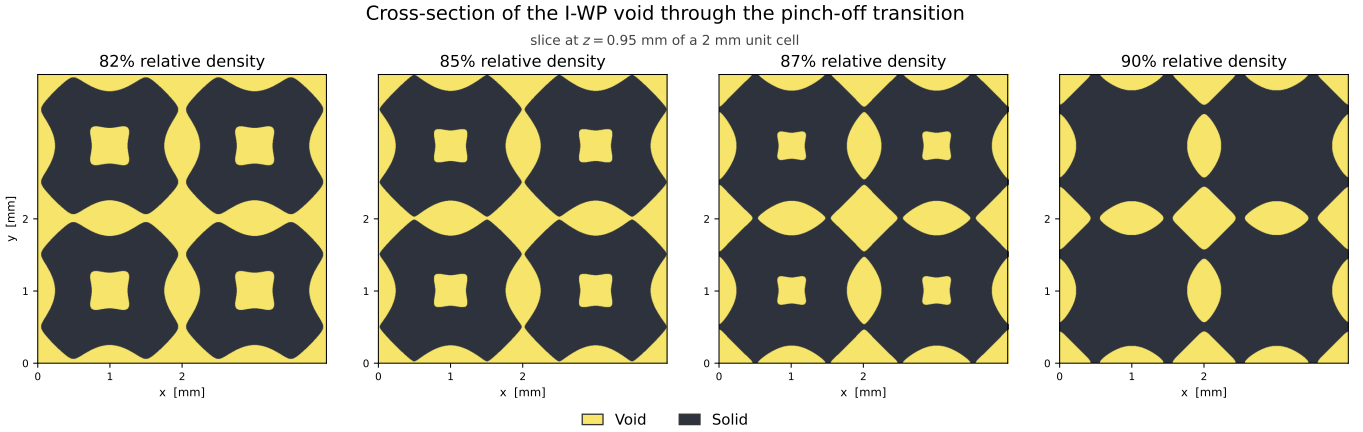


Figure 4.14: Cross-section of the I-WP unit cell (slice at $z = 0.95$ mm of a 2 mm cell) at four relative densities, with the void in yellow and the solid in dark grey. As the density rises the void’s connecting necks thin and, around 86%, pinch off, so the single connected void network of the lower densities breaks into the isolated pockets seen at 90%. These thin necks are the finest feature in the structure and set the peak meshing requirement for I-WP.

The high-density limitation for I-WP is not meshing but manufacturability. The collapsing necks are already far below the powder-bed minimum feature size, a few tens of microns at a two-millimetre cell, so a sufficiently dense I-WP element cannot be built whatever the mesh resolution. This case is therefore handled by the manufacturability check of Section 4.6.2.2, which reports the sub-limit feature and advises raising the element to unity, realising it as solid rather than as an all-but-solid lattice whose voids could be neither printed nor usefully retained.

The law is expressed in samples per period, a property of the morphology and density alone. The exporter converts it to a subgrid, the number of samples placed across one element, through the geometric relation

$$n_s = N_p \frac{h_e}{\ell}, \quad (4.21)$$

where n_s is the subgrid, N_p the required samples per period from the law, h_e the element size and ℓ the cell period. Because the requirement is U-shaped, the binding value for a

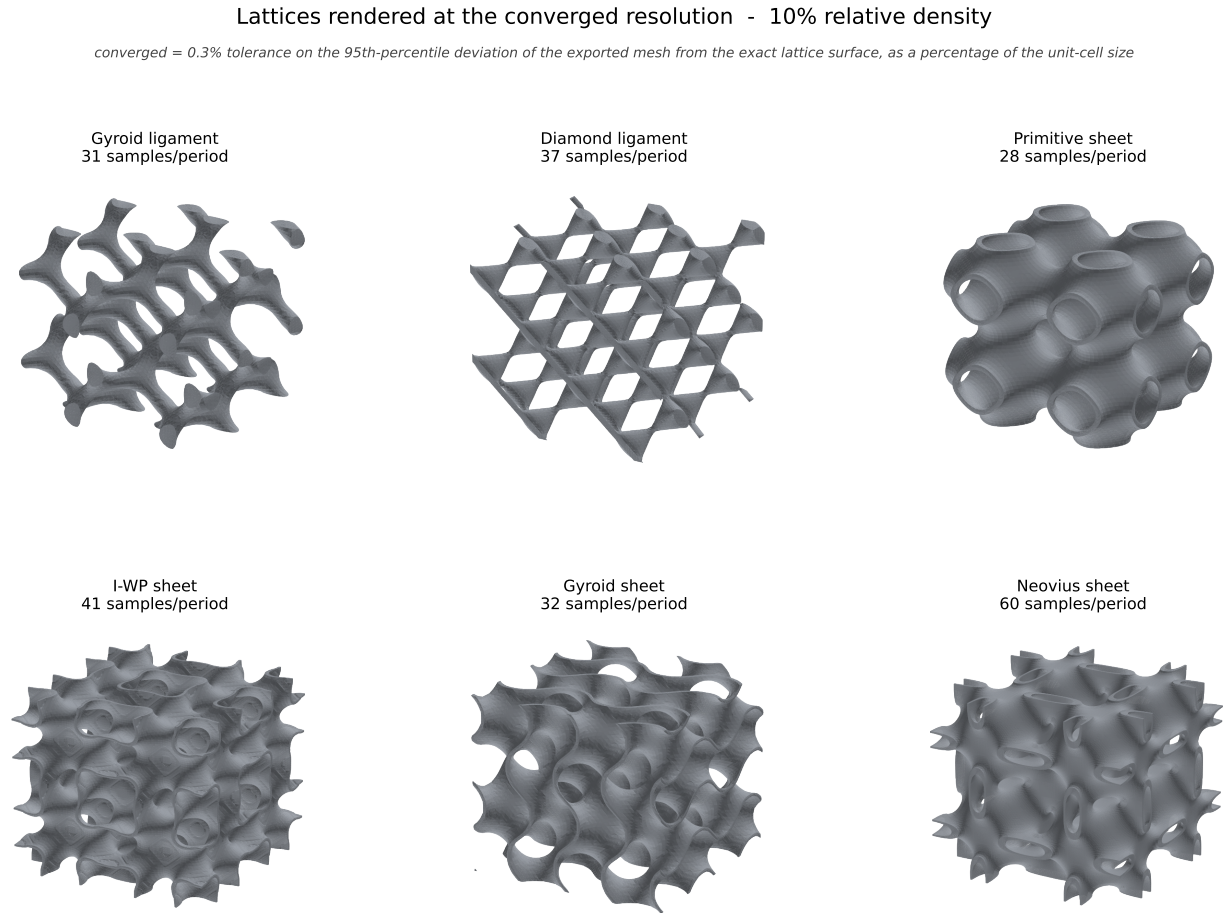


Figure 4.15: The six morphologies meshed at the resolution the law of Figure 4.13 prescribes, at 10% relative density, with the samples per period labelled on each panel. All of the morphologies render clean and pore-free at the topology-corrected resolution.

part spanning a range of densities lies at one of the two extremes, so the law is evaluated at the lowest and highest active densities and the larger result taken, guaranteeing that both the thinnest solid and the thinnest void in the part are resolved. Keeping the subgrid at the minimum the law requires is important because the meshing cost scales with the cube of the subgrid: the field is three-dimensional, so doubling the sampling multiplies the voxel count, triangle count, memory and run time by roughly eight. Modest-looking increases therefore become very expensive very quickly, which is why the exporter recommends the specific law value rather than simply advising that a finer grid is safer. The cell period itself is a user option. It defaults to the analysis element size, which is the value that reproduces the optimised density field exactly, one unit cell carrying one element's density. It may be overridden to a coarser period to manage cost, but at the expense of fidelity: a period larger than the element size makes a single cell span several elements, so the per-element density variation the optimiser produced can no longer be reproduced and is averaged across the

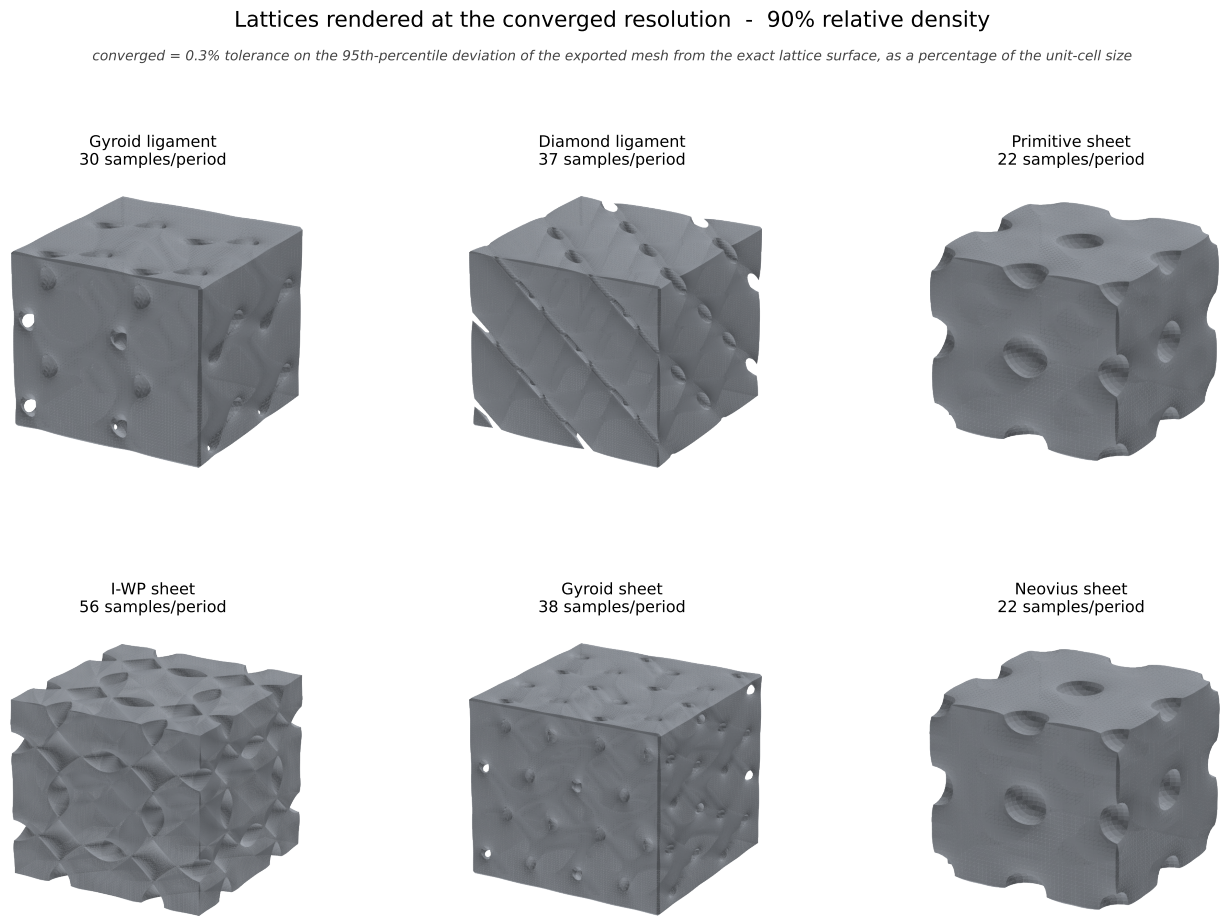


Figure 4.16: The six morphologies meshed at the prescribed resolution at 90% relative density. The near-solid structures, pierced by the thin void features that bind the resolution at high density, are reproduced cleanly and remain manifold.

cell. A period finer than the mesh, conversely, rarely repays its expense.

4.6.2.2 Manufacturability and smoothing

Independently of meshing, the exporter assesses whether the part can actually be built. It estimates the thinnest physical feature, the limiting strut diameter for a ligament morphology or the wall thickness for a sheet, from the solid region of the field, and compares it against the minimum manufacturable feature of the powder-bed process, which lies at a few tenths of a millimetre [45, 52]. A feature approaching the warning band, near a quarter of a millimetre, or below the hard floor, near 0.15 mm, is flagged, with the exporter suggesting raising the thinnest regions to a printable thickness by imposing a minimum-density floor.

Both assessments are surfaced before any geometry is generated. The exporter can run a pre-flight pass that evaluates the part and returns a structured report without meshing, which the launcher renders to the operator so that problems are caught before the lengthy

meshing step. The report carries the manufacturability assessment above, and a resolution assessment that compares the requested samples per element against the law and returns one of four graded statuses. *Critical* marks a request below half the required sampling, where thin features are heavily faceted and the sheet walls perforate; *warning* marks a request under the law but within a factor of two of it; *pass* marks a request that meets the law; and *overkill* marks a request finer than necessary, buying accuracy below the level the part requires while paying the full cube-law cost. Because the report is produced by the same law lookup the mesher applies, the value the launcher shows is exactly the value meshing will use, and for the void-limited regime of I-WP it advises a density cap rather than an unbounded increase in resolution.

A final option addresses the piecewise-constant nature of the optimised field. Because density is delivered per element, neighbouring elements can differ sharply, and without intervention the exported geometry then steps abruptly in feature thickness at every such boundary, which both raises stress concentrations and leaves visible seams in a load-bearing part. An optional density-blend smoothing removes these steps: a Gaussian blur is applied to the density field at subgrid resolution before it is mapped to thresholds, masked by the active material so that it mixes density only among non-void voxels and is renormalised by the local valid weight, so that density does not bleed into the surrounding void and the outer surfaces are preserved. An abrupt density discontinuity then becomes a continuous ramp, and the strut or wall thickness tapers smoothly across the boundary. The operation redistributes material rather than adding or removing it, so it is volume-conserving and the realised volume fraction is essentially unchanged. For a field that already varies gently the operation does little, by design, so it is left off by default and enabled when the optimised field carries sharp density transitions that warrant blending.

With the optimised result of either engine now realised as a manufacturable surface, the framework is complete. Section 4.7 verifies it against benchmark problems of known solution before it is applied to a real component.

4.7 Verification on Benchmark Cases

A framework assembled from many interacting parts, two engines on a shared core, a filter, a stabilised history and a solver in the loop, must be shown to work correctly before it is trusted with a real component. The standard used throughout this section is the distinction between verification and validation [113]: verification asks whether the framework solves its stated problem correctly, returning solutions that are known by other means, while validation asks whether that problem is itself a faithful model of physical reality. This section verifies. The validation of the framework against a physical component, through experimental and independent numerical comparison, belongs to the application that follows in the next chapter.

The two engines invite different verification, because they solve different problems. The

solid-void engine minimises compliance, a problem whose optimal layouts on canonical domains are long established, so it can be held against an external reference: its results are checked twice over, against the classical benchmark topologies the literature establishes, whose features the layout should reproduce, and, independently, against those a commercial optimiser produces on identical problems. The lattice engine solves a fully-stressed-design problem for which no canonical optimal layout is published, so it has no external answer to match and is instead verified by internal consistency, by confirming that its converged designs behave exactly as the fully-stressed principle of Section 4.5 requires. Section 4.7.1 sets out the finite-element and optimisation settings shared by every run and lists the runs themselves; Section 4.7.2 presents the solid-void benchmarks; Section 4.7.3 the lattice consistency check; and Section 4.7.4 a direct comparison of the two engines at equal mass.

4.7.1 Finite-element and optimisation settings

Every run in this section shares a common finite-element and optimisation configuration, so that any difference between results reflects the design problem and not the solver setup. The design domains are meshed with first-order reduced-integration elements at a one-millimetre size: the four-node plane-stress quadrilateral CPS4R in two dimensions and the eight-node hexahedron C3D8R in three. Reduced integration uses a single integration point per element, which makes each analysis inexpensive, a decisive economy when a full solve is performed at every iteration, and is well suited to the near-uniform strain states of a topology-optimisation domain; the accompanying hourglass control suppresses the spurious zero-energy modes that a single integration point would otherwise admit [114]. The sensitivity field of every run is regularised by the linear-kernel spatial filter of Section 4.1 at a radius of three millimetres, three element-lengths on this mesh. The filter is not an optional refinement but a necessary part of a convergent scheme: without it the solution checkerboards and depends on the mesh [14], and its radius governs the length scale of the finest members, as discussed in Section 4.1. Material is removed on the dynamic evolutionary schedule of Section 4.3, the ratio falling from 2% to 0.5% as the target fraction is approached, Equation (4.8), and a single load case drives each run. The parent material is the selective-laser-melting alloy AlSi10Mg, of modulus near seventy gigapascals and yield strength near two hundred and fifty megapascals.

Two independence properties of the solid-void engine bound what its benchmarks depend on, and are worth stating before the results. First, the solid-void topology is independent of both the elastic modulus and the magnitude of the applied load: the strain-energy ranking of Section 4.3 scales uniformly with each, so neither alters the order in which elements are ranked, and therefore neither alters the layout produced. The modulus and the load magnitude, with the material strength, enter only the stress-driven lattice engine. Second, in the two-dimensional runs the plane-stress thickness is left at unity, because the compliance objective also scales uniformly with thickness and so the thickness likewise leaves the optimised topology unchanged. The benchmarks below therefore test the layout

the engine finds, free of any dependence on these quantities.

Table 4.2 lists the runs used across this section. The solid-void benchmarks (A) are each solved to a prescribed volume fraction; the overhang showcase (B) repeats a full-beam build with the mitigation off, the build endpoint, and on at three thermal weights; the lattice runs (C) exercise the engine’s options on a common three-dimensional domain, their retained volume fraction an outcome of the search rather than an input to it (Section 4.5); and the equal-mass comparison (D) re-runs the solid-void engine at each lattice design’s own volume fraction.

Table 4.2: The runs used in the verification study. MBB denotes the Messerschmitt-Bolkow-Blohm beam, the standard simply supported beam benchmark of topology optimisation. All runs share the finite-element and optimisation settings of Section 4.7.1; the columns record only what distinguishes them. The lattice volume fractions are outcomes of the search, not targets.

Run	Case (size, mm)	VF	Engine and configuration
<i>Solid-void benchmarks (literature and Tosca cross-check)</i>			
A1	MBB beam, half (150×50)	0.50	solid-void
A2	Short cantilever (150×100)	0.40	solid-void
A3	Three-point bend ($120 \times 20 \times 20$)	0.30	solid-void
<i>Overhang mitigation showcase</i>			
B1	MBB beam, full (300×50)	0.40	solid-void, mitigation off (build endpoint)
B2	MBB beam, full (300×50)	0.40	solid-void, mitigation on, $t_w = 0.1$
B3	MBB beam, full (300×50)	0.40	solid-void, mitigation on, $t_w = 0.4$
B4	MBB beam, full (300×50)	0.40	solid-void, mitigation on, $t_w = 0.9$
<i>Lattice consistency</i>			
C1	Three-point bend ($120 \times 20 \times 20$)	–	Scaled + Safe, gyroid ligament
C2	Three-point bend ($120 \times 20 \times 20$)	–	Sized-to-Load + Safe, gyroid ligament
C3	Three-point bend ($120 \times 20 \times 20$)	–	Sized-to-Load + Experimental, gyroid ligament
C4	Three-point bend ($120 \times 20 \times 20$)	–	As C3, with morphological skin
C5	Three-point bend ($120 \times 20 \times 20$)	–	Sized-to-Load + Experimental, I-WP sheet
<i>Equal-mass dual-engine comparison</i>			
D1	Three-point bend ($120 \times 20 \times 20$)	0.670	solid-void, matched to C3
D2	Three-point bend ($120 \times 20 \times 20$)	0.462	solid-void, matched to C5

4.7.2 Solid-void benchmarks

The solid-void engine is verified on three problems of established solution: the Messerschmitt-Bolkow-Blohm (MBB) beam, a short cantilever, and a three-dimensional three-point-bending beam. Each is solved to the target volume fraction of its benchmark, and the design taken for comparison is the target-fraction design of Section 4.3, which is the one to set against a layout defined at a fixed volume fraction. Each result is then checked twice: against the canonical topology reported in the literature, and against the topology a commercial optimiser produces on an identical model.

The commercial comparison was produced with a condition-based topology optimisation in Tosca Structure, as distributed with Abaqus 2025, configured to minimise strain energy under a whole-model volume-fraction constraint set to each case's target of 0.50, 0.40 and 0.30. To match the freedom given to the present engine, the load and boundary-condition regions were left unfrozen in the MBB and cantilever runs, as they were for the present engine; in the three-dimensional run both optimisers held those regions frozen. No member-size restriction was imposed, so member scale was governed only by Tosca's internal filtering, and each optimisation ran for up to fifty design cycles.

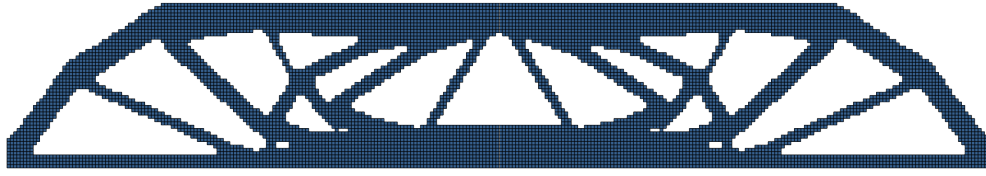
The two families of result are placed on a common footing before they are compared. The density-based output of Tosca is a continuous field with a grey transition between solid and void, and the smoothed iso-surface its post-processor draws by default is a manufacturable-surface approximation that would not compare cleanly with the discrete topology the present engine produces. The Tosca field is therefore shown as a whole-element quantity: the normalised element density is contoured on the undeformed mesh with the iso-surface cut disabled, and a display group retains only elements above one-half relative density. This discards the intermediate grey material intrinsic to a density-based method and isolates the load-bearing layout on the original mesh, giving the voxelised representation seen in the figures, comparable element for element with the solid-void topology beside it.

On the MBB beam (Figure 4.17) the engine returns the classical result: a triangulated truss spanning a tension tie along the lower edge and a compression arch above, the textbook optimum for this problem [13]. The Tosca layout is the same in every essential, the same tie, the same arch, and a similar pattern of interior diagonal struts.

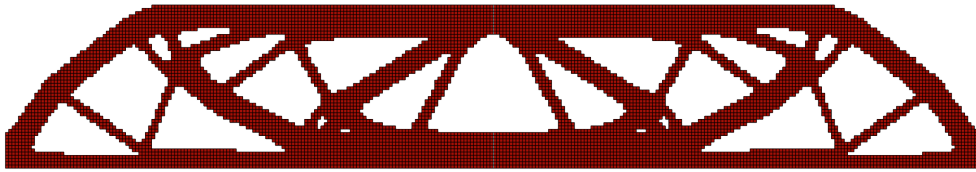
The short cantilever (Figure 4.18) likewise reproduces the established layout, a fan of members spreading from the encastred edge toward the loaded corner, and Tosca again agrees. This case also gives a concrete instance of a property stated in Section 4.3: that the design at the target volume fraction and the design of highest specific stiffness need not fall on the same iteration. Here they do not. The engine reaches its highest specific stiffness two iterations before the target, at a volume fraction of 0.409; the final increments of removal that bring the design down to 0.40 sever a slender member, and the stiffness falls with it, its absolute stiffness dropping from 1640.69 N/mm to 1610.57 N/mm. Panels (a) and (b) show the two designs, the target-fraction design carried into the benchmark comparison and the marginally fuller, slightly stiffer design the engine also reports; the difference between them is that single lost member, which is exactly the mechanism Section 4.3 anticipated for a low target fraction.

The three-dimensional three-point-bending beam (Figure 4.19) carries the check into three dimensions. The engine hollows the interior into a system of struts that carry the central load out to the two supports, and the Tosca result again matches it in layout.

Beyond the visual match the two optimisers can be compared quantitatively. Although the engine selects its designs by specific stiffness, Equation (4.5), the comparison here is



(a) present framework



(b) Tosca Structure

Figure 4.17: The MBB beam benchmark at a volume fraction of 0.50. (a) The present framework returns the classical triangulated truss, a tension tie along the lower edge beneath a compression arch. (b) Tosca Structure, thresholded to the same volume fraction, reproduces a similar layout.

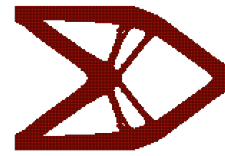
(a) framework, $f = 0.40$ (b) framework, $f = 0.409$ (c) Tosca, $f = 0.40$

Figure 4.18: The short cantilever benchmark at a volume fraction of 0.40. (a) The present framework at the target fraction, the design used for comparison. (b) The framework at its highest specific stiffness, two iterations earlier at a fraction of 0.409; the two differ by a single slender member lost in reaching the target, the distinction of Section 4.3 made visible. (c) Tosca Structure at the target fraction, reproducing a similar layout, with less but thicker struts.

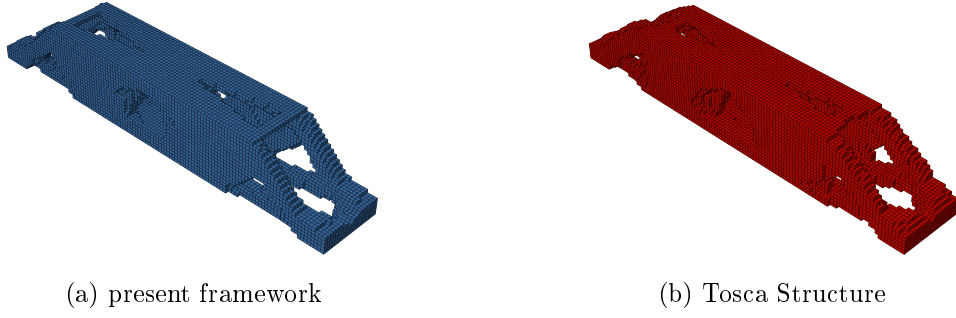


Figure 4.19: The three-dimensional three-point-bending benchmark at a volume fraction of 0.30. (a) The present framework and (b) Tosca Structure, thresholded to the same volume fraction, return the same strut system carrying the central load to the supports.

made on the absolute stiffness, the applied load divided by the resulting displacement. Because each benchmark pair is solved to the same volume fraction, and therefore to the same mass, specific and absolute stiffness differ only by that shared mass factor, so comparing the absolute values is equivalent to comparing the specific ones and is the more direct measure when the mass is fixed. Table 4.3 collects the results. In every case the present engine reaches a slightly higher absolute stiffness than the commercial tool at the same volume fraction. The margin is small and is consistent with the difference in representation: the discrete solid-void update resolves the load path into a larger number of thinner members, where the density-based method, through its internal filtering, forms fewer and broader ones.

Table 4.3: Absolute stiffness, the applied load per unit displacement (N mm^{-1}), of the present framework and of Tosca Structure at each benchmark’s target volume fraction f . Because each pair shares a volume fraction, this comparison is equivalent to one on specific stiffness. The present framework reaches a slightly higher stiffness in every case.

Benchmark	Present framework	Tosca Structure
MBB beam ($f = 0.50$)	342.00	341.53
Short cantilever ($f = 0.40$)	1610.57	1605.65
Three-point bend ($f = 0.30$)	12975.18	12581.78

Across the three problems the solid-void engine reproduces the established optimal topologies and matches, and slightly exceeds in stiffness, an independent commercial optimiser. This verifies the engine, and with it the shared core of Section 4.1 on which both engines are built.

4.7.2.1 Qualitative overhang-mitigation results

The overhang-mitigation feature of Section 4.4 was characterised quantitatively in that section, on this same full-span MBB beam, across a sweep of the thermal weight t_w from 0.1 to 0.9 (Figure 4.6). What that study reported as trade-off curves, this section shows as

the designs themselves: the build endpoint the refine phase starts from, and the mitigated results it produces at three representative weights, $t_w = 0.1, 0.4$ and 0.9 , each coloured by the thermal proxy field that drives the mitigation. These are the full-beam overhang runs of Table 4.2. All four designs share the build phase and the target volume fraction of 0.40, and only the thermal weight of the refine phase differs between the three mitigated runs, so the comparison isolates its effect.

The field shown is the steady-state thermal proxy of Section 4.4.1, the temperature labelled NT11 in the figures. As established there, it is a surrogate for how readily each region sheds heat to the build plate along the build direction: a region with a supported column of material beneath it has a short path to the plate and stays cool, while an unsupported, overhanging region must detour heat sideways through the throttled transverse conductivity and so runs hot. The field is therefore a direct picture of where the design is hard to build. Each plot is auto-scaled to its own peak, so the colours read within a plot and the peak values between plots.

Figure 4.20 shows the four designs. At the build endpoint (a), before the mitigation has acted, a large span of the upper chord is hot and poorly supported, and the peak proxy temperature reaches about 527. As the thermal weight rises through (b), (c) and (d), the refine phase reshapes the structure toward printability: the diagonal members reorient toward the self-supporting angle, the unsupported spans shorten, and the hot region contracts and cools, its peak falling to about 475, 424 and 385. The reshaping mitigates poor support rather than forbidding it. The overhangs that remain at every weight are the flat horizontal members: to resist bending the optimiser drives material away from the neutral axis, forming the horizontal chords and bridges of the truss, and the downward-facing undersides of those members are flat overhangs that no reorientation can remove. The diagonals, whose orientation is free, are reshaped toward the self-supporting angle; the discretionary overhangs the higher weights remove lie at the junctions where those diagonals meet the flat horizontal undersides, where the mitigation promotes supporting elements and clears the overhang, largely by filleting the intersection corner. That removal is not free of side-effects. At the highest weight the second flat bridge from the left has grown an inverted, almost tear-drop, wedge (Figure 4.20d) that lowers the violator count because it exposes little unsupported area, yet is not manufacturable as it stands, hanging with no support beneath it. It reads as a transient of the search rather than a finished feature, since the hottest point of the whole structure sits at its lower tip and is already casting strong support toward the diagonal it is growing to meet, so a further refine iteration would close the gap; but it shows how the more aggressive weights raise the tendency toward such artefacts. With that qualification, the highest weight is the most self-supporting of the four.

Table 4.4 gives the numbers behind these pictures, restating for the three showcased weights the quantities already reported in Section 4.4.5, this time, their refine endpoint (last iteration) values, instead of the mean of the 10 final iterations, as they were previously

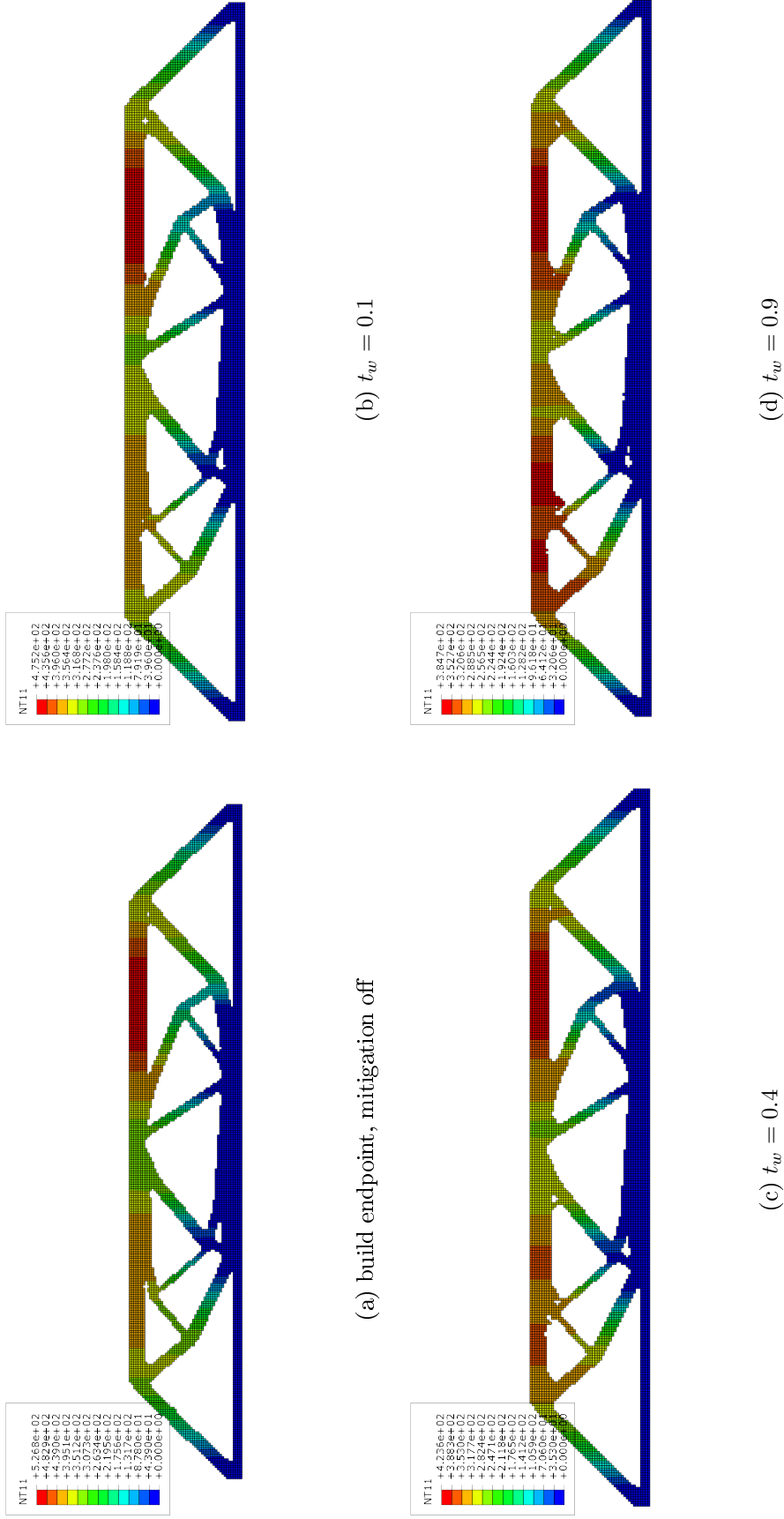


Figure 4.20: The overhang-mitigation showcase on the full MBB beam at a volume fraction of 0.40, each design coloured by the steady-state thermal proxy field NT11 of Section 4.4.1, the surrogate for poor heat evacuation that marks unsupported material. (a) The build endpoint, before mitigation, with a hot unsupported span along the upper chord. (b)–(d) The mitigated designs at rising thermal weight: the structure reshapes toward self-support and the hot region contracts and cools, the peak proxy temperature falling from about 527 to about 385. Each plot is auto-scaled to its own peak.

plotted. Overhang reduction is the fall in the active-violator count relative to the build endpoint, which carries 222 violators; it rises with the thermal weight, from about 11% at $t_w = 0.1$ to about 28% at $t_w = 0.9$, the saturating trend of Figure 4.6. The specific stiffness is not spent to buy this reduction: at every weight it stands above the build-endpoint value of 1397.64 (N/mm)/%, by around 3%, because the refine phase keeps redistributing material mechanically as it reshapes for printability, exactly as Section 4.4.5 found. The showcase therefore confirms, in the designs themselves, what the earlier study reported in aggregate: a substantial share of the overhangs removed at no cost in stiffness, and in fact at a small gain.

Table 4.4: The overhang-mitigation showcase on the full MBB beam at a volume fraction of 0.40, restating for the three showcased thermal weights the quantities characterised in Section 4.4.5. Overhang reduction is the fall in the active-violator count from the build endpoint. The specific stiffness stands above its build-endpoint value at every weight.

Design	Active violators	Overhang reduction (%)	Specific stiffness (N/mm)/%
Build endpoint	222	–	1397.64
$t_w = 0.1$	197	11.3	1442.37
$t_w = 0.4$	176	20.7	1446.13
$t_w = 0.9$	160	27.9	1436.57

4.7.3 Consistency of the lattice engine

The benchmarks of Section 4.7.2 could be checked against a known answer, because the compliance-optimal solid-void topology of each classical problem is established in the literature and reproduced by an independent commercial solver. The lattice engine admits no such check. There is no canonical optimal graded-density field for a given load: the result depends on the morphology, the property model and the load mode, and no published solution exists to compare against. The engine is therefore verified in a different sense, by internal consistency. Each of its principal options is exercised in turn on the common three-dimensional three-point-bending domain of Table 4.2, and the converged field is read against what the method of Section 4.5 predicts it should do. A result that responds to each option in the anticipated direction, and whose utilisation field carries the signature the fully-stressed formulation implies, is evidence that the engine behaves as designed. Five runs are used, labelled C1 to C5 in Table 4.2, each adding one change to the one before. All are driven by the same central load of twelve and a half kilonewtons against a parent yield of two hundred and fifty megapascals.

Before the individual runs, it is worth restating how the utilisation plots should be read, since the property that governs them was set out in Section 4.5.2 and recurs in every run. A perfectly fully-stressed design would show every design element at a utilisation of exactly one, each proportioned so that it just reaches the strength its density provides. The converged fields depart from that ideal in two ways, both anticipated there. The first

is a consequence of how the sizing is posed. Each element is driven toward the density that would just yield under the stress field of the current design, as though its own change of density left that field untouched; but changing one element's density redistributes load, so the field the next solve returns is not the one each element sized against. Every element's reference is therefore slightly wrong, and the accumulation of these small errors across the domain leaves the converged field a little off the ideal, settling below unity through the bulk of the structure rather than exactly on it. The second is the fixed stress concentration at the points where load and reactions are introduced. These sit on the non-design load and support pads, which are held solid and excluded from the sizing, and on the design elements bordering them, which inherit the concentration and cannot size it away. Away from the load and the supports the utilisation therefore rests at or just below one, and it climbs above one only in that boundary neighbourhood. This is exactly what the Sized-to-Load feasibility check of Section 4.5.2.1 reports: it is written on the converged design and classifies it as *locally exceeded* when the excess over unity is confined to that boundary and the bulk is within yield, or *overloaded* when yielding is more widespread. In the utilisation plots the scale is capped at unity, so that every element carrying more than its full-density strength shows in a single grey band and the over-unity regions can be located at a glance.

4.7.3.1 Engine verification through the incremental run sequence

The first run, C1, uses the Scaled load mode with the Safe property model on a gyroid ligament (Figure 4.21). The Scaled mode normalises the stress field so that the single most-stressed element sits at yield and sizes every other element relative to that peak, so the result is the shape of the optimal load path rather than a dimensioned part. The density field resolves the block into a solid load path with graded lattice only where the stress is low, at the outer ends and the lightly loaded corners. Because the Safe model trusts the power law only below half density and promotes anything denser to solid (Section 4.5.1.1), the loaded material appears as solid and the graded band is confined to those low-stress regions. The retained mass is high, at 77.5%, and the specific stiffness is 34 410 (N/mm)/%. The end of the optimisation run then returns a feasibility verdict, which for C1 is locally exceeded with a 99th-percentile utilisation of 0.87 across the design region, confirming that the bulk is within yield (utilisation value ≤ 1) and the excess is at the support zones.

The second run, C2, changes only the load mode, from Scaled to Sized-to-Load, keeping the Safe model and the gyroid ligament (Figure 4.22). Where the Scaled mode returns a shape, the Sized-to-Load mode sizes each element against the real stress it carries under the applied load, so the converged lattice is a dimensioned part rather than a normalised layout. At a common ratio of load to strength the two modes size against the same relative stress field, and the density fields of C1 and C2 accordingly show little difference: the retained mass is 76.9% and the specific stiffness 34 544 (N/mm)/%, both within a fraction of a per cent of C1. This is the expected result, and it is itself a consistency check: the load mode selects the engineering question, a shape or a dimensioned part, without altering the

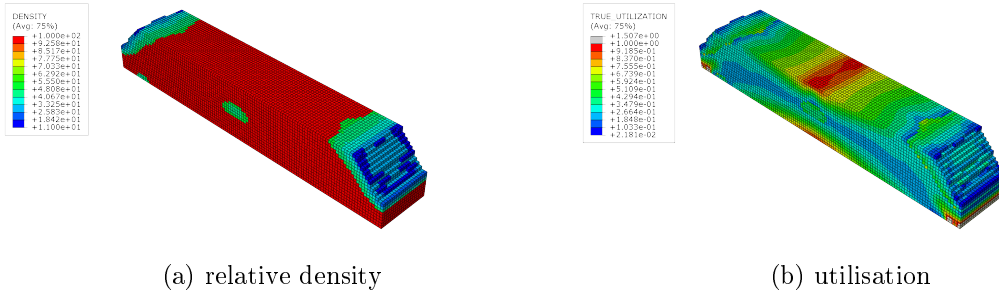


Figure 4.21: Run C1, the Scaled load mode with the Safe property model on a gyroid ligament. (a) The relative density resolves into a solid load path with graded lattice confined to the lightly loaded ends and corners, the Safe cap rendering the loaded material with utilisation above 0.5 solid. (b) The utilisation rests at or below unity through the bulk and exceeds it only at the support zones.

load path the optimiser finds. The feasibility verdict for C2 is consequently similar to C1, locally exceeded, with the same 99th-percentile utilisation of 0.87 across the design region, confirming that the bulk is within yield and the excess is at the boundary.

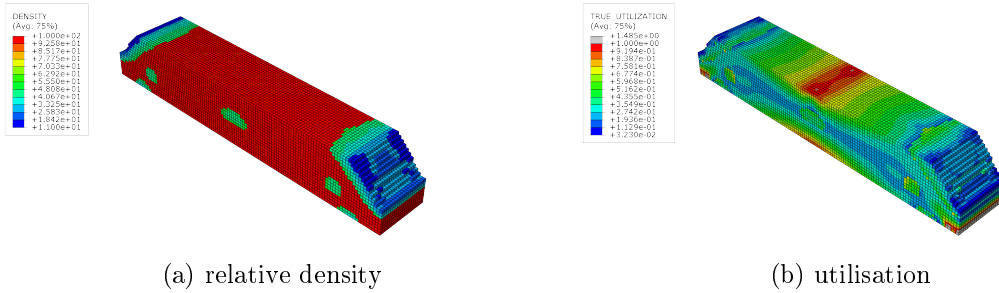


Figure 4.22: Run C2, the Sized-to-Load mode with the Safe model on a gyroid ligament. (a) The density field is almost indistinguishable from that of C1, the two load modes sizing against the same relative stress field. (b) The utilisation field is likewise unchanged; the feasibility check returns the same locally exceeded verdict.

The third run, C3, changes the property model from Safe to Experimental, keeping the Sized-to-Load mode and the gyroid ligament (Figure 4.23). The Experimental model blends the power law into the exact solid at full density, extending the graded band from the half-density cap all the way to unity (Section 4.5.1.1). The optimiser can now grade material through the mid-to-high density range that the Safe model rendered solid, and the density field shows it: intermediate densities appear through the load path where C1 and C2 held solid. Carrying the same load through graded rather than solid material lets the optimiser remove mass, and the retained fraction falls sharply to 67.0% while the specific stiffness rises to 35 127 (N/mm)/%, higher stiffness at lower mass. The verdict remains locally exceeded, at a 99th-percentile utilisation of 0.97. This is the anticipated effect of the model: extending the usable density band is what allows the graded design to be lighter and more efficient than the capped one, at the cost of resting on the power law in a range where it is an extrapolation.

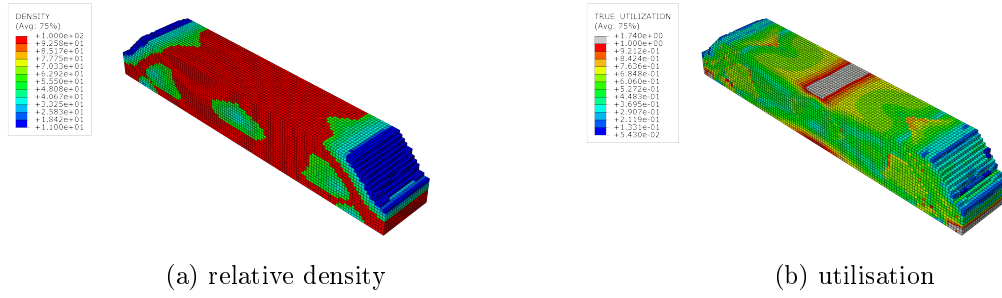


Figure 4.23: Run C3, the Experimental model on the same Sized-to-Load gyroid ligament as C2. (a) Intermediate densities now grade through the load path, where the Safe cap of C2 held solid, and the retained mass falls to 67.0%. (b) The utilisation field is denser through the graded interior; the verdict remains locally exceeded, although with the load patch zone showcasing utilisation over 1.

The fourth run, C4, is the C3 design with the optional solid skin applied (Figure 4.24). The skin seals the porous free surface into a solid shell one element thick, applied once after the search rather than at every iteration (Section 4.5.4). The full external view (a) shows the whole visible surface as solid, the skin envelope. The longitudinal half-cut (b) shows what the skin encloses: a single solid layer wrapping the graded lattice core untouched by the sealing. The half-cut utilisation (c) shows that the skin elements at the lightly loaded ends of the beam carry little stress, as expected of material added to close a surface rather than to carry load. The skin adds the least mass needed to seal the surface, raising the retained fraction from 67.0% to 70.5% and reducing the specific stiffness only slightly, from 35 127 to 34 838 (N/mm)/%, about 1%, so that the manufacturable part keeps almost all of the efficiency of the bare lattice.

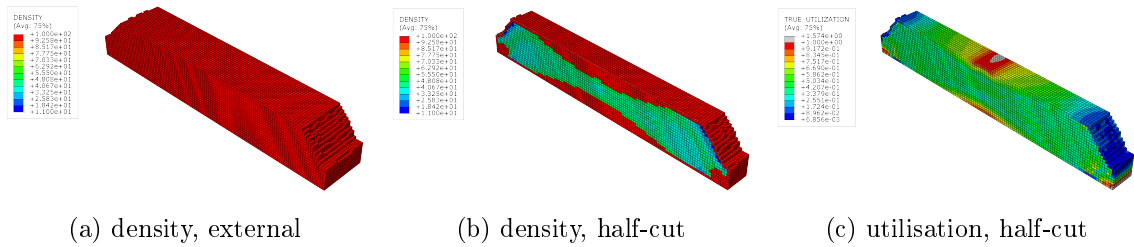


Figure 4.24: Run C4, the C3 design with the optional solid skin. (a) Externally the part reads solid, the one-element skin sealing the whole surface. (b) A longitudinal half-cut shows the single solid layer wrapping the graded lattice core. (c) The half-cut utilisation shows the skin elements at the unloaded ends carrying little stress, as expected of surface-sealing material.

The fifth run, C5, changes the morphology from the gyroid ligament to the I-WP sheet, keeping the Sized-to-Load mode and the Experimental model (Figure 4.25). The I-WP sheet is a matrix-phase triply periodic minimal surface, and by the Maxwell argument of Section 3.4 (Figure 3.9) a sheet morphology carries load by stretching rather than bending and is far stiffer and stronger for its mass than a bending-dominated ligament [93]. The

result follows that expectation directly. Much more of the domain drops to low density, the retained mass falling to 46.2%, well below C3's 67.0% on the identical problem, while the specific stiffness rises to 40 675 (N/mm)/%, the highest of the five runs and clearly above the gyroid ligament. The stretching-dominated sheet reaches higher specific stiffness at lower mass, precisely the trade-off Chapter 3 sets out between the two deformation regimes. The feasibility check here returns an overloaded verdict, with a 99th-percentile utilisation of 1.06: the efficiency of the sheet lets the search reduce the mass to a point at which the utilisation exceeds unity not only at the boundary but more widely through the design. The engine reports this rather than suppressing it, and the contrast with C3's locally exceeded verdict on the same domain shows the feasibility check discriminating between an excess confined to the load introduction and one that already spreads through part of the structure.

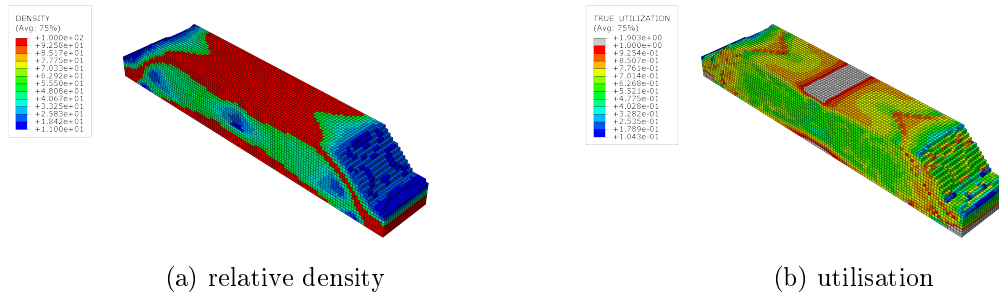


Figure 4.25: Run C5, the I-WP sheet in place of the gyroid ligament of C3. (a) Far more of the domain drops to low density, the stretching-dominated sheet carrying the load at 46.2% mass against C3's 67.0%. (b) The utilisation is higher throughout; the feasibility check returns an overloaded verdict at this reduced mass.

Table 4.5 collects the five results. Read down the table, each option moves the outcome in the direction the method of Section 4.5 predicts: the load mode leaves the layout unchanged, the Experimental model lightens the design, the skin restores a manufacturable surface at a small cost, and the sheet morphology lifts the specific stiffness sharply. None of these can be checked against a published optimum, but each is the response the formulation implies, and their agreement is the internal consistency that stands in for a benchmark here.

A note on the utilisation values applies equally to the table and to every utilisation plot in this section. The peak and 99th-percentile utilisation given in the table are raw per-element values, each read directly from an element's stress and its local yield. The utilisation contours in the figures are instead drawn with nodal averaging, indicated by the (Avg: 75%) in every legend, which blends each element's value with those of its neighbours and rounds off an isolated peak. The maximum read from a plot therefore sits a little below the tabulated peak, most visibly where the peak is confined to a few elements at the load pad and least where it is spread over a broader region.

Table 4.5: The five lattice consistency runs on the three-point-bending domain, all at a load of 12 500 N against a parent yield of 250 MPa. The gyroid ligament is used throughout except in C5. Specific stiffness is in units of (N/mm)/%. Specific stiffness and mass fraction are those of the carried-forward design of highest specific stiffness; the peak and 99th-percentile utilisation are measured on the converged design, on which the Sized-to-Load feasibility check is written and whose specific stiffness lies within 2% of the carried-forward value in every run.

Run	Configuration	Mass (%)	Specific stiffness	Peak util.	99th-pct util.	Verdict
C1	Scaled, Safe	77.5	34 410	1.48	0.87	locally exceeded
C2	Sized-to-Load, Safe	76.9	34 544	1.49	0.87	locally exceeded
C3	Sized-to-Load, Exp.	67.0	35 127	1.82	0.97	locally exceeded
C4	C3 with solid skin	70.5	34 838	1.82	0.97	locally exceeded
C5	Sized-to-Load, Exp., I-WP sheet	46.2	40 675	2.04	1.06	overloaded

4.7.3.2 Robustness to the evolution quota

The density update carries one free numerical parameter, the evolution quota, the fraction of design elements permitted to change state in a single iteration (Section 4.5.3). A converged result that depended strongly on this parameter would be an artefact of the search rather than a property of the design problem. To test this, the C5 configuration was rerun at quotas from one to 5%, and the outcomes are collected in Table 4.6. The first column is the ratio of the converged design’s specific stiffness to the carried-forward peak, and it is near unity throughout, so that the utilisation values, which are read from the converged design, describe essentially the same design as the carried-forward specific stiffness and are admissible alongside it.

The central result is that the peak utilisation is essentially constant across the range, between 2.00 and 2.05. It does not grow with the quota. The over-unity of the most-stressed elements is therefore not a step-size effect but a property of the converged state, fixed by the load and support concentration and the material limit rather than by how much of the structure is resized at each iteration. Around this constant peak the quota trades three things off. The specific stiffness is marginally higher at low quota, by approximately 0.8% (from 5% to 1%), the finer search capturing a slightly better optimum in the same way the dynamic evolutionary ratio of Section 4.3 does for the solid-void engine. Convergence is markedly faster at high quota, reaching the best design in fifteen iterations at 5% against eighty-one at 1%, and the converged-to-peak ratio falls at the higher quotas, the coarser search overshooting its own optimum and being carried back to it. The density distribution also broadens as the quota rises. This follows from the update itself: the quota draws the lowest-utilisation elements to be resized each iteration, so a larger quota reaches further up the utilisation ranking and pulls moderate-utilisation elements toward their own intermediate targets, spreading the density field, whereas a small quota resizes only the least-stressed elements each step and polarises it. In every case, though, the peak utilisation and the converged specific stiffness are recovered to within 1% to 2%, which is the robustness the check was meant to establish.

Table 4.6: The C5 (Sized-to-Load, Experimental, I-WP sheet) configuration rerun at evolution quotas from one to five %. The converged-to-peak ratio confirms that the converged design, on which the utilisation is measured, is close to the carried-forward peak-stiffness design. Peak utilisation is essentially constant, confirming that the over-unity is not a step-size effect; specific stiffness is marginally higher, and convergence markedly slower, at low quota. Specific stiffness is in units of (N/mm)/%.

Quota (%)	Converged/peak	Peak specific stiffness	Peak util.	99th-pct util.	Iterations to peak
1	1.000	40 810	2.00	1.03	81
2	0.999	40 723	2.02	1.04	41
3	0.998	40 675	2.04	1.06	27
4	0.981	40 579	2.05	1.13	21
5	0.983	40 479	2.05	1.13	15

Taken together, the five configuration runs and the quota study verify the lattice engine in the only sense available to it. Each option produces the response the fully-stressed formulation predicts, the utilisation fields carry the characteristic over-unity confined to the load and support boundary, and the converged result is robust to the one free parameter of the search. The engine reaches its fully-stressed target across the bulk of the structure, reports explicitly where and by how much that target is exceeded, and does so consistently across morphologies, property models and load modes.

4.7.4 Comparison of the two engines

Sections 4.7.2 and 4.7.3 verified each engine on its own terms. A final check sets the two against each other on the same problem, the three-dimensional three-point-bending domain of Table 4.2, to see whether they agree on the load path and to measure the difference in efficiency between a discrete solid-void layout and a graded lattice. The two lattice designs carried forward from Section 4.7.3, the Experimental gyroid ligament C3 at a retained fraction of 67.0% and the Experimental I-WP sheet C5 at 46.2%, converged to very different masses, so a single common-fraction comparison would flatter one and penalise the other. Each lattice is therefore set against a solid-void design optimised, with the settings of the A3 benchmark, to that lattice’s own volume fraction (runs D1 and D2 of Table 4.2), so that every comparison is made at equal mass.

Figure 4.26 sets the gyroid ligament against the solid-void design at 67.0%. The two topology renders (a, b) are the same load path realised in two ways: the solid-void engine returns a solid body with material cut away from the low-stress regions, discrete voids opened at the neutral axis and the lightly loaded ends, while the lattice engine returns a graded gyroid, near-solid along the load path and opening into visible cells where the stress falls, the ligament morphology shown in the close-up (e). The von Mises fields (c, d), each scaled to its own peak, carry the same reading: both engines drive the stress along the bending load path from the central load down to the supports, both leave the bulk of the design within the parent yield of 250 megapascals, and both exceed that yield, at

similar peak magnitudes near four hundred megapascals, only at the load and support pads, the load-introduction artefact already identified in Section 4.7.3. On specific stiffness the solid-void design reaches 38 721 (N/mm)/% against the lattice’s 35 127, higher by about 10%.

Figure 4.27 repeats the comparison for the I-WP sheet against the solid-void design at 46.2%. The picture is the same in kind: the solid-void body carries more open voids at this lower fraction, and the sheet lattice, whose morphology the close-up shows, grades from near-solid to open cells along the same load path, its von Mises field again matching the solid-void field in distribution and in peak. Here the solid-void design reaches 43 316 (N/mm)/% against the lattice’s 40 675, higher by about 6.5%, a smaller margin than for the gyroid.

Table 4.7 collects the four results. At both fractions the solid-void engine reaches the higher specific stiffness, which is the expected outcome: free to make any element fully solid or fully void, it concentrates all of its material into the load path and removes the rest, whereas the lattice must grade a bounded cellular density through the whole domain and cannot exceed the effective stiffness its morphology provides at each density. The margin is the more informative figure. It is about 10% for the bending-dominated gyroid ligament but only six and a half for the stretching-dominated I-WP sheet: the stiffer morphology of Section 3.4 recovers most of the distance to the discrete optimum, as the Maxwell argument would predict [93].

Table 4.7: Specific stiffness, κ of Equation (4.5) in units of (N/mm)/%, of each lattice against a solid-void design optimised to the same volume fraction. The solid-void engine reaches the higher value at both fractions, by a margin that is smaller for the stiffer sheet morphology. The lattice figures use the Experimental property model.

Volume fraction	Lattice morphology	Lattice κ	Solid-void κ	Solid-void margin
67.0%	gyroid ligament (C3)	35 127	38 721	+10.2%
46.2%	I-WP sheet (C5)	40 675	43 316	+6.5%

This margin is the price of what the lattice returns in its place. A graded lattice is self-supporting by construction and presents a smooth cellular interior in place of the sharp re-entrant boundaries of a cut solid, and it carries the functional grading and the multifunctional behaviour set out in Chapter 3; the comparison is a measurement of that trade-off rather than a verdict for either engine. Two qualifications attach to the lattice figures. They rest on the Experimental property model of Section 4.5.1.1, a fast estimate of the graded response rather than a homogenisation from first principles, and C5 carries the overloaded verdict noted in Section 4.7.3; the comparison should be read with both in mind.

A final note concerns the renders. Realising a graded lattice as an explicit surface at these densities, which reach the eleven-per-cent floor, at the one-millimetre period that would match the analysis resolution exactly is impractical: the sampling cost scales with

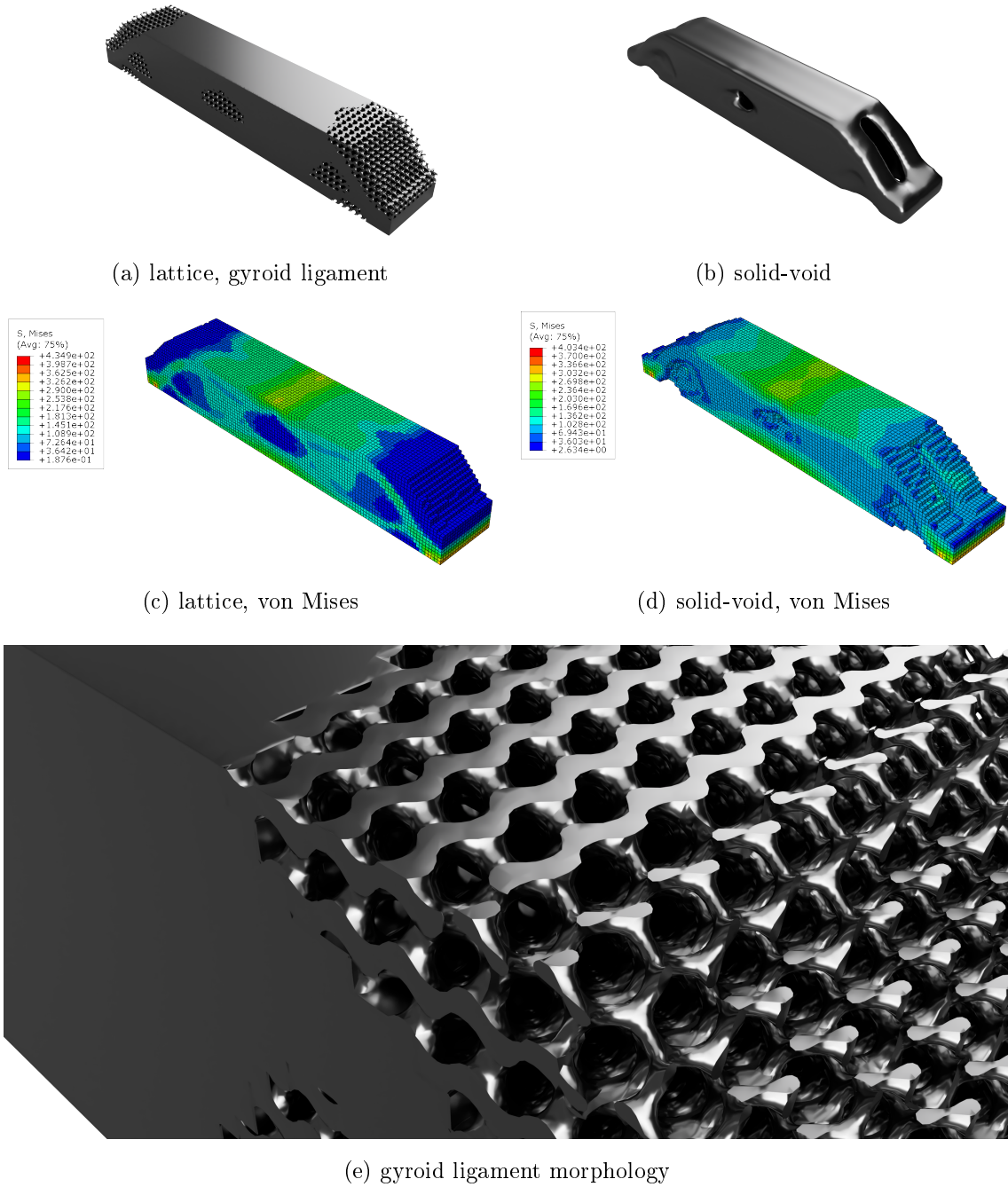


Figure 4.26: The two engines at a volume fraction of 67.0%: the gyroid ligament C3 against a solid-void design at the same fraction. The topology renders show the same load path realised as (a) a graded gyroid and (b) a solid body with discrete voids, with (e) the ligament morphology in detail. The von Mises fields (c, d), each on its own scale, show both engines finding the same bending load path and exceeding the 250 megapascal yield only at the load and support pads.

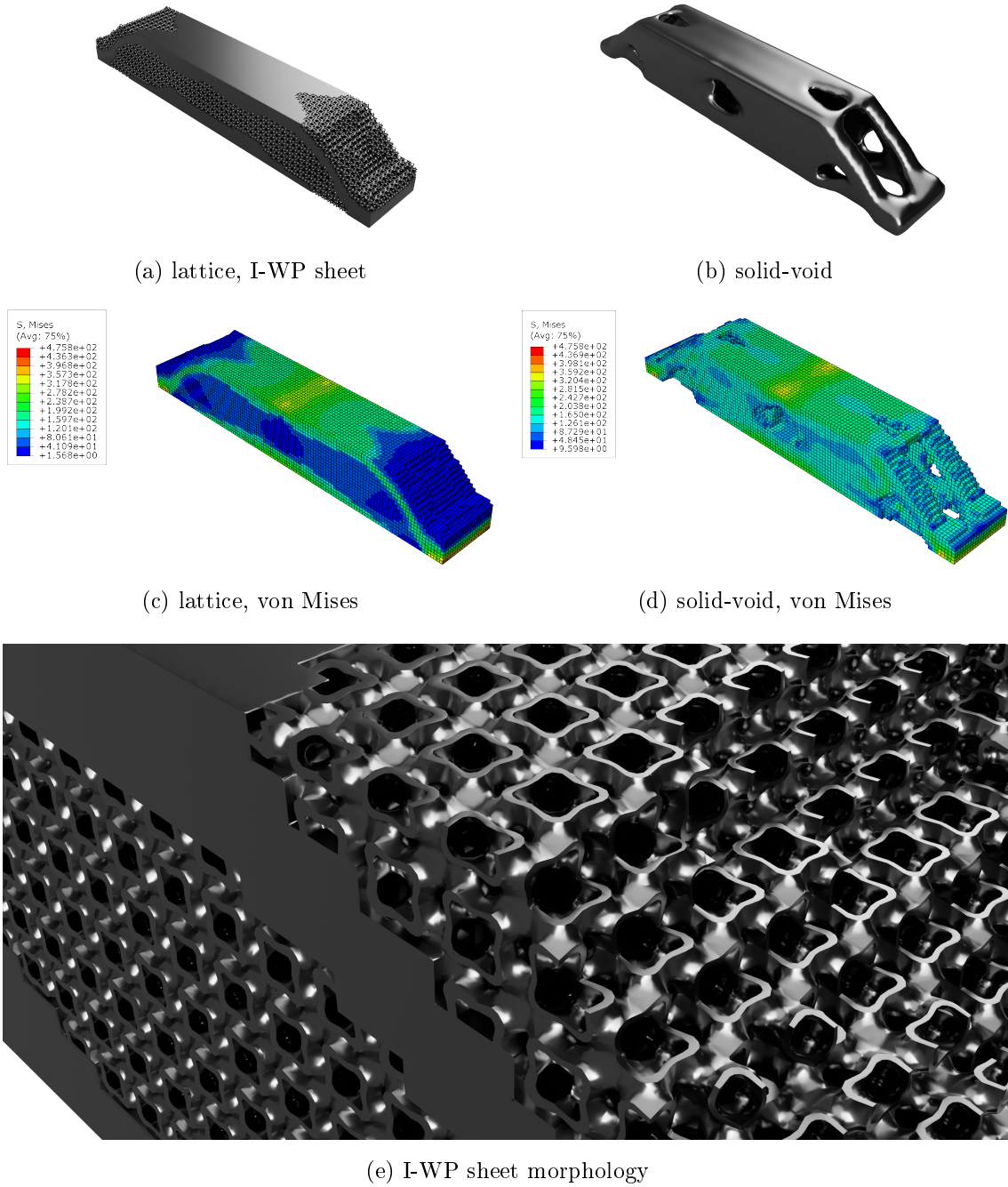


Figure 4.27: The two engines at a volume fraction of 46.2%: the I-WP sheet C5 against a solid-void design at the same fraction. As in Figure 4.26, (a, b) show the topology renders, (c, d) the two von Mises fields, each on its own scale, agreeing on the load path, and (e) the sheet morphology in detail.

the cube of the resolution (Section 4.6.2.1), and a faithful mesh at that period would run to several hundred million triangles and well over a hundred gigabytes of memory. The renders in Figures 4.26 and 4.27 were therefore generated at a two-millimetre period, twice the analysis element size, which brings the cost down to a few tens of millions of triangles and of the order of ten to twenty gigabytes. At that period a single unit cell spans a block of elements and averages their densities, as Section 4.6.2.1 sets out, so the renders reproduce the morphology and the load path faithfully but are a visual estimate of the exact optimised geometry rather than a reproduction of it. The solid-void bodies in the same figures are shown after the boundary-smoothing pass of Section 4.6.1, one hundred iterations of the shrink-free Taubin filter at its default parameters ($\lambda = 0.50$, $\mu = -0.53$) [110]; without it the surface would carry the staircase of the hexahedral mesh the optimiser works on, and the smoothing recovers the manufacturable surface the discrete layout stands for, which is why the solid-void designs appear smooth rather than voxelised.

Across the two fractions the two engines reproduce one another’s load path and differ in realisation and in efficiency as their formulations predict, the discrete layout the stiffer per unit mass and the graded lattice the more manufacturable and functional. With this the verification of both engines, and of the shared formulation beneath them, is complete.

Chapter 5

Application to the Furia Suspension Component

The optimisation framework of Chapter 4 was developed and verified on benchmark problems of known solution, which established its correctness but not its behaviour on a real component. This chapter addresses that gap. It applies the framework to the roll-damper bracket of the Adamastor Furia, the component and its industrial setting introduced in Chapter 1, following it from its original steel design through optimisation, manufacture-oriented realisation, and structural validation.

The chapter sets out the patented suspension mechanism, the target component, and the kinematic model built to understand them (Section 5.1). Section 5.2 defines the optimisation model: the design space and the clearance study that fixes it, the finite-element model and its boundary conditions, and the load cases derived from the mechanism. Section 5.3 reports the mesh convergence and sensitivity-field rank study that justifies the discretisation adopted for the run. Section 5.4 presents the optimisation itself, Section 5.5 the realisation of its result as a manufacturable body, and Section 5.6 the structural validation, by static analysis and by fatigue, of the optimised part. Section 5.7 closes with the findings and their implications for the partner Adamastor.

5.1 Suspension Mechanism and Target Component

5.1.1 Decoupled roll-heave mechanism

The Furia’s suspension is built on a mechanism that the manufacturer has patented [115], and understanding it is a prerequisite for everything that follows, because it is the mechanism that decides how the target component is loaded. Its governing idea is the separation of two motions that a conventional suspension treats together. When the two wheels of an axle move in the same direction the axle moves symmetrically, in heave; when they move in opposite directions, as the car leans into a corner, the motion is asymmetric, in roll. A conventional arrangement reacts both through a shared set of springs and an anti-roll

bar, so that tuning one affects the other. The patented mechanism instead routes the symmetric and the asymmetric motions to two separate dampers, a heave damper and a roll damper, each of which can be tuned without disturbing the other [115].

The distinctive feature of the arrangement, and the one from which the target component's loading follows, is the way the roll motion is fed to its damper. Two rocker arms, one per wheel, are linked through a pair of brackets and guide rods so that a roll in either direction, left or right, always compresses the single roll damper, and the return from roll always extends it. The damper therefore sees the same sense of motion whichever way the car leans, which gives it a response that is symmetric between the two roll directions and lets the damping of the roll-building and roll-releasing phases be set independently [115]. A secondary benefit, and one directly relevant to a lightweighting study, is that consolidating the roll function into this single decoupled path reduces the part count, mass and installed volume of the suspension relative to a conventional roll-control arrangement [115]. Figure 5.1 shows the assembly and names its principal parts.

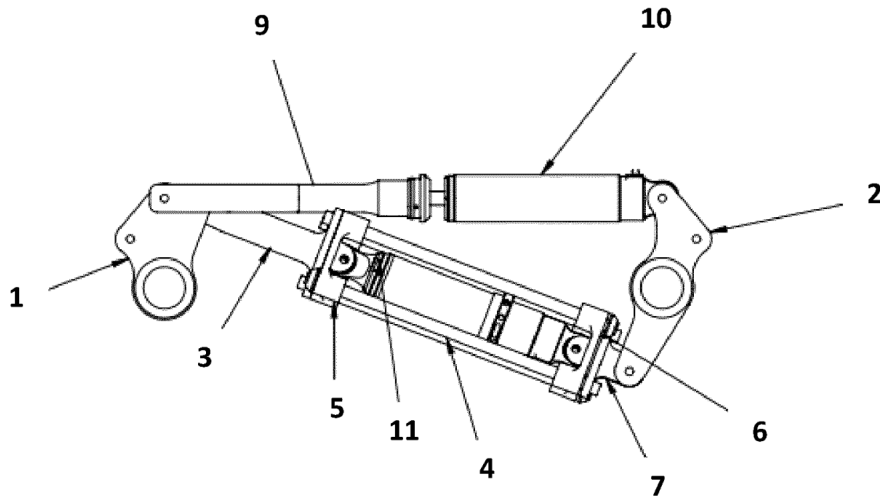


Figure 5.1: The Furia's decoupled roll-heave suspension mechanism, reproduced from Figure 4 of the patent [115]. The two rocker arms (components 1 and 2), linked through the brackets and guide rods (item 4), feed roll motion to the single roll damper (component 11) so that a roll in either direction compresses it, while the heave damper (component 10) reacts the symmetric motion along a separate path. The component optimised in this chapter is the left-hand outer bracket that carries the roll-damper load into the rocker, identified with the number 3 on the Figure.

5.1.2 Target component

The component optimised in this work is the left-hand outer bracket of the roll-damper assembly, element 3 of the patent [115], referred to hereafter as the roll-damper bracket. It is the part through which the roll-damper load is introduced into the rocker on one side of

the car: two guide rods pass through it and are secured by bolts threaded into their ends, and it pivots on the rocker through a pin joint, so that the bracket keeps its orientation while the rocker rotates beneath it over the suspension's travel. The same mechanism is used at both ends of the car, the rear axle moving through a larger roll range than the front, so that a single bracket geometry able to serve the rear also serves the front; this commonality is returned to when the design space is fixed in Section 5.2.1.

In its original form the bracket is machined from FR-3, a chromium-nickel-molybdenum construction steel supplied by the Portuguese manufacturer Ramada Aços [116]. At a density of 7.84 g/cm^3 the part (Figure 5.3), of volume $1.935 \times 10^5 \text{ mm}^3$, weighs 1.520 kg, which is heavy for its function and makes it an attractive candidate for lightweighting. The reason it is steel at all lies in its load path. The geometric constraints on the bracket's position, between the rocker and the neighbouring components and across the full range of suspension travel, prevent the line of action of the roll-damper load from passing through the pin about which the bracket reacts it. The two are separated by a lateral offset of approximately 13 mm (Figure 5.2), so that the load is reacted at the pin as a bending couple rather than as a direct force, and it is this eccentric, bending-dominated load path that originally dictated a high-strength steel: a lighter alloy, in the same solid form, would not carry the resulting bending within the space available. The eccentricity cannot be designed out, since it follows from the packaging of the mechanism, but a bending load path concentrated along a known line is exactly the kind of problem topology optimisation is suited to, placing material precisely where the bending demands it, and it is this that makes a lighter alloy viable where a solid redesign would not be.

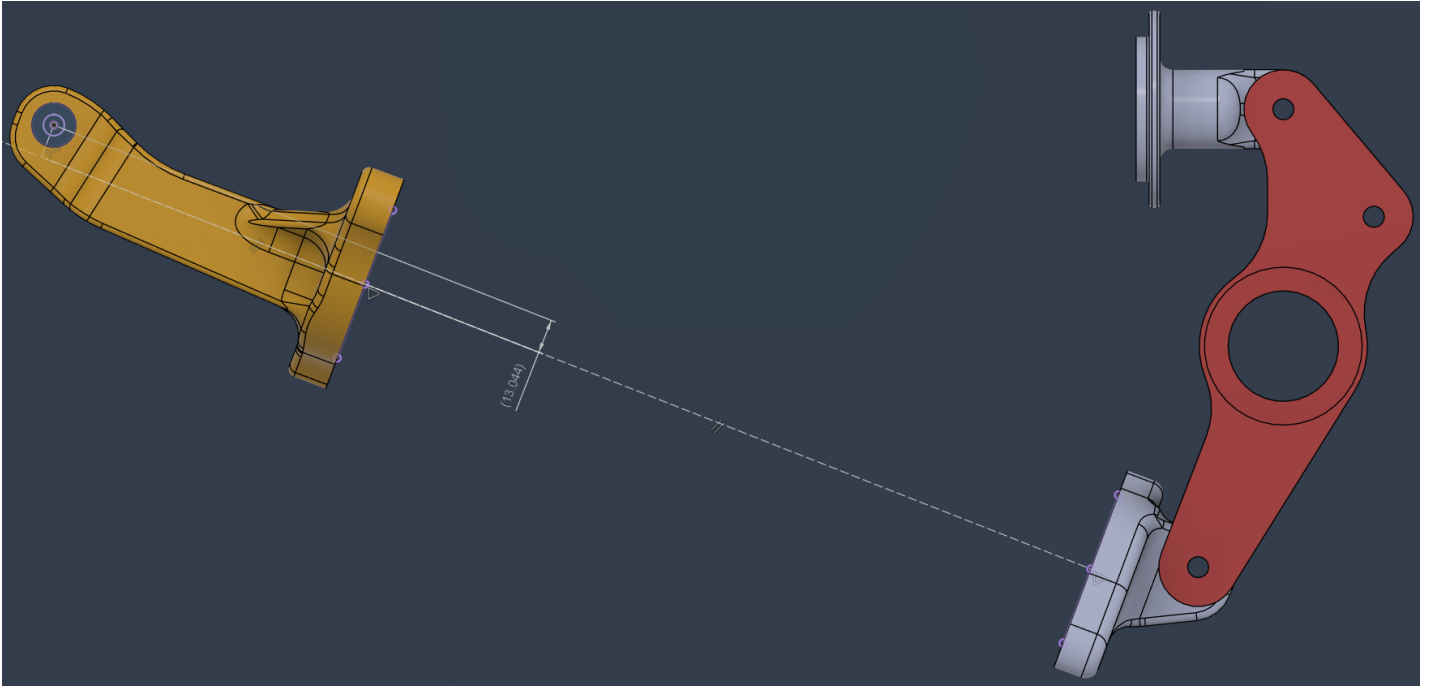


Figure 5.2: The load-path eccentricity of the roll-damper bracket. The roll-damper load axis is offset from the pin axis by approximately 13 mm, so the load does not pass cleanly through the pin but reaches it as a bending couple; this eccentric, bending-dominated load path is what originally dictated a high-strength steel.



Figure 5.3: The original roll-damper bracket, machined from FR-3 steel, of volume $1.935 \times 10^5 \text{ mm}^3$ and mass 1.520 kg.

5.1.3 Assembly and kinematic modelling

To ground the optimisation in the real behaviour of the mechanism rather than in assumptions about it, the complete suspension assembly was modelled in Autodesk Fusion. The kinematic model served three purposes. It recovered the true motion of the bracket relative to the rocker and its neighbours across the suspension's travel; it established, for each of the load cases treated below, how and where the corresponding load is introduced into the bracket; and it generated the volume available to the optimiser, by sweeping the moving assembly through its range and recording the space that remains clear of every other part throughout. Figure 5.4 shows the assembly at the two extremes of the rear axle's roll range, at -10° and $+10^\circ$, the positions that bound the clearance study of the next Section 5.2.



Figure 5.4: The suspension assembly modelled in Autodesk Fusion, shown at the two extremes of the rear axle's roll range, (a) -10° and (b) $+10^\circ$. The roll-damper bracket is the yellow part. Sweeping the assembly through this range recovers the bracket's motion, identifies how each load is introduced, and generates the clearance envelope that defines the design space.

5.2 Optimisation Model

5.2.1 Design space

The design space is the volume within which the optimiser is free to place or remove material, and for this component it was constructed from the clearance envelope of the kinematic model. The rear axle moves through a larger roll range than the front, from -10° to $+10^\circ$ against the front's -7.5° to $+7.5^\circ$, so the rear range is the more demanding and was used to fix the geometry. The moving assembly was swept through the full -10° to $+10^\circ$, and the design space was taken as the volume that remains clear of the rocker and of the heave-damper mounting bracket, element 9 of the patent, throughout that sweep, with a margin of at least 2 mm maintained to each of them at every position. The single exception is the pin-boss region of the bracket, which was deliberately thickened to improve the load introduction at the bore, reducing its clearance to the rocker clevis to 1 mm. A geometry that clears the rear range by this margin necessarily clears the smaller front range as well, so the resulting bracket serves both axles. The functional interfaces of the part,

the pin bore, the two counterbore holes that take the guide-rod bolts, and the planar face against which the roll-damper load bears, are held as non-design regions, fixed as solid throughout the optimisation so that the mating features are preserved. The design space so defined has a volume of $5.529 \times 10^5 \text{ mm}^3$. Figure 5.5 shows it in isolation and in place within the assembly.

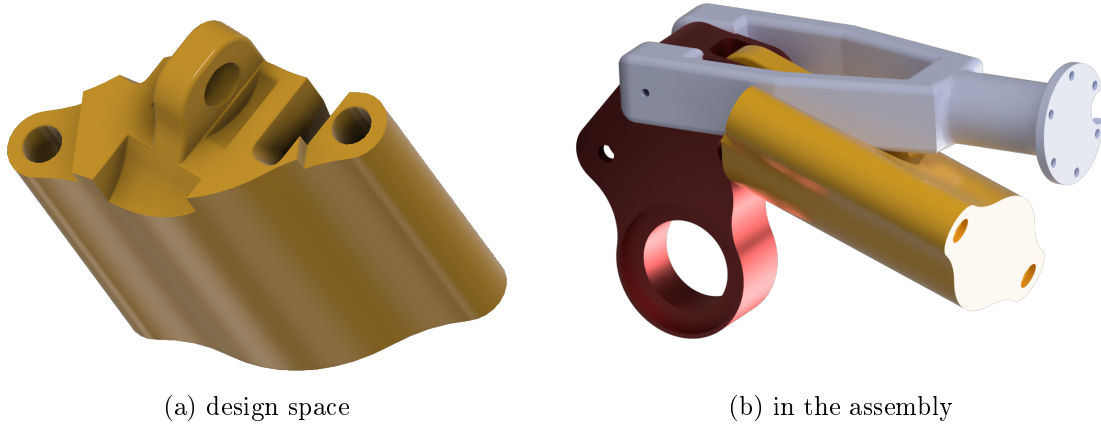


Figure 5.5: The design space of the roll-damper bracket, of volume $5.529 \times 10^5 \text{ mm}^3$, shown (a) in isolation and (b) in place within the suspension assembly. It is the volume that remains clear of the rocker and the heave-damper mounting bracket, by at least 2 mm, across the full -10° to $+10^\circ$ rear roll range; the pin-boss region is the sole exception, thickened to a 1 mm clearance to improve the load introduction at the bore.

5.2.2 Finite-element model and boundary conditions

The optimisation is driven by a finite-element model of the bracket in its assembly, and the fidelity of that model's boundary conditions governs the credibility of the sensitivities the optimiser works from, so the model was built to represent the real load path rather than an idealisation of it. The same model is used for the optimisation and, later, for the validation of Section 5.6; only the mesh is refined between the two, and that refinement is described where it is used. This section sets out the model itself.

The bracket is assembled with the rocker it acts against, both as meshed bodies. The connection between them is not a bonded interface, which would fuse the two parts and carry a moment across a joint that in reality carries none, but a hinge: the pin joint is represented by a revolute connector that transmits the three force components and the two transverse moments while releasing rotation about the pin axis, reproducing the freedom of the cylindrical bush that lets the bracket keep its orientation as the rocker turns beneath it. Load is introduced at the pin bore through a distributing coupling between the bore surface and a reference point at the pin centre, chosen in preference to a rigid coupling so that the bore is free to deform under load rather than being artificially stiffened. The two guide rods that locate the bracket are represented as axial connectors, each carrying force

along its own axis to a grounded anchor, since it is these rods, not the freely rotating pin, that provide the orientation restraint of the real assembly.

The external loads are introduced through kinematic couplings, each tying a reference point to the surface over which its load is physically distributed. The push-rod load enters at a reference point coupled to the clevis bores through which the push rod installs; the heave-damper load enters similarly at its own reference point and clevis bores. The roll-damper load enters by one of two paths, according to the sense of the roll, as the mechanism of Section 5.1.1 dictates. In a left roll the load reaches the bracket through the two guide-rod bolts, and is applied at two reference points, each coupled to the counterbore surface against which its bolt head seats. In a right roll the same load bears instead on the planar face of the bracket, pushed there by the same-side inner bracket, and is applied at a reference point coupled to that whole face. These load paths, and the reference points that carry them, are shown for each load case in Figure 5.6 of the next section.

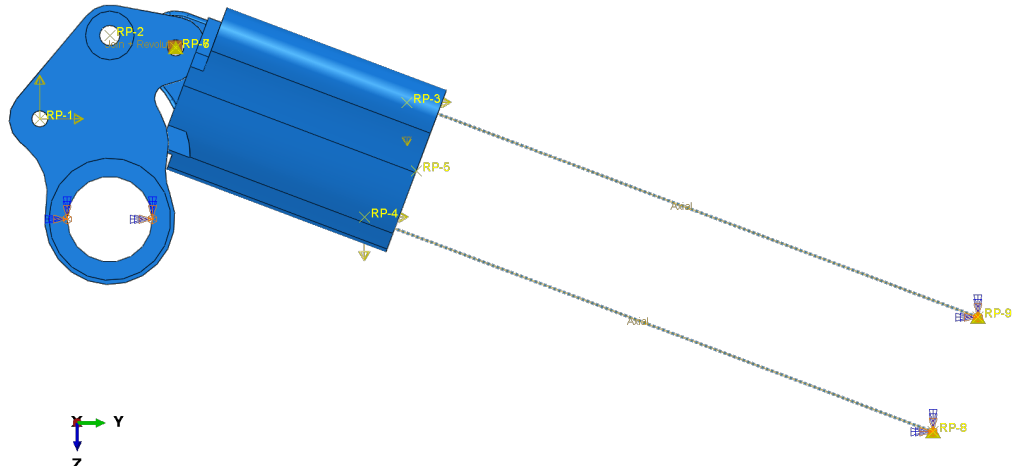
The bracket material is the AlSi10Mg to be built by selective laser melting, modelled as linear elastic with a Young's modulus of 70 GPa and a Poisson's ratio of 0.33. Of the two engines of Chapter 4, the solid-void engine was chosen for this component. The bracket is a performance-critical part for which stiffness per unit mass is the decisive measure, and the solid-void engine, free to make any element fully solid or fully void, reaches a higher specific stiffness than the graded lattice at a given mass, as the equal-mass comparison of Section 4.7.4 established. Where that comparison measured the price of the lattice's manufacturability and multifunctionality, here the priority is structural efficiency, and the discrete engine is the appropriate choice.

5.2.3 Load cases

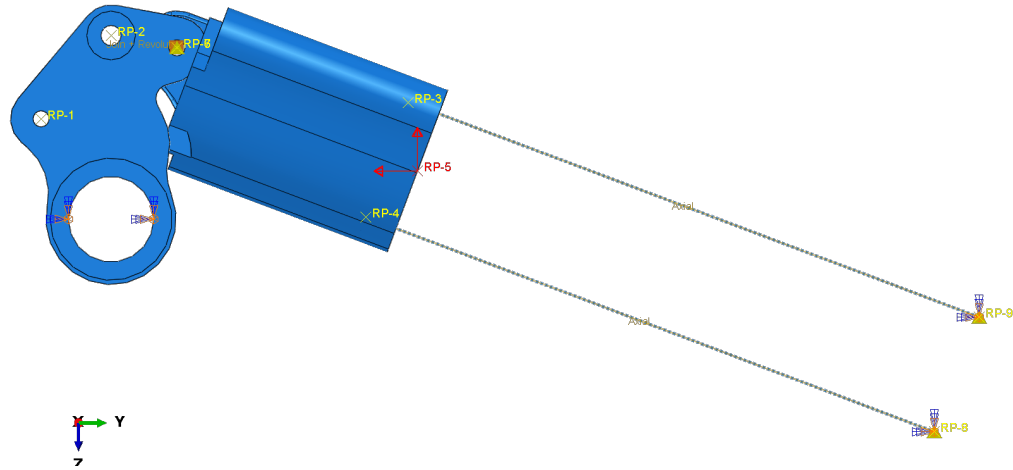
Three load cases were derived from the mechanism, each loaded to the peak force its source can deliver, so that the optimisation is driven by the worst case of each scenario rather than by a nominal load. The peak forces are set by the dampers' rated capacity. Each damper is characterised by a spring rate and a maximum stroke, their product being the largest force it can apply: the heave damper and the front-axle roll damper are each rated at 445 kgf/cm over a maximum stroke of 7 cm, which corresponds to a peak force of about 30.56 kN, while the rear-axle roll damper, rated at 253 kgf/cm over the same stroke, delivers the lower peak of about 17.4 kN. The push rod contributes a separate peak of 27 kN, the largest force any push rod applies in the assembly, quoted by Adamastor. Because the front-axle roll damper is the more severe of the two and the bracket is designed to serve both axles, the front-axle roll-damper load was used throughout; the geometry, by contrast, was validated across the larger rear-axle roll range in Section 5.2.1, so that the single design is proven for both ends of the car in loading and in packaging alike.

With the peak forces fixed, the three load cases follow from how the mechanism introduces them, and are shown in Figure 5.6. In the left-roll case the push-rod load and the roll-damper load act together, the roll damper entering through the two guide-rod bolts

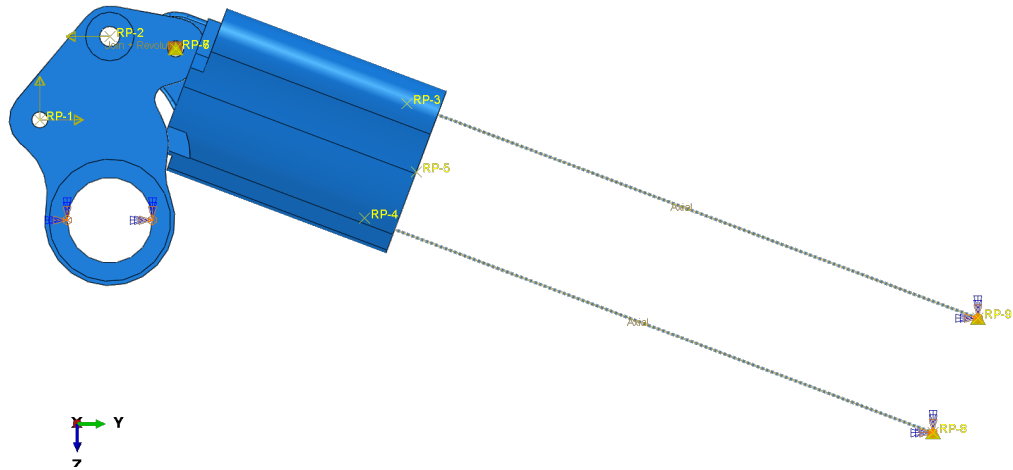
at the counterbore reference points. In the right-roll case only the roll-damper load acts, entering instead on the planar face, as the same-side inner bracket compresses the damper and reacts against that face. In the heave, or bump, case the roll damper is not loaded at all, and the bracket carries the push-rod and heave-damper loads together, both introduced through the rocker. The three cases were combined into a single sensitivity field by the weighted sum of Equation (4.1) of Chapter 4, with equal weights across the three, so that no single scenario dominates the optimised layout.



(a) left roll



(b) right roll



(c) heave

Figure 5.6: The three load cases, as introduced into the finite-element model. (a) Left roll: push-rod load and roll-damper load together, the roll damper entering through the two guide-rod bolts at the counterbore reference points. (b) Right roll: roll-damper load only, entering on the planar face. (c) Heave: push-rod and heave-damper loads together, with the roll damper unloaded. The guide rods appear as the axial connectors running to their grounded anchors, and the pin as the revolute connection to the rocker.

5.3 Mesh Convergence and Sensitivity-Field Rank Study

The optimisation model of the previous section must be discretised before it can be solved, and the choice of mesh is not free: it must be fine enough that the result is trustworthy, yet coarse enough that the many solves of an optimisation run remain affordable. This section reports the study by which that choice was made. It has an unusual outcome. A conventional mesh convergence study, reported first, does not converge in the ordinary sense for one part of the model, which would normally forbid using the coarse mesh the run needs. The resolution is to recognise what the optimiser actually requires of the mesh, and a dedicated study, reported second, establishes that this weaker requirement is met even where ordinary convergence is not, which is what justifies the discretisation finally adopted.

5.3.1 Mesh convergence

Two mesh convergence studies were carried out, each refining one part of the model while holding the other fixed. The first refined the mesh of the component itself, from a 2 mm to a 0.75 mm element size with the rocker held fixed, and behaved as convergence studies are expected to: the reference displacement changed monotonically and by a diminishing amount at each refinement, the step from 1 mm to 0.75 mm being 2.67%, within the 5% threshold taken as converged. The 1 mm component mesh is therefore justified, and the finer 0.75 mm mesh, which demands of the order of 40 GB of memory and almost a full hour to solve, is impractical for a study that must solve the model many hundreds of times.

The second study refined the mesh of the rocker, from 4 mm to 1 mm with the component held at 1 mm, and did not converge. The reference displacement did not settle but moved non-monotonically, falling slightly from 4 mm to 3 mm and then rising sharply, by some 35% and then a further 52%, as the mesh was refined to 2 mm and to 1 mm. Switching the rocker elements from linear to quadratic tetrahedra reproduced the same pattern rather than removing it, which shows the cause is not element order. The cause is geometric. The rocker meets the component at the clevis bore, where the hinge connection between them is made, and at a coarse mesh the circular bore is represented as a coarse polygon; as the mesh refines, the bore geometry, and with it the surface over which the hinge coupling engages, changes qualitatively, and the apparent stiffness of the joint steps with it. This is a discretisation artefact localised at the connection, not a genuine mesh sensitivity of the component, and it meant the rocker mesh could not be converged in the ordinary way.

5.3.2 Sensitivity-field rank preservation

That the rocker mesh does not converge would ordinarily rule out using the coarse 4 mm rocker that keeps the optimisation affordable. It does not, and the reason lies in what the optimiser requires of the model. The bidirectional evolutionary scheme of Chapter 4 does

not act on the absolute values of the sensitivity field: at each iteration it ranks the elements by sensitivity and removes the lowest-ranked fraction (Section 4.3). What must be right is therefore not the magnitude of the sensitivities but their ordering (Section 3.1.1), and specifically the ordering within the component, whose elements the optimiser adds and removes. The question is thus not whether the coarse rocker gives the right displacement, which it does not, but whether it gives the same ranking of the component's elements as a fine rocker would.

A dedicated study was carried out to answer this. For each finer rocker mesh, the per-element sensitivity field over the design region was computed with the same load-case weighting used in the optimisation, and its ranking was compared against that of the 4 mm rocker, the mesh actually used in the run, taken as the reference. The agreement is near-perfect. Across the comparisons the Spearman rank correlation exceeds 0.998 and the Kendall correlation lies between 0.988 and 0.997, for both linear and quadratic rocker elements (Figure 5.7); both coefficients are defined in Section 3.1.1. The elements the optimiser acts on most directly agree just as strongly: the most highly stressed elements, which it preferentially retains, agree with the reference to between 96.1% and 99.2%, and the least stressed elements, which it removes first, to between 98.2% and 99.6% (Figure 5.8); the set of elements removed in the very first iteration, at the evolutionary ratio of the run, agrees to between 96.9% and 99.4%. The ordering the optimiser depends on is thus essentially independent of the rocker mesh.

Sensitivity Field Rank Preservation vs 4mm Rocker Reference

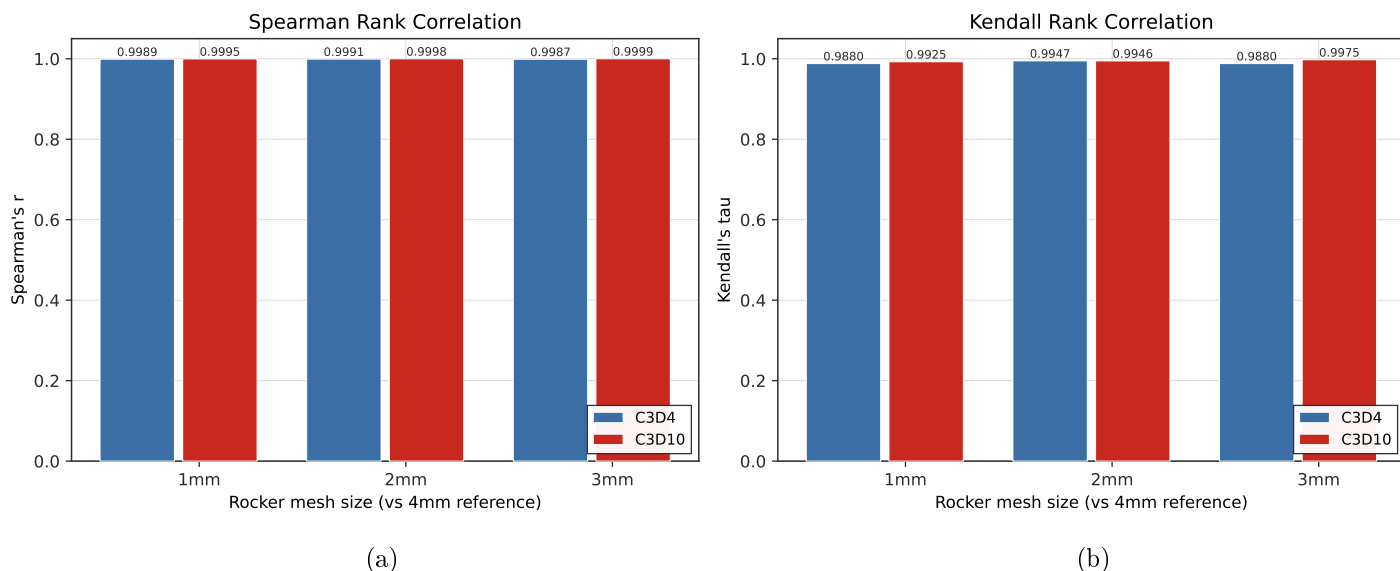


Figure 5.7: Spearman rank correlation (a) and Kendall correlation (b) of the component's sensitivity field, for each rocker mesh compared against the 4 mm reference, for both linear (C3D4) and quadratic (C3D10) rocker elements. All values exceed 0.988, indicating a near-identical ranking of the component's elements regardless of the rocker mesh.

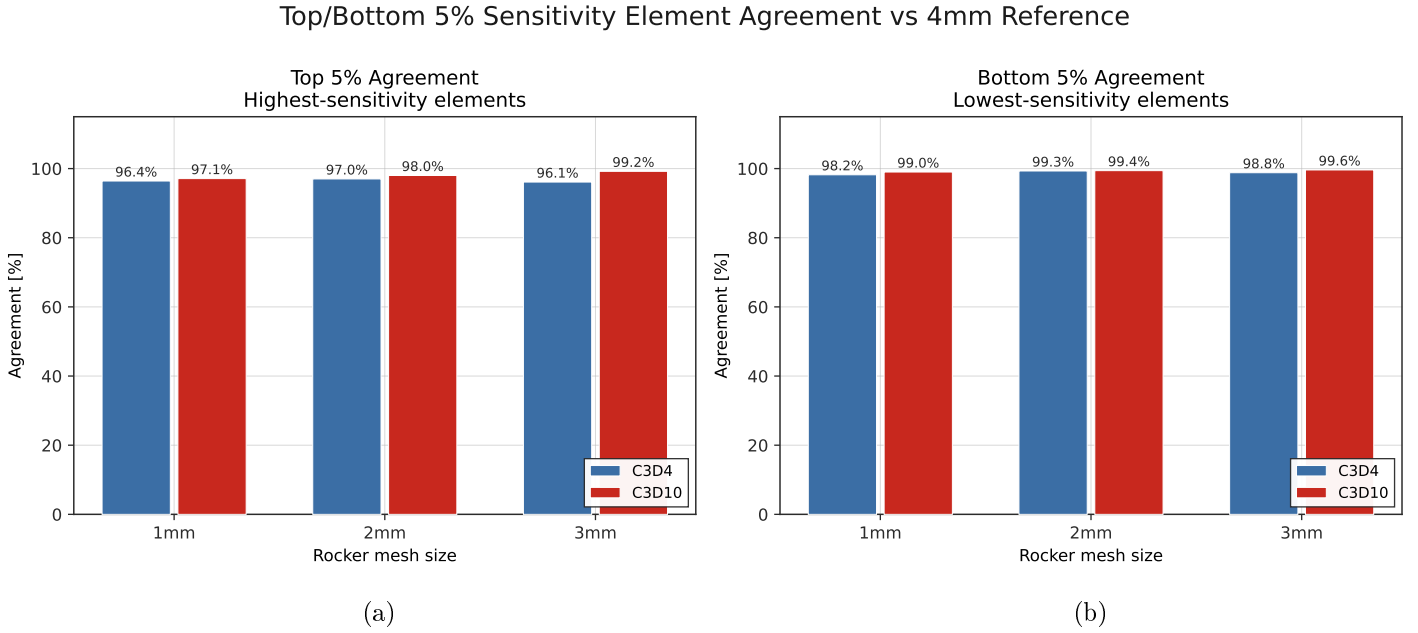


Figure 5.8: Agreement, against the 4 mm reference, of the top 5% highest-sensitivity elements the optimiser preferentially retains (a) and the bottom 5% lowest-sensitivity elements it removes first (b), for each rocker mesh and element type. The elements most active in the optimiser’s decisions are in near-perfect agreement regardless of the rocker mesh.

The small disagreement that does exist is confined to exactly where the artefact of Section 5.3.1 predicts. Grouping the design elements into equal-count bins by distance from the pin bore and correlating each bin separately (Figure 5.9) shows the correlation reduced only in the bin nearest the bore, to between 0.843 and 0.886 for linear and between 0.944 and 0.993 for quadratic rocker elements, and recovering to between 0.995 and 1.000 from the second bin outward. The recovery is abrupt rather than gradual, which is the signature of Saint-Venant’s principle (Section 3.1.1): the perturbation the coarse rocker introduces at the clevis bore is felt only in a thin shell of material immediately around the connection and does not reach the bulk of the design domain. The few hundred elements whose rank shifts most between meshes all lie within this shell, between about 12 and 29 mm of the bore center, and they are high-sensitivity elements that the optimiser retains regardless of their exact order within that cluster. The direct comparison of the sensitivity values themselves, in Figure 5.10 for linear and Figure 5.11 for quadratic rocker elements, tells the same story: the points lie on the line of equality, the only visible scatter is in the elements nearest the bore, and the high-sensitivity elements that matter most to the optimiser sit squarely on it.

Per-Bin Spearman Correlation vs Distance from Pin Boss

Saint-Venant's principle check

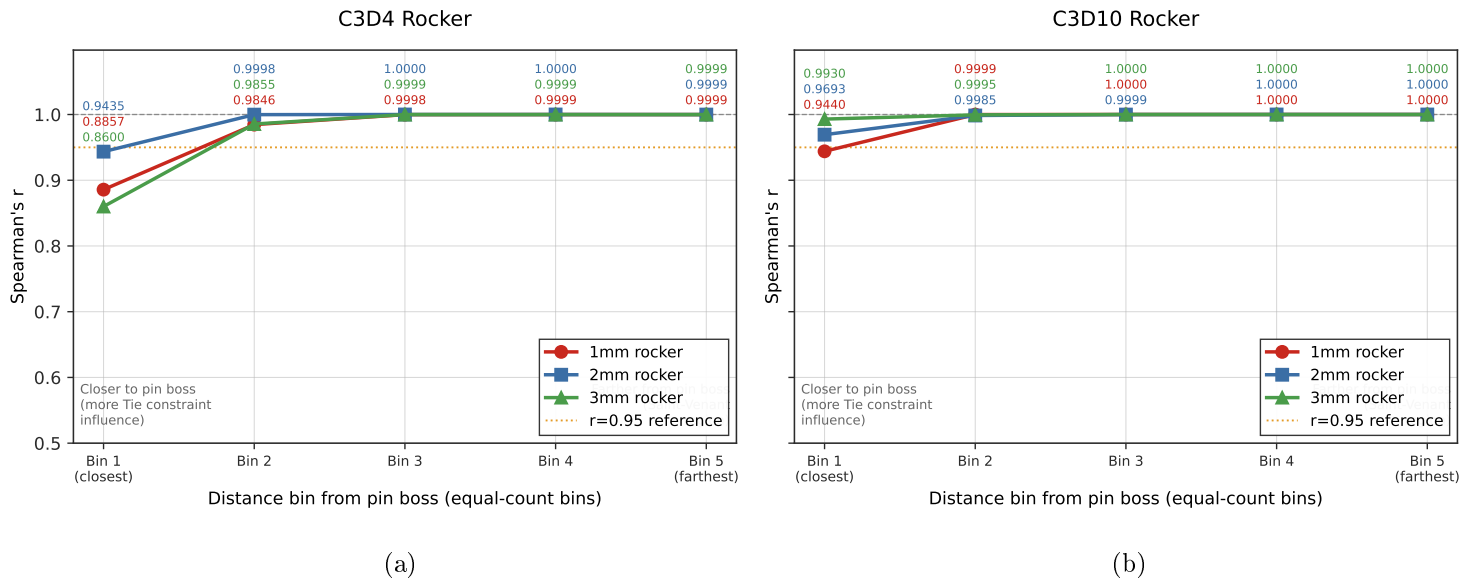


Figure 5.9: Spearman correlation computed separately in equal-count bins by distance from the pin bore, for each element mesh size used on the Rocker, for both C3D4 (a) and C3D10 (b) elements. The correlation is reduced only in the bin nearest the bore and recovers abruptly from the second bin outward, confirming that the coarse-rocker perturbation is spatially confined to a thin shell around the connection, in keeping with Saint-Venant's principle.

Sensitivity Scatter: Reference (4mm) vs Test Rocker Mesh

Variant: C3D4 | coloured by distance bin from pin boss

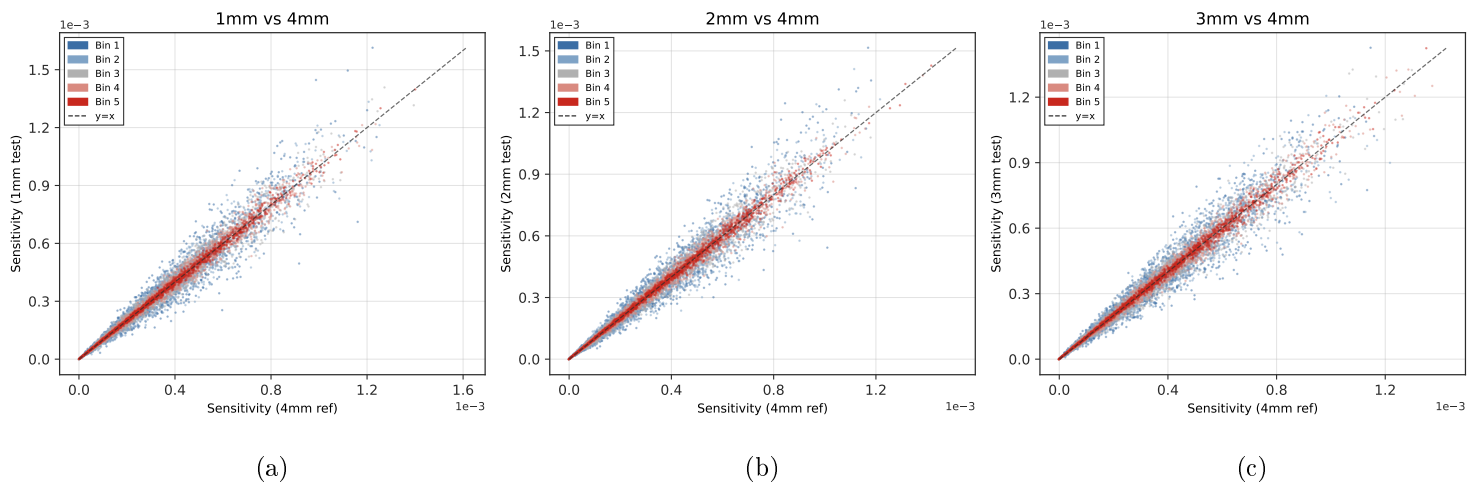


Figure 5.10: Per-element sensitivity values for the linear (C3D4) rocker variant, each test rocker mesh against the 4 mm reference, coloured by distance bin from the pin bore. The points cluster on the line of equality, with the visible scatter confined to the elements nearest the bore; the high-sensitivity elements most relevant to the optimiser sit on the line.

Sensitivity Scatter: Reference (4mm) vs Test Rocker Mesh

Variant: C3D10 | coloured by distance bin from pin boss

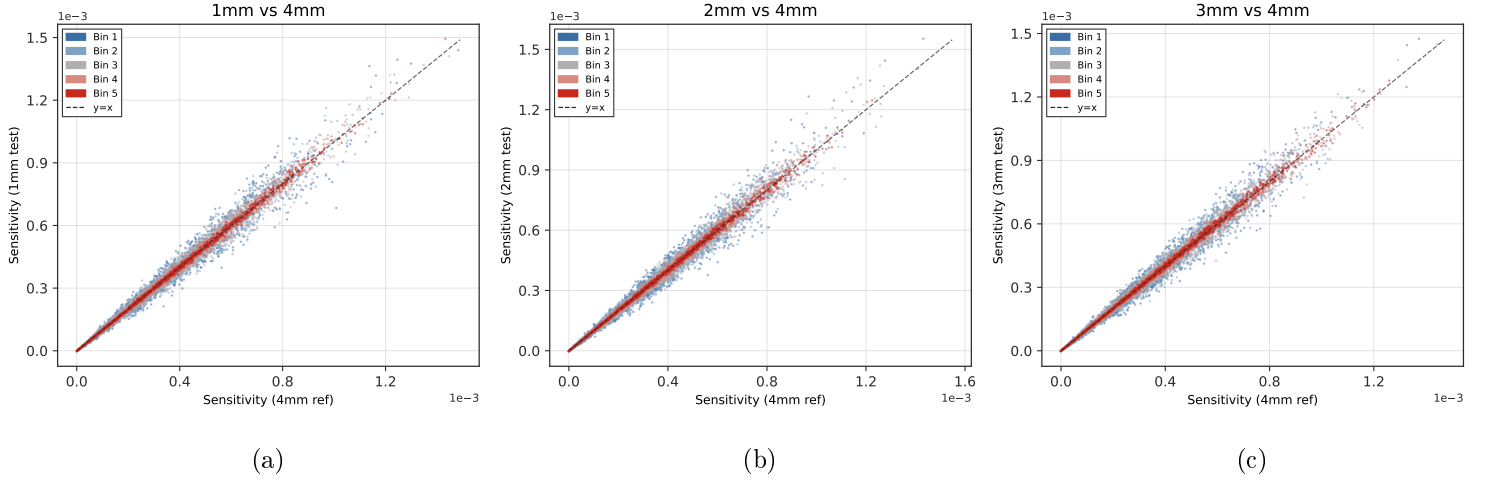


Figure 5.11: Per-element sensitivity values for the quadratic (C3D10) rocker variant, each test rocker mesh against the 4 mm reference, coloured by distance bin from the pin bore. The scatter is tighter than for the linear variant, consistent with its higher rank correlations, and is again confined to the elements nearest the bore.

The conclusion is that the coarse rocker, though it cannot give a converged displacement, gives the same sensitivity ranking within the component as a fine one, and so leads the optimiser to the same design. The 4 mm rocker is therefore justified for the optimisation run, not because it is numerically exact, which it is not, but because the optimiser is a comparative algorithm to which only the ranking is material.

5.3.3 Adopted mesh

On this basis the optimisation run used a 4 mm rocker meshed with linear tetrahedra, of 18379 elements, and a 1 mm component meshed with linear tetrahedra, of 1708873 elements. The component mesh is deliberately the finer of the two, and is kept fine for a reason beyond ordinary accuracy: the filter radius that sets the smallest member the optimiser can resolve is tied to the element size (Section 4.1), so a finer component mesh permits finer structural features and, with them, a more efficient optimised part. The rocker, which is a supporting body the optimiser does not modify and which the rank study has shown may be coarse without consequence for the result, is meshed coarsely to keep each of the run's many solves affordable.

5.4 Optimisation Results

With the model and its discretisation fixed, the optimisation was run. Because the engineering aim is a specified reduction in mass rather than an abstract volume target, the run was organised as a sequence of stages, each removing material from the design domain down to a successively lower target expressed as a fraction of the original steel part's mass: 80%,

50%, 40%, 35% and finally 30%. Figure 5.12 traces the specific stiffness of Equation (4.5) across these stages. It rises as material is removed: each successively leaner target reaches a higher specific stiffness than the last, the structure becoming more mass-efficient as the optimiser strips away the material that carries least, and the 30 per cent stage attains the highest specific stiffness of the five. On this measure alone the search would continue further still, since specific stiffness is still climbing when the sweep ends.

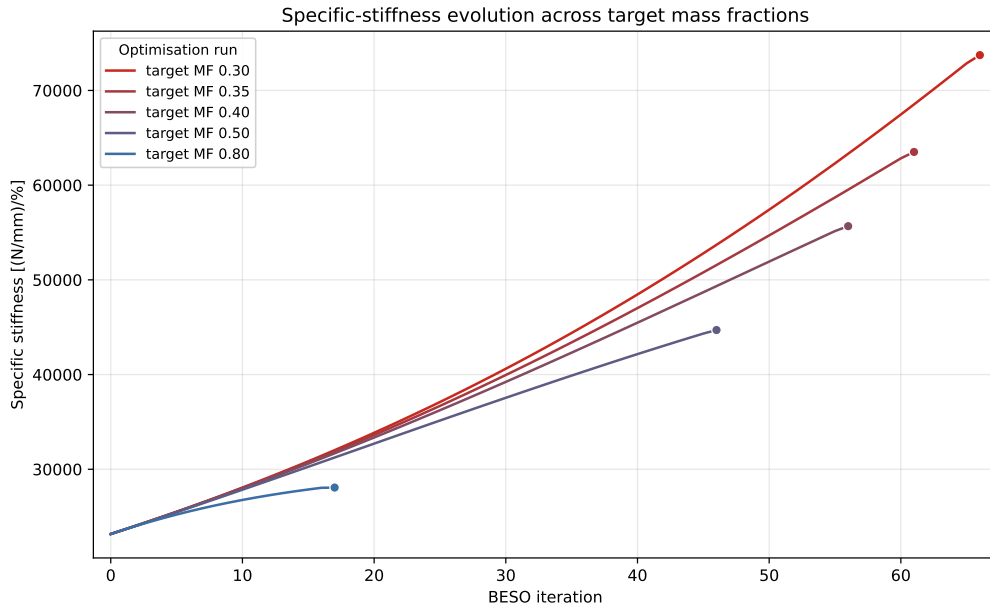


Figure 5.12: The specific stiffness of Equation (4.5) across the five optimisation stages, from a target of 80% of the original mass down to 30%. Read comparatively, the specific stiffness rises as material is removed, each leaner target reaching a higher value than the last; the 30% stage attains the highest specific stiffness of the five.

What stops the sweep is not specific stiffness but stress, and two limits meet at the 30% target. The first is revealed by the static validation of Section 5.6, run at each stage: below this mass fraction the material around the counterbore holes, where the guide-rod loads enter, begins to approach the admissible stress, so that removing more material there would push the part beyond its allowable. The second is manufacturability. The arm-like members that the optimiser forms to carry the counterbore loads back into the body of the bracket grow gradually more slender as the target tightens, and although they remain of adequate thickness at the 30 per cent target, reducing the target further would begin to bring them into a range in which fine features become more susceptible to thermal distortion during the build and the subsequent heat treatment, as discussed in (Section 3.2) of Chapter 3. The target was therefore fixed at 30% of the original mass, the leanest design that keeps both the counterbore stresses within the admissible and the load-carrying members thick enough to build reliably.

The design at this target retains a volume of $1.72 \times 10^5 \text{ mm}^3$, a fraction 0.3094 of the $5.529 \times 10^5 \text{ mm}^3$ design domain. At the AlSi10Mg density of 2.67 g/cm^3 quoted by the

manufacturer [117] this is a mass of 0.459 kg, against the 1.520 kg of the original steel part, a reduction of very nearly 70%. It is worth being clear about where that reduction comes from, because the two contributions are of very different size. The larger by far is the change of material: replacing steel with an aluminium alloy of roughly a third the density would, at the same volume, already remove about two thirds of the mass. The optimisation contributes the rest, trimming the part to some 11% below the original volume, but its more important role is the one that makes the substitution possible at all. As Section 5.1.2 set out, it was the eccentric, bending-dominated load path that forced the original part to be steel; by placing the aluminium precisely along that load path, the optimiser produces a part that carries the bending in the lighter alloy, which a solid aluminium redesign could not. The mass is saved by the substitution, but it is the optimisation that licenses it. The realisation of this optimised design as a manufacturable body, and its structural validation, are the subjects of the two sections that follow.

5.5 Optimised Geometry

The solid-void engine returns its result as a list of retained elements rather than as a surface, and turning it into a body that can be manufactured and re-analysed is the task of the post-processing described in (Section 4.6.1) of Chapter 4. The boundary of the retained set, taken directly from the optimiser, is the irregular faceted surface left by the element faces of the analysis mesh (Figure 5.13a): faithful to the optimised layout but far too rough, in its raw form, to represent a real part. It was smoothed with the shrink-free Taubin filter of Section 4.6.1 [110], over 200 passes at $\lambda = 0.685$ and $\mu = -0.7$, which removes the faceting without the systematic volume loss that a plain Laplacian smoothing would cause and while holding the non-design mating faces fixed (Figure 5.13b). The smoothed surface was then re-meshed in FEMAP, so that the validation of the next section runs on a clean, regular mesh free of the artefacts the optimisation boundary carries, rather than on the faceted surface itself.

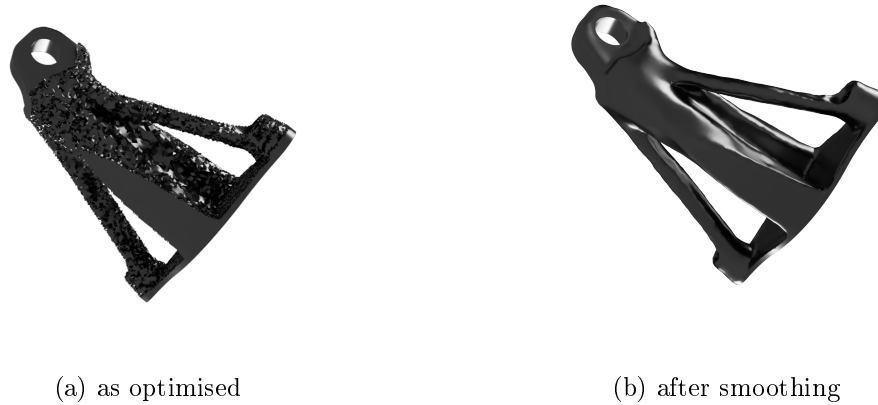


Figure 5.13: The optimised roll-damper bracket. (a) The raw boundary of the retained element set, the irregular faceted surface left by the analysis mesh. (b) The same geometry after 200 passes of the shrink-free Taubin filter ($\lambda = 0.685$, $\mu = -0.7$), which removes the faceting without volume loss and while preserving the non-design mating faces, giving the manufacturable surface the discrete layout stands for.

5.6 Structural Validation

The optimised bracket must finally be shown to meet its structural requirements, which the optimisation does not by itself guarantee: the search maximises specific stiffness under a mass target, not a stress limit, so the resulting stresses must be checked directly. This was done in two parts. A static analysis, reported first, verifies that the part sustains the peak load of each case within the admissible stress; a fatigue assessment, reported second, verifies that it also survives the repeated loading of a representative duty cycle.

5.6.1 Validation model

The static and fatigue analyses use the same finite-element model as the optimisation, described in Section 5.2.2: the same assembly of bracket and rocker, the same hinge at the pin, the same rod connectors and load-introduction couplings, and the same three load cases. The optimised geometry, re-meshed as described in Section 5.5, replaces the design domain, and one thing else changes. Where the optimisation could afford only a coarse rocker, the validation is a small number of analyses rather than a long iterative search, so the rocker is meshed at the highest fidelity: quadratic modified tetrahedra (C3D10M) at a 1 mm element size, chosen because the modified quadratic tetrahedron behaves well in the contact-like conditions at the hinge interface, and because, the validation no longer being cost-constrained, there is no reason to accept the coarse-rocker artefact of Section 5.3.1 when it can simply be resolved. The optimised component itself is meshed at approximately 1 mm; the re-meshing produced linear tetrahedra (C3D4) for it, which is the discretisation the meshing tool was able to generate for the smoothed geometry.

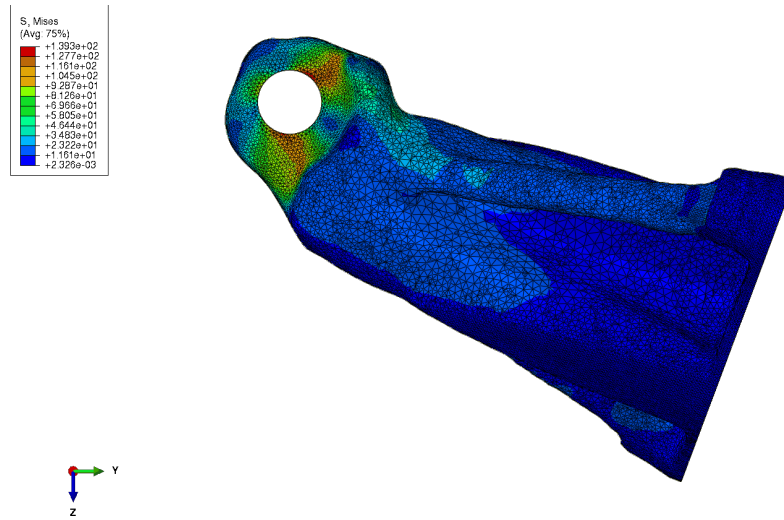
Before drawing conclusions from the model, its load path was confirmed to be physically

closed, using the connector outputs. The force and moment carried by each connector, that is the connector total-force and total-moment histories (CTF1 to CTF3 and CTM1 to CTM3), were requested as history output over the run and reported in each connector's local frame, so that every joint reports what it transmits along its own axes. On the pin, the moment about the pin axis (CTM1, the released rotational component of the revolute part of the hinge connection) was zero in every load case while the two transverse moments (CTM2 and CTM3) were not, confirming that the hinge behaves as the physical bush, free to rotate about the pin yet resisting tilt, and that its release axis is correctly oriented; the pin forces (CTF1 to CTF3) combined to an in-plane resultant of the same order as the applied load, confirming a real load path through the joint. The rod forces were read from their grounded anchors, where the reaction (the reaction force RF at the fully restrained far node) equals the axial force each rod carries, and both returned substantial values, confirming that the rods actively share the reaction as intended. Summed as vectors, the pin and rod forces balanced the applied load to within a small fraction of 1%, an independent confirmation that the load path from the applied load through the pin and rods to ground is closed.

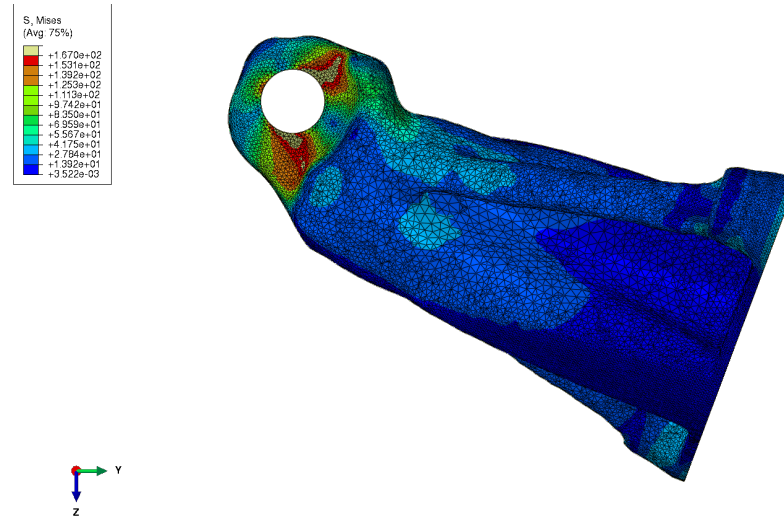
5.6.2 Static verification

The admissible stress for the static check is 167 MPa, the 250 MPa yield of the AlSi10Mg in its worst, build-direction orientation quoted by the manufacturer [117], reduced by a coefficient of safety of 1.5. Each of the three load cases was solved at its peak load and its von Mises field capped at this admissible value; the three fields are shown in Figure 5.14. Across effectively the entire volume of the part, in every load case, the stress lies below the admissible. The most severe case is the right roll, in which the roll-damper load bears directly on the planar face and is reacted through the pin across the eccentric lever arm of Section 5.1.2; its stress peaks at the pin bore, where it just exceeds the admissible over a small region, so that the part is, by this measure, sized by the right-roll case. The left roll, whose load enters more indirectly through the guide-rod bolts, peaks at about 139 MPa, and the heave case, in which the roll damper is not loaded at all, at about 80 MPa. In all three the pin bore is the most stressed region, as the eccentric load path leads one to expect.

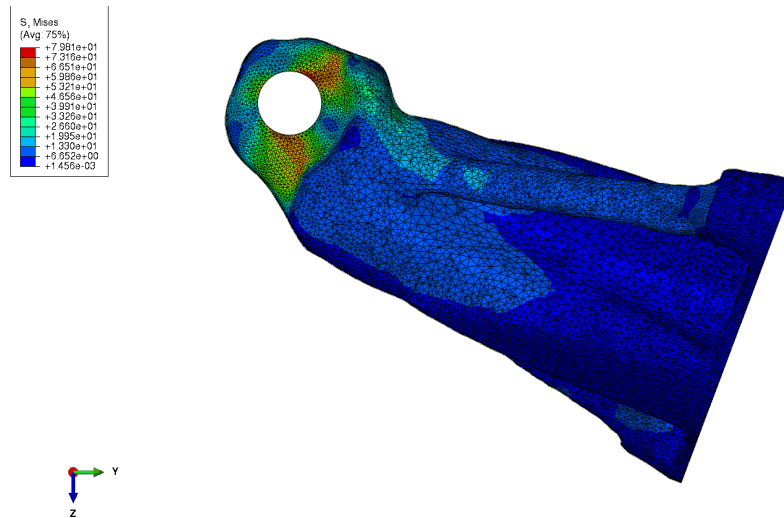
There is a single exception to the admissible being met. In the right-roll case a small region immediately adjacent to the pin bore rises above the admissible, appearing as the capped region in Figure 5.14b. This exceedance is highly localised and, tellingly, is not distributed symmetrically around the bore as a genuine pin-bearing concentration would be. A true bearing stress at a pin-loaded hole is governed by the direction in which the pin presses against the bore and would track the loaded contact arc; the observed peak instead keys to a geometric feature, appearing only on the side of the bore associated with a smaller-diameter internal shoulder, a step about 2 mm wide, and being essentially absent on the opposite side. This asymmetry, keyed to a feature rather than to the load direction,



(a) left roll



(b) right roll



(c) heave

Figure 5.14: Von Mises stress in the optimised bracket at the peak load of each case (Rocker was hidden). (a) Left roll, peak about 139 MPa. (b) Right roll, the governing case, peaking at the pin bore at the admissible 167 MPa; the small capped region on the shoulder side of the bore marks the localised exceedance discussed in the text. (c) Heave, peak about 80 MPa. In every case the pin bore is the most stressed region.

is visible in the close-up of Figure 5.15, and is the signature of a stress raised by the way the feature interacts with the load-introduction model rather than by a true overload of the material.

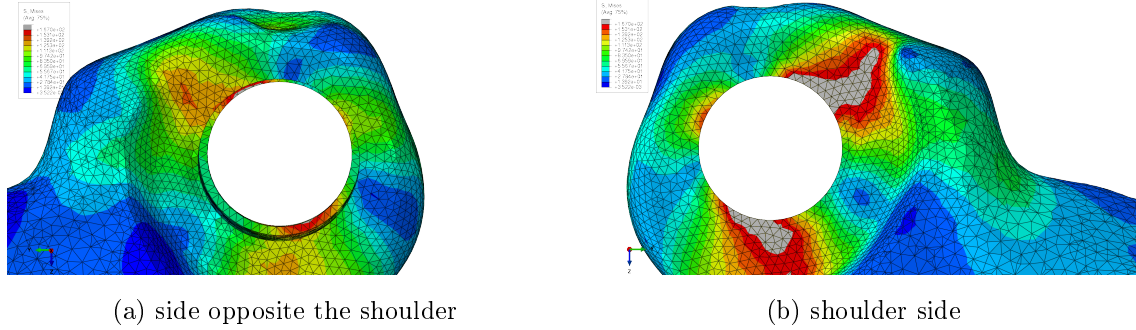


Figure 5.15: Close-up of the pin bore in the right-roll case, the contour scale capped at the 167 MPa admissible so that grey marks the exceedance. (a) On the side of the bore opposite the internal shoulder the stress remains below the admissible around the whole arc. (b) On the shoulder side a small capped region appears. A genuine pin-bearing concentration would track the loaded contact arc and be broadly symmetric about it; that the exceedance instead keys to the shoulder, appearing on one side and essentially absent on the other, is the evidence that it is governed by the interaction of that feature with the load-introduction model rather than by a true overload.

Several aspects of the idealisation support that reading and suggest the peak overstates the stress the physical part carries. The pin bore is deliberately oversized relative to its mating holes, with a bush occupying the clearance to permit the rotation the hinge represents, so the real load enters as a distributed bearing pressure over a compliant interface rather than as a coupling applied directly to the meshed bore surface; the coupling cannot reproduce that compliant, distributed introduction and tends to concentrate load at the surface it acts on. The shoulder itself is a sharp internal step, a geometric stress raiser in its own right, and it sits precisely in the load-introduction region, so that a modelling discontinuity and a geometric one coincide. Taken together, the localisation of the exceedance to a single small feature, its asymmetry relative to the load direction, and its coincidence with both a geometric notch and a load-introduction interface indicate that the region is not representative of genuine yielding, and that the reported peak is influenced to a meaningful degree by the modelling of the connection rather than by the true load-carrying capacity of the part. Subject to that one qualification, the optimised bracket satisfies the static requirement across its volume in all three load cases.

5.6.3 Fatigue assessment

The static verification of Section 5.6.2 confirms that the optimised rocker bracket sustains the absolute maximum predicted loads with the required coefficient of safety of 1.5. Static strength alone, however, does not guarantee durability. A component may survive a single application of a load and still fail after many repetitions of a smaller load, because sub-

critical damage accumulates cycle by cycle until a crack initiates and propagates [79]. As a suspension component experiences the track loads repeatedly throughout its service life, a fatigue assessment is required in addition to the static check. This subsection reports a stress-life (S-N) fatigue assessment of the optimised bracket under a representative track duty cycle, using linear-elastic stress superposition, rainflow cycle counting, mean-stress correction and linear damage accumulation.

5.6.3.1 Load history and sign-convention validation

The duty cycle was supplied by the team as a simulated flying lap of the Spa-Francorchamps circuit, computed with the Canopy lap-time simulator [118]. Canopy solves the vehicle trajectory as an optimal-control problem, its Dynamic Lap simulation returning a dynamically consistent state history that traverses the lap in minimum time. The export is sampled at approximately 14 Hz along the 6.861 km lap (lap time 134.96 s) and contains, among other channels, the roll-spring, heave spring and push-rod forces, together with the lateral and longitudinal accelerations, speed and steering angle. The front-axle channels are used throughout, as the front push rod carries the higher peak force and this choice is consistent with the static verification, which was also driven by the (larger) front roll load.

The sign of the roll-spring channel was verified before use, since it drives the switching of the roll load path described below. The roll-spring force correlates almost perfectly with the lateral acceleration (correlation coefficient of 1.00), so its sign encodes the direction of the corner. This interpretation was validated against the known geometry of the circuit, as shown in Figure 5.16: the channel is positive through every right-hand corner (La Source, Raidillon, the entry to Les Combes, Rivage and the entry to the Bus Stop) and negative through the sustained left-handers (Pouhon and Blanchimont). A positive roll force therefore corresponds to a right-hand corner, in which the roll damper reacts against the planar face of the bracket (reference point RP-5), and a negative roll force corresponds to a left-hand corner, in which the load is transmitted through the two counterbore bolts (reference points RP-3 and RP-4). These two situations load the bracket through physically distinct paths and are treated as separate load cases in the reconstruction that follows.

5.6.3.2 Stress reconstruction and damage model

The stress history at every node of the bracket is reconstructed from four linear-elastic unit load cases, each computed at a reference force $F_{\text{ref}} = 1000 \text{ N}$ and giving a nodal stress tensor field: roll through the counterbore bolts (roll-left, σ_{rL}), roll through the planar face (roll-right, σ_{rR}), push rod (σ_{p}) and heave (σ_{h}). Because the analysis is linear elastic, the stress tensor at any instant is the sum of these reference fields, each scaled by the ratio of the instantaneous force to the reference force, with the roll contribution switched between the two paths according to the sign of the roll channel,

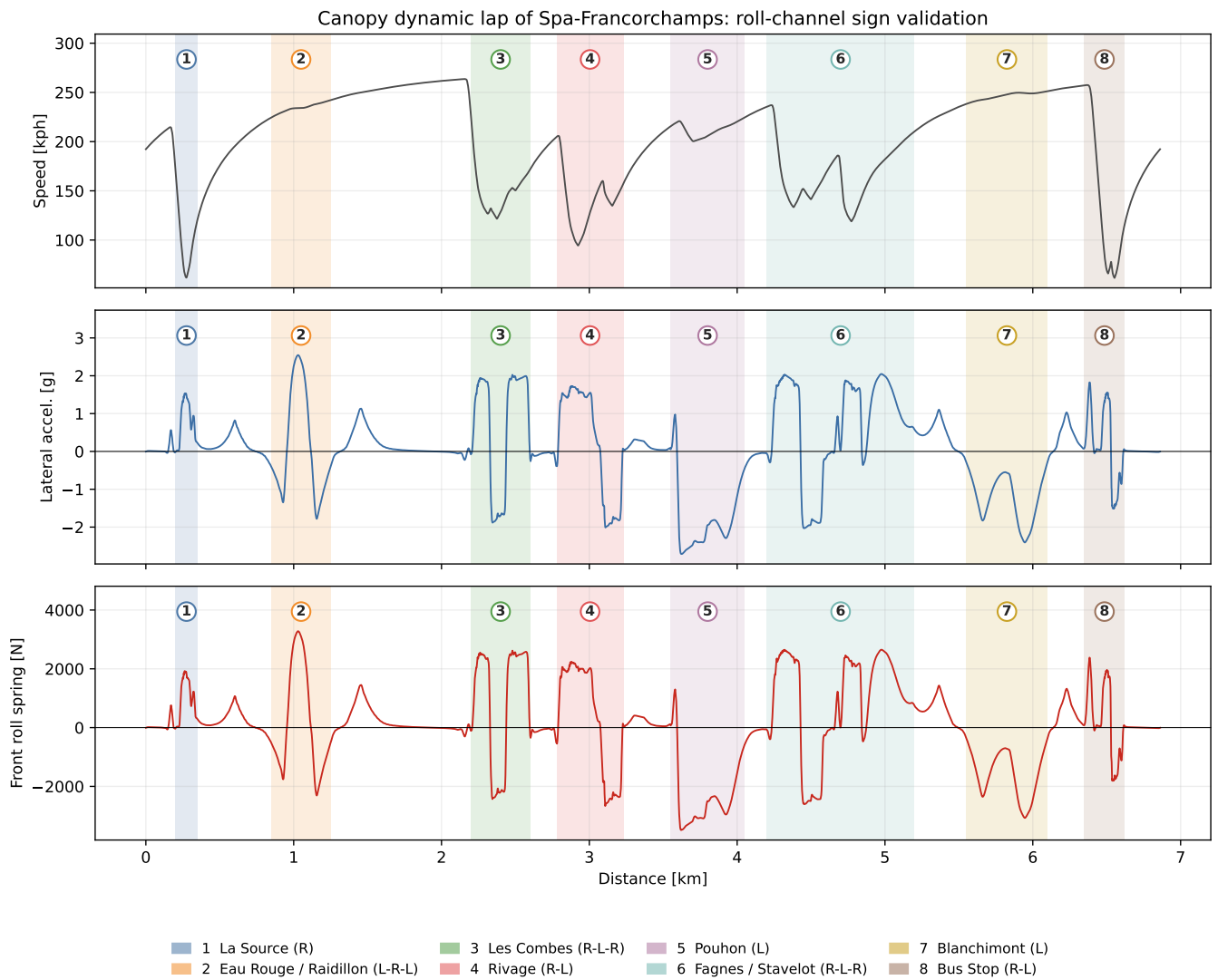


Figure 5.16: Validation of the roll-spring sign convention against the circuit layout. From top to bottom: vehicle speed, lateral acceleration and front roll-spring force along the lap. The numbered bands mark the main sections of the circuit, identified in the key below the figure. The roll-spring force changes sign exactly at each change of corner direction, positive for right-hand corners and negative for left-hand corners, confirming that its sign can be used to select the roll load path.

$$\boldsymbol{\sigma}(\mathbf{x}, t) = \frac{F_{\text{roll}}^-(t)}{F_{\text{ref}}} \boldsymbol{\sigma}_{\text{rL}}(\mathbf{x}) + \frac{F_{\text{roll}}^+(t)}{F_{\text{ref}}} \boldsymbol{\sigma}_{\text{rR}}(\mathbf{x}) + \frac{F_{\text{push}}(t)}{F_{\text{ref}}} \boldsymbol{\sigma}_{\text{p}}(\mathbf{x}) + \frac{F_{\text{heave}}(t)}{F_{\text{ref}}} \boldsymbol{\sigma}_{\text{h}}(\mathbf{x}), \quad (5.1)$$

with $F_{\text{roll}}^- = \max(-F_{\text{roll}}, 0)$ and $F_{\text{roll}}^+ = \max(F_{\text{roll}}, 0)$, so that only one roll path is active at a time. The push-rod and heave loads are always present, entering the bracket indirectly through the rocker pin. Superposition is performed at tensor level, that is, all six stress components are summed before any equivalent stress is computed.

The tensor history is reduced to a single scalar per node by a signed von Mises stress. The von Mises magnitude,

$$\sigma_{\text{vM}} = \sqrt{\frac{1}{2} [(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2] + 3(\sigma_{12}^2 + \sigma_{13}^2 + \sigma_{23}^2)}, \quad (5.2)$$

is given the sign of the first stress invariant so that the alternation between tension and compression, which fatigue is sensitive to, is preserved,

$$\tilde{\sigma} = \text{sgn}(I_1) \sigma_{\text{vM}}, \quad I_1 = \sigma_{11} + \sigma_{22} + \sigma_{33}. \quad (5.3)$$

The resulting scalar history is decomposed into closed cycles by the rainflow method of Section 3.3, which returns for each cycle an amplitude σ_{a} , a mean σ_{m} and a count n . A safety factor $j = 1.75$ is applied to the amplitude of every cycle. This value follows the philosophy of the FKM guideline [90], which assigns higher partial safety factors to materials whose fatigue behaviour is governed by process-induced defects and to components whose failure has severe consequences. The fatigue strength of laser powder-bed AlSi10Mg is defect-driven, controlled by porosity and surface condition in a manner analogous to a cast alloy [88], and a suspension bracket is a safety-critical component, so a factor above the static value of 1.5 is warranted. A value of 1.75 is adopted on the understanding that the finished part is subject to quality control appropriate to a defect-driven material, such as non-destructive inspection of the critical regions and control of the build parameters and surface finish, so that the large defects that would undermine the assumed fatigue strength are detected. The factor is applied to stress rather than to life, which is the more conservative choice, and matches the way the guideline reduces the permissible fatigue strength.

Each amplitude is corrected for its mean stress to obtain an equivalent fully reversed amplitude, using the Goodman and Soderberg constructions of Equation (3.9) of Chapter 3, with the safety factor j applied to the amplitude so that $j \sigma_{\text{a}}$ is substituted for σ_{a} in the correction. The material yield of 250 MPa in the worst build orientation is used for the yield strength σ_{y} and 330 MPa for the ultimate strength σ_{u} , and compressive means are conservatively not credited, σ_{m} being set to zero when negative.

The number of cycles to failure at each corrected amplitude follows from the Basquin

law of Equation (3.8) of Chapter 3, inverted for N_f and with a surface-finish factor k_s scaling the strength coefficient, so that $N_f = (\sigma_{ar}/k_s a)^{1/b}$. Its coefficients are taken from the experimental characterisation of laser powder-bed AlSi10Mg by Sajadi et al. [91] at a load ratio of $R = -1$. The headline assessment uses the T6 heat-treated curve ($a = 1710 \text{ MPa}$, $b = -0.19$, giving a fatigue strength of 80 MPa at 10^7 cycles), which matches the required condition for the manufacturing of the optimised part. Aluminium alloys do not exhibit a true fatigue limit, so the Basquin line is extrapolated below the tested range, which is conservative. The fatigue hot spot is the pin bore, which should be machined or reamed for the pin fit; the Sajadi curve is itself a machined-surface curve, so $k_s = 1$ is appropriate for the damaging location. As-built surfaces elsewhere carry a lower fatigue strength [89], but they are not where the damage concentrates, so finishing the bore is both standard practice and the correct basis for the assessment.

Damage is accumulated linearly by the Palmgren-Miner rule of Equation (3.10) of Chapter 3; since the damage is evaluated over one lap, the life follows as its reciprocal, $L = 1/D$ laps.

The fatigue hot spot is not assumed to coincide with the static hot spot, because the most damaging location depends on the size of the stress swings rather than on the peak stress. The whole procedure is therefore applied at every node of the bracket, in two passes for efficiency. A first pass ranks all nodes by a rapid one-cycle damage estimate, taking the largest stress swing of each node over the lap as a single equivalent cycle. A second pass performs the full rainflow counting, mean-stress correction and Miner summation on the highest-ranked candidate nodes, from which the node of maximum damage is identified as the hot spot. The procedure was implemented in Python.

5.6.3.3 Results

The full-field scan identifies the fatigue hot spot at the pin bore, coinciding in this case with the static hot spot. At this location the signed von Mises stress varies between about -34 and $+39$ MPa over the lap, a range of roughly 73 MPa, well below the 250 MPa yield. The rainflow decomposition returns about 146 cycles per lap, of which the great majority are small (around 128 cycles have amplitudes below 5 MPa) and only two are large. The largest cycle has an amplitude of about 36 MPa at a mean of only 2.5 MPa, that is, it is very nearly fully reversed; the push-rod and heave contributions dominate the stress magnitude while the reversing roll load, entering directly through the bolts and face, largely sets the alternation.

Figure 5.17 summarises the assessment at the hot spot. Panel (a) shows the reconstructed stress history over the lap for the hotspot node. Panel (b) is a Haigh diagram in which each point is one rainflow cycle, positioned by its mean and amplitude and coloured by its damage, compared against the Goodman and Soderberg constant-life boundaries drawn at the 10^7 -cycle endurance level; the cycles cluster near zero mean and low amplitude, far inside both boundaries. Panel (c) is the damage spectrum, obtained by sorting

the corrected cycle amplitudes into eighteen bins and summing the damage contributed by the cycles in each bin; it shows that essentially all of the damage comes from the few large-amplitude cycles, the many small ones being negligible. Panel (d) places the cycles on the T6 S-N curve; most fall so far to the right, at such high predicted lives, that they lie beyond the plotted range and are non-damaging. The spatial distribution of damage is shown in Figure 5.18, which confirms that damage is confined to the material around the pin bore and the adjacent loaded interfaces.

The resulting life is very long. On the T6 curve, with the 1.75 safety factor applied to the cycle amplitudes, the Goodman assessment gives a life of about 1.42×10^7 laps, equivalent to roughly 9.75×10^7 km, and the Soderberg assessment 1.40×10^7 laps. Because these mean stresses are so small, the two mean-stress corrections barely differ: at the dominant cycle they separate the corrected amplitude by only about 0.25%, a difference the Basquin power law amplifies, through its exponent, into the roughly 1.5% that separates the two lives just quoted. The Goodman boundary is drawn with the ultimate strength of 330 MPa that accompanies the S-N curve of Sajadi et al. [91], so that the mean-stress correction rests on the same characterisation as the life curve it is paired with; using the manufacturer's 310 MPa instead would move the Goodman line imperceptibly towards the Soderberg one, with no bearing on the outcome, and the assessment is in any case carried on the more conservative Soderberg estimate. These figures exceed any realistic service by several orders of magnitude, so the assessment shows that the optimised bracket is not fatigue-limited under this duty cycle and that its design is governed by the static strength requirement. The physical explanation is that the bracket is sized by the absolute maximum damper loads, of the order of 30 kN, at a coefficient of safety of 1.5, whereas a smooth representative lap only ever demands a fraction of those forces, leaving the operating stresses deep in the high-cycle regime. The one condition attached to this conclusion is that the pin bore should be finished by machining or reaming, as assumed above; this is standard practice for a bore that must accept a pin, and it is the correct basis on which the fatigue strength has been evaluated.

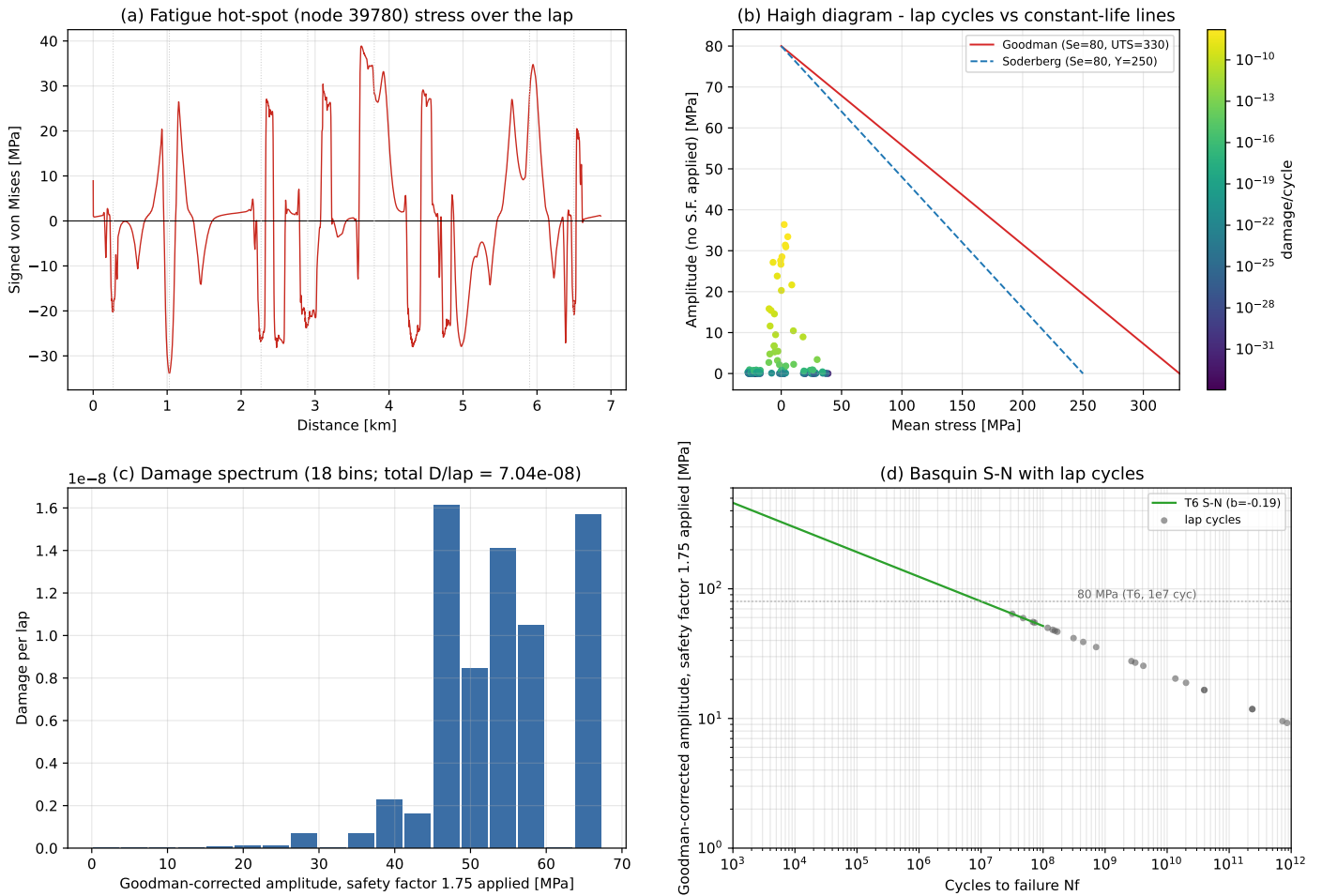


Figure 5.17: Fatigue assessment at the hot-spot node. (a) Reconstructed signed von Mises stress over the Spa lap. (b) Haigh diagram of the rainflow cycles (points coloured by damage per cycle, no safety factor applied to the plotted amplitudes) against the Goodman and Soderberg constant-life boundaries at the 10^7 -cycle endurance level. (c) Damage spectrum, the summed damage per lap in each of eighteen bins of Goodman-corrected amplitude with the safety factor of 1.75 applied. (d) The lap cycles placed on the T6 Basquin S-N curve, with the 80 MPa endurance stress marked; cycles beyond the plotted range are non-damaging.

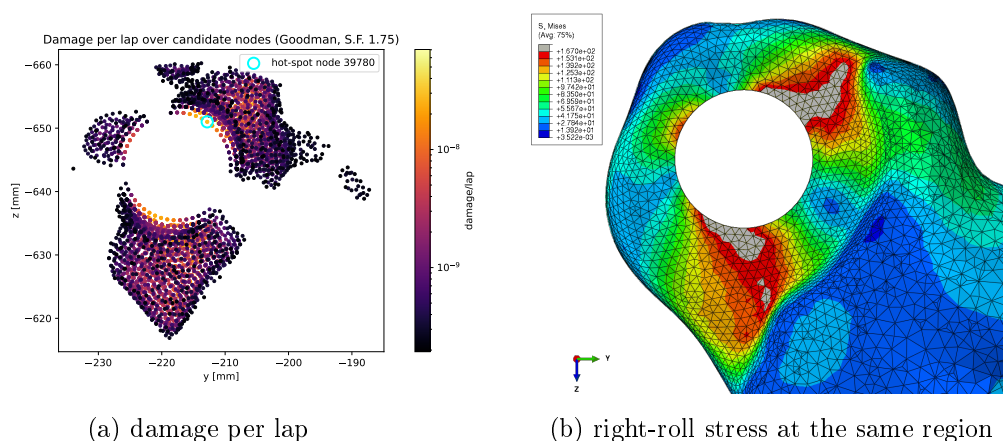


Figure 5.18: (a) Damage per lap over the candidate nodes, projected on the y - z plane with the vertical axis oriented as in the finite-element model; the hot-spot node is circled. (b) The right-roll von Mises field over the same region of the bore. The fatigue candidate nodes fall in the same material around the pin bore as the static concentration of the governing right-roll case, so that the static and fatigue assessments are governed by one and the same region of the part.

5.7 Discussion

This chapter has taken the optimisation framework of Chapter 4 from benchmark problems of known solution to a genuine, performance-critical component of a car, and the study supports several conclusions.

The first is that the exercise succeeded on its own terms. The roll-damper bracket, originally a 1.520 kg steel part, has been redesigned as a 0.459 kg part in selective-laser-melted AlSi10Mg, a reduction of very nearly 70%, and the redesigned part meets its static requirement across its volume, subject only to the localised and demonstrably artefactual exceedance at the pin bore, and is shown to be far from fatigue-limited under a representative Spa-Francorchamps duty cycle, its life exceeding any realistic service by orders of magnitude. The part is governed by static strength, and specifically by the stress at the counterbore holes and pin bore where its loads enter, which is what fixed the mass target where it was fixed.

The second concerns how the saving was achieved, because the mechanism is instructive. Most of the mass was removed not by the optimiser directly but by the change of material it enabled: the bending-dominated, eccentric load path that had forced the original part to be steel was carried into a light alloy by placing that alloy precisely along the load path, which is exactly the redistribution the method exists to perform. The optimisation and the material substitution are therefore not two separate savings but one, since the substitution is only admissible because the optimisation makes the aluminium part strong enough, and the direct volume saving of the optimiser, some 11%, is modest beside it. One point of detail carries a specific recommendation. To improve the load introduction at the pin bore, the boss was thickened during the design so that its clearance to the rocker clevis

is reduced to 1 mm, against the 2 mm margin held elsewhere (Section 5.2.1). This is a deliberate and acceptable local reduction, but it leaves little tolerance for build deviation and assembly at that interface, so it is recommended that the boss clearance be verified against the as-built dimensions and the rocker tolerances before the part is committed, and that the pin bore itself be machined or reamed to its final fit after building, which the fatigue assessment already assumes.

The result carries the caveats its idealisations imply. The optimisation model, for reasons of cost, used a coarse rocker whose displacement does not converge, and it is only the comparative nature of the evolutionary algorithm, established quantitatively by the rank study of Section 5.3, that makes this admissible; the topology is trustworthy. The static exceedance at the pin bore, argued above to be a load-introduction artefact, would be worth confirming with a model that represents the bush and pin as a contact pair rather than as a coupling, which would replace the argument with a direct demonstration. And the fatigue assessment rests, as it states, on the pin bore being finished by machining or reaming, which is standard practice for a bore that must accept a pin and is the basis on which its fatigue strength was evaluated.

On this basis the recommendations to the partner are straightforward. The optimised bracket is suitable to build in AlSi10Mg with T6 heat treatment, at a mass saving of about 70% over the steel original, with the pin bore machined to its final fit after building. Because the geometry was fixed against the larger rear-axle roll range while being loaded with the higher front-axle damper forces, the same part serves both ends of the car, which halves the number of distinct components to qualify and build. Where a further saving is sought, the framework's lattice engine offers a route not pursued here: a graded lattice core within a solid skin, of the kind verified in Section 4.7.4, would trade a little of the discrete engine's efficiency for the manufacturability and multifunctionality of a cellular interior, and would be a natural next step for a part whose static margins, outside the load-introduction regions, are comfortable. The approach is consistent with current industrial practice: additively manufactured, topology-optimised suspension components have recently reached series production in high-performance applications, including the printed suspension wishbones of the McLaren W1 [36] and the optimised structural nodes of the Czinger 21C [6]. Laser powder-bed fusion remains a comparatively expensive route to a finished part, so a saving of this kind is justified where the value placed on mass reduction is high and the production volume low, conditions that a limited-series hypercar such as the Furia satisfies. The present study demonstrates that the same methodology can be realised within a low-volume national programme using an optimisation framework developed in-house.

Chapter 6

Machine Learning Surrogate for Topology Optimization

The two preceding chapters built an optimisation framework and applied it to a real component, in both cases at the full price of the method: every candidate design is evaluated by a finite-element solve, and a converged result is reached only after many such solves in sequence. This chapter turns to the third phase of the work and to a different question, the one left open at the close of Chapter 2 (Section 2.4): whether a neural network can learn the mapping from the physical conditions of a problem to its optimised material distribution, and so stand in for the iterative optimiser with a single forward pass that returns a design in milliseconds rather than minutes.

The study pursued here is deliberately bounded, and serves two purposes at once. It is an introduction to the methods, undertaken to establish that a compact network can reproduce the solid-void optima of a verified public dataset to a high pixel-level accuracy. It is also an empirical test of a sharper hypothesis: that the representation such a network learns is not a memorised lookup but captures transferable structural knowledge, the load paths and stress trajectories that decide where a structure must run, and that this knowledge can be reused to predict the output of a different optimiser. The second optimiser is the lattice engine of Chapter 4 (Section 4.5), whose graded-density results follow from a fully-stressed-design criterion rather than from compliance minimisation, and which therefore poses a related but genuinely harder prediction problem. Whether a network pre-trained on solid-void optima adapts to that harder task more readily than one trained from nothing is the organising thread of the chapter and the subject of its closing study.

The chapter proceeds as follows. Section 6.1 frames the surrogate problem and the two prediction tasks it comprises. Section 6.2 describes the dataset on which the binary predictor is trained. Section 6.3 sets out the network and the incremental refinements that shaped it, and Section 6.4 the training procedure, loss functions and evaluation metrics. Section 6.5 reports the binary-topology results and identifies the failure modes that bound

them. Section 6.6 gathers the extensions explored beyond that baseline, the loss-function and augmentation studies that did not improve on it and the lattice fine-tuning study that is the transfer experiment proper, and Section 6.7 closes with what the surrogate does and does not establish.

6.1 Motivation and Problem Framing

Section 3.5 introduced the data-driven route in general terms; this section fixes the particular problem solved here. The premise is the one common to supervised surrogate modelling, stated in Equation (3.13): a network f_θ with parameters θ is fitted to a body of solved problems by minimising a loss over its predictions, after which it maps an unseen problem specification to a predicted optimum in a single evaluation. The appeal for topology optimisation is direct. The iterative optimiser, with its repeated finite-element solves, is replaced by one forward pass, and a result that the framework of the previous chapters reaches in minutes is returned in the time of a single network evaluation. What is given up in exchange, and the reason this remains a surrogate rather than a substitute, is any guarantee that the prediction is the true optimum or even a physically admissible structure; that question is returned to in Section 6.7.

The work addresses two related but structurally distinct prediction tasks, one for each engine of the framework. The first is solid-void prediction. Here the target is the binary topology of the classical compliance problem, in which every element is either solid or void, and the network learns *where* to place material. The second is lattice density prediction. Here the target is the graded relative-density field produced by the lattice engine, in which an element may take any value between void and fully solid, and the network must learn not only where material belongs but *how much* of it, a dense regression rather than a binary classification.

The harder of the two is the lattice task, for two reasons that recur through the chapter. The first is that its target is a continuous field rather than a two-state mask, so the network must resolve a magnitude at every element and not merely a decision. The second, and the more consequential, is that the lattice engine takes no prescribed volume fraction. The solid-void optimiser, like the SIMP optimiser that generated the public dataset, is handed a target material fraction as part of its problem specification and returns the stiffest structure that respects it; the lattice engine instead converges to whatever fraction achieves full stress utilisation, an emergent property of the fully-stressed-design criterion set out in Section 4.5. The quantity that, for the binary task, is supplied to the predictor as an input is therefore, for the lattice task, an unknown the predictor must infer from the load and stress fields alone. This asymmetry is small to state and large in consequence, and it is taken up again where it bears on the results, in Section 6.6.

The chapter’s organising hypothesis follows from the relationship between the two tasks. A network trained to reproduce solid-void optima has, if it has learned anything beyond the

surface statistics of its training set, acquired a representation of where structural material must go: the principal load paths, the stress concentrations, the connectivity that the boundary conditions demand. That representation is in principle equally relevant to the lattice task, which differs from the binary one in how much material it places and in the precise morphology of the result, but not in the underlying physics that decides where the structure runs. The hypothesis tested in Section 6.6 is therefore that a network pre-trained on the binary task provides an effective initialisation for the lattice task, adapting faster and generalising better than a network trained from scratch, and the comparison of those two routes is the empirical core of the chapter.

Throughout, the surrogate is held to be a complement to the verified optimiser rather than a replacement for it. A prediction carries no guarantee of equilibrium or connectivity, so every design route in this work remains subject to independent finite-element validation, exactly as the solid-void and lattice results of the previous chapters were. The network on which both tasks are carried out is described next, after the dataset that trains it.

6.2 Dataset

The binary predictor is trained on the dataset released with the TopoDiff framework [2], introduced in Section 2.4 as a verified public benchmark. It contains thirty thousand structural optimisation samples, each the result of a SIMP optimisation on a sixty-four by sixty-four element plane-stress mesh, so that a sample is a pairing of a problem specification with its optimised topology at a fixed resolution of 64×64 elements.

The specification is encoded as five spatial channels on that grid. The first two carry the applied load, decomposed into its horizontal and vertical components at the single loaded node, so that each is zero everywhere except at one pixel. The third is the prescribed volume fraction, a single scalar repeated across the whole grid, which records the material budget the SIMP optimiser was required to meet. The remaining two are physical fields computed by one linear finite-element analysis of the unoptimised domain, the strain energy density and the von Mises stress, which mark where the initial structure works hardest and so give the network a physics-informed prior on where material is likely to be needed. The target is the optimised topology itself, a binary field in which zero denotes solid material and one denotes void.

One property of the dataset becomes important for the augmentation study of Section 6.6 and is recorded here. The applied load never points downward. The load angle is confined to the upper half plane, so that the vertical component, the sine of the angle, is non-negative for every sample, while the horizontal component, the cosine, spans the full range from minus one to one. This was confirmed directly across all thirty thousand samples: the vertical load channel is everywhere greater than or equal to zero. The consequence, drawn out later, is that only a horizontal mirror keeps a sample within the dataset's physical convention, since a vertical mirror or a rotation would produce a downward load

that the dataset never contains.

The thirty thousand samples are split into twenty-seven thousand for training and three thousand for validation, a ninety to ten division fixed by a single random seed so that the same split is used in every run reported here and in the appendix. The problems are drawn from forty-two distinct boundary-condition configurations, distributed roughly uniformly, a structure that is otherwise unremarkable but that the augmentation study later exploits and is constrained by. With the data fixed, the network is described next.

6.3 Network Architecture

The predictor is a convolutional encoder-decoder of the U-Net family introduced in Section 3.5 and sketched in its generic form in Figure 3.12. Its concrete form, with the channel widths and the additions specific to this work, is shown in Figure 6.1. It was not arrived at in a single step but through a short sequence of refinements, each one isolated from the last and retained only where it improved the measured accuracy. The description follows that sequence, since the reason for each component is clearest when set against the limitation it was introduced to remove.

The starting point was a plain encoder-decoder without skip connections. A contracting encoder of three blocks halves the spatial resolution at each stage while widening the channels, taking the five-channel input through thirty-two, sixty-four and one hundred and twenty-eight channels down to an eight by eight feature map; a bottleneck convolution then lifts this to two hundred and fifty-six channels at the same eight by eight resolution; and an expanding decoder mirrors the encoder, upsampling back to full resolution through one hundred and twenty-eight, sixty-four and thirty-two channels before a final convolution and a sigmoid produce the single-channel output. The architectural weakness of this design is concentrated at the bottleneck. With every spatial detail compressed into an eight by eight grid, the decoder must reconstruct fine structure from abstract features alone, which blurs boundaries and erases the thinnest members. This eight by eight bottleneck recurs as the binding limit on accuracy throughout the chapter, and the refinements that follow are, in large part, attempts to work around it.

The first refinement was the addition of skip connections, which turns the encoder-decoder into a U-Net proper. Each encoder block now passes its pre-pooling feature map directly to the corresponding decoder block, where it is concatenated with the upsampled features before the convolutions. The concatenation widens the decoder inputs, to three hundred and eighty-four, one hundred and ninety-two and ninety-six channels at the three stages, and, more to the point, gives the decoder a route to high-resolution spatial detail that does not pass through the bottleneck. Members and boundaries that the plain decoder could only approximate are now reconstructed from features carried across at their original resolution. This is the single largest source of accuracy improvement in the architecture and adds about 11% to the parameter count, taking the network from roughly 1.75 million

to roughly 1.95 million parameters.

The second refinement was batch normalisation [119], inserted after every convolution in the encoder, bottleneck and decoder. By normalising the distribution of activations within each training mini-batch, it stabilises the signal passed between layers, which permits a higher learning rate, acts as a mild regulariser, and noticeably speeds convergence. Its cost in parameters is negligible, a few thousand, and its effect on accuracy is small but real.

The third and last refinement was the addition of attention gates [120] on the skip connections. A skip connection carries an encoder feature map across in full, including activations in regions the decoder has no use for at that stage; an attention gate learns to suppress those regions before the skip is concatenated. The gate takes two inputs, the decoder’s gating signal, which is at the coarser resolution and carries the context of what the decoder is currently reconstructing, and the skip features, which are at the finer resolution, and from them computes a single-channel spatial map that weights the skip,

$$\boldsymbol{\psi} = \text{sig}\left(\mathbf{W}_{\psi}^{\top} \text{ReLU}(\mathbf{W}_g \mathbf{g} + \mathbf{W}_x \mathbf{x})\right), \quad \hat{\mathbf{x}} = \boldsymbol{\psi} \odot \mathbf{x}, \quad (6.1)$$

where $\mathbf{g}$ is the gating signal from the decoder, brought to the resolution of the skip, $\mathbf{x}$ is the skip feature map from the encoder, $\mathbf{W}_g$, $\mathbf{W}_x$ and $\mathbf{W}_{\psi}$ are learned one by one convolutions, ReLU and sig are the rectified-linear and sigmoid nonlinearities, $\boldsymbol{\psi}$ is the resulting attention map with values in the unit interval, $\odot$ denotes the elementwise product, and $\hat{\mathbf{x}}$ is the gated skip that is passed on to the decoder. Three such gates are used, one on each skip connection, sized to the channels they gate. They add only a few thousand parameters yet sharpen precisely the features the task depends on, the thin members and the boundaries, by letting the decoder attend to the parts of each skip that matter for the region it is reconstructing.

This network, a U-Net with batch normalisation and attention gates and just under two million parameters, is the architecture used for every result reported in this chapter, both the binary predictions and, after the adaptation described in Section 6.6, the lattice predictions. Its training is described next.

6.4 Training

With the network fixed, its training remains to be specified: the loss it minimises, the optimiser that drives that minimisation, the metrics by which its output is judged, and the evaluation protocol and hardware on which every run was carried out.

The binary predictor is trained against a loss that combines a pixel-wise term with an overlap term,

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}(\hat{\mathbf{y}}, \mathbf{y}) + \mathcal{L}_{\text{Dice}}(\mathbf{1} - \hat{\mathbf{y}}, \mathbf{1} - \mathbf{y}), \quad (6.2)$$

where $\hat{\mathbf{y}}$ is the predicted field, $\mathbf{y}$ the target, $\mathcal{L}_{\text{BCE}}$ the mean pixel-wise binary cross-entropy, and $\mathcal{L}_{\text{Dice}}$ the Dice loss of Equation (3.14), the two terms equally weighted. The comple-

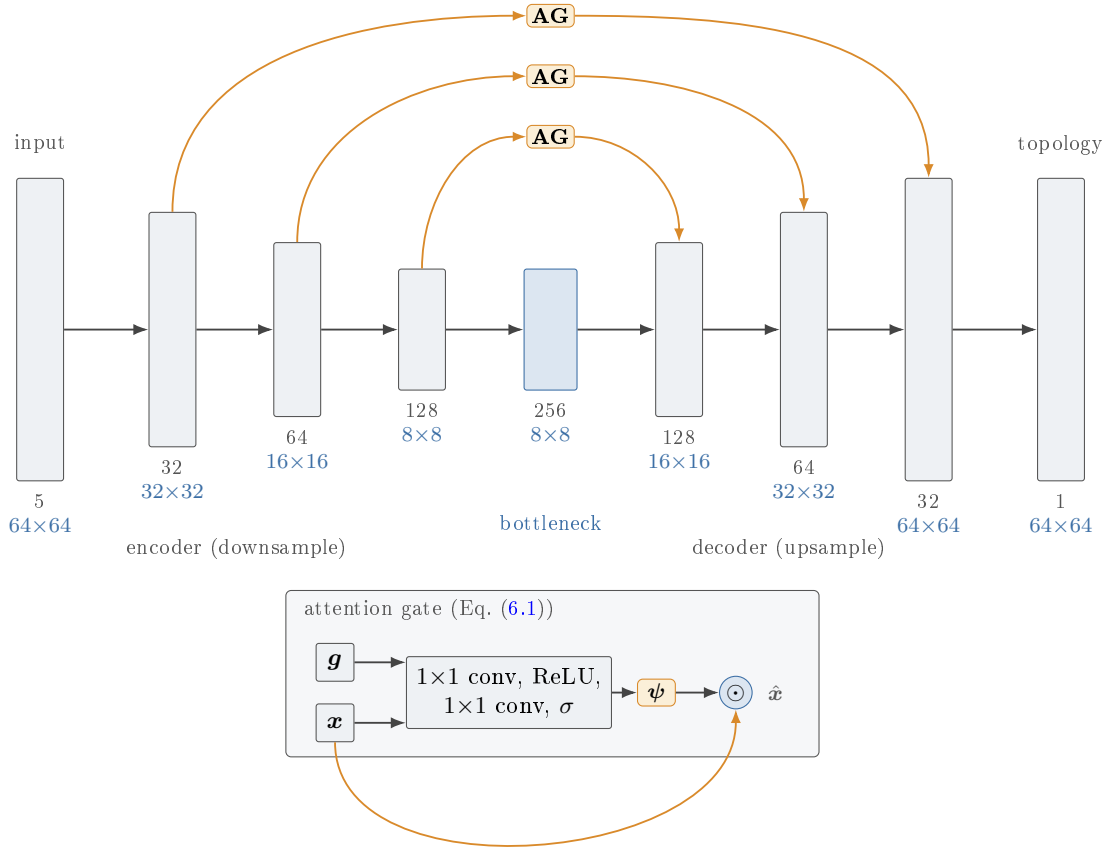


Figure 6.1: The network used throughout this chapter: a U-Net encoder-decoder with batch normalisation and attention gates on the skip connections. The encoder downsamples the five-channel input to an eight by eight, 256 channel bottleneck; the decoder upsamples back to full resolution, and each skip connection passes through an attention gate (AG) before it is concatenated. Channel counts and spatial resolutions are shown beneath each stage. The inset details the attention gate of Equation (6.1): the decoder gating signal g and the encoder skip x produce an attention map ψ that reweights x .

ment $\mathbf{1} - \hat{\mathbf{y}}$, $\mathbf{1} - \mathbf{y}$ in the Dice term selects the solid class under the convention that zero denotes material and one denotes void. The two terms answer to different weaknesses. The cross-entropy supplies a dense and stable gradient at every one of the four thousand and ninety-six pixels, but weights them equally and so is dominated by the abundant background. The Dice term is normalised by the count of material pixels, so that a slender member of a few pixels contributes as strongly to the gradient as a large solid region, which directly counters the network’s tendency to neglect the thin structure on which the accuracy ultimately turns. Their combination is what the experiments of Section 6.6 repeatedly fail to improve upon.

This loss is minimised with AdamW [121], the decoupled-weight-decay variant of the Adam optimiser [122], with a learning rate of 10^{-3} and a weight decay of 10^{-3} , and a batch size of thirty-two. The learning rate is governed by a plateau scheduler that halves

it whenever the validation loss fails to improve for a set number of epochs, down to a floor of 10^{-6} . All runs share the fixed data split of Section 6.2, so that any difference between runs is attributable to the change under test rather than to the data.

Progress is measured by two metrics, one for each task. For the binary task the metric is pixel accuracy, the fraction of pixels at which the thresholded prediction matches the target,

$$A = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\text{round}(\hat{y}_i) = y_i], \quad N = 64 \times 64 = 4096, \quad (6.3)$$

where $\hat{y}_i$ and y_i are the predicted and target values at pixel i , $\text{round}(\cdot)$ thresholds the prediction at one half, $\mathbb{I}[\cdot]$ is the indicator function, and N is the number of pixels. For the continuous lattice task a pixel accuracy is meaningless, since the target is a density and not a label; the metric there is the mean absolute error, expressed in density percentage points,

$$\text{MAE} = \frac{100}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad (6.4)$$

so that a value of, say, eleven means the prediction is on average eleven density percentage points from the true value. The loss minimised during lattice training is the mean squared error, which penalises large deviations more steeply, but the mean squared and mean absolute errors are not minimised at the same point. In the fine-tuning run of Section 6.6, for instance, the validation mean squared error reaches its minimum near epoch one hundred and twenty-three, while the validation mean absolute error continues to fall to its own minimum near epoch two hundred and five. Because the mean absolute error is the interpretable metric of record for the density task, the lattice models are selected by their best validation mean absolute error rather than by their best mean squared error, and the model reported is the one at that epoch.

Both metrics are tracked as aggregates over the full validation set and, more revealingly, on six fixed samples held constant across every run. These six span a range of topology types and let a qualitative behaviour that an aggregate average can conceal, a broken member on one sample, a filled void on another, be followed from run to run; the disagreement they expose between aggregate loss and per-sample quality is itself a finding of Section 6.6. All training was carried out on a single consumer graphics card, an NVIDIA RTX 3060 Ti with eight gigabytes of memory, on which a one-hundred-epoch run completes in about an hour, so that the entire study was conducted on commodity hardware.

6.5 Results

The architecture of Section 6.3 was reached by retaining, in turn, each refinement that improved the measured accuracy, and the cumulative effect of that sequence is given in Table 6.1. Each row is the best configuration of its stage, differing from the row above it by the single change named, and the trend is monotone: from a plain encoder-decoder at

Table 6.1: Progression of the retained refinements on the binary topology task. Each row applies one further change to the configuration above it; the accuracy is the mean pixel accuracy over the six fixed evaluation samples. The full run history is given in Appendix A.

Configuration (cumulative change)	Mean pixel accuracy
Encoder-decoder, no skip connections	94.1%
+ reduced batch size	96.0%
+ skip connections (U-Net)	96.7%
+ combined BCE and Dice loss	97.3%
+ batch normalisation	98.07%
+ attention gates	98.37%
+ extended training schedule	98.50%

94.1% mean pixel accuracy, the skip connections, the combined loss, batch normalisation and the attention gates each add a measurable increment, and a final extension of the training schedule to two hundred and fifty epochs brings the network to its best result of 98.50%. The full run-by-run history, including the tuning of the optimiser between these milestones, is collected in Appendix A.

The behaviour that produces this figure is shown in Figures 6.2 and 6.3. The first compares the predictions of the final network against the ground truth on the six evaluation samples. The overriding observation is how little there is to separate them: at the level of the global structural type, the brackets, arcs, chevrons and trusses are reproduced faithfully on every sample, a behaviour the network in fact reached from its very first version, which confirms that the mapping from the load and stress fields to the type of structure is learned readily. What improves across the architecture, and what remains imperfect even at the end, is the fine detail. Two failure modes persist on the hardest samples and are visible on close inspection of Figure 6.2: the thinnest members, those one or two elements wide, are occasionally broken or slightly displaced, and the smallest enclosed voids are sometimes filled in as solid. Both are consequences of the same cause. A feature of that scale must survive compression through the eight by eight bottleneck, and the bottleneck does not retain enough spatial information to place it exactly. Crucially, neither failure is addressed by the combined loss or by longer training; they are architectural, and the experiments of the next section confirm as much by trying, and failing, to remove them by other means.

The training history in Figure 6.3 shows the optimisation behind the result. The training and validation losses fall together over the first epochs and then separate, the validation loss settling onto a shallow plateau while the training loss continues to drift down, the familiar signature of mild overfitting that the plateau scheduler, visible halving the learning rate in the lower panel, holds in check. The best validation performance is reached well before the final epoch, and it is that best model, not the last, that is reported.

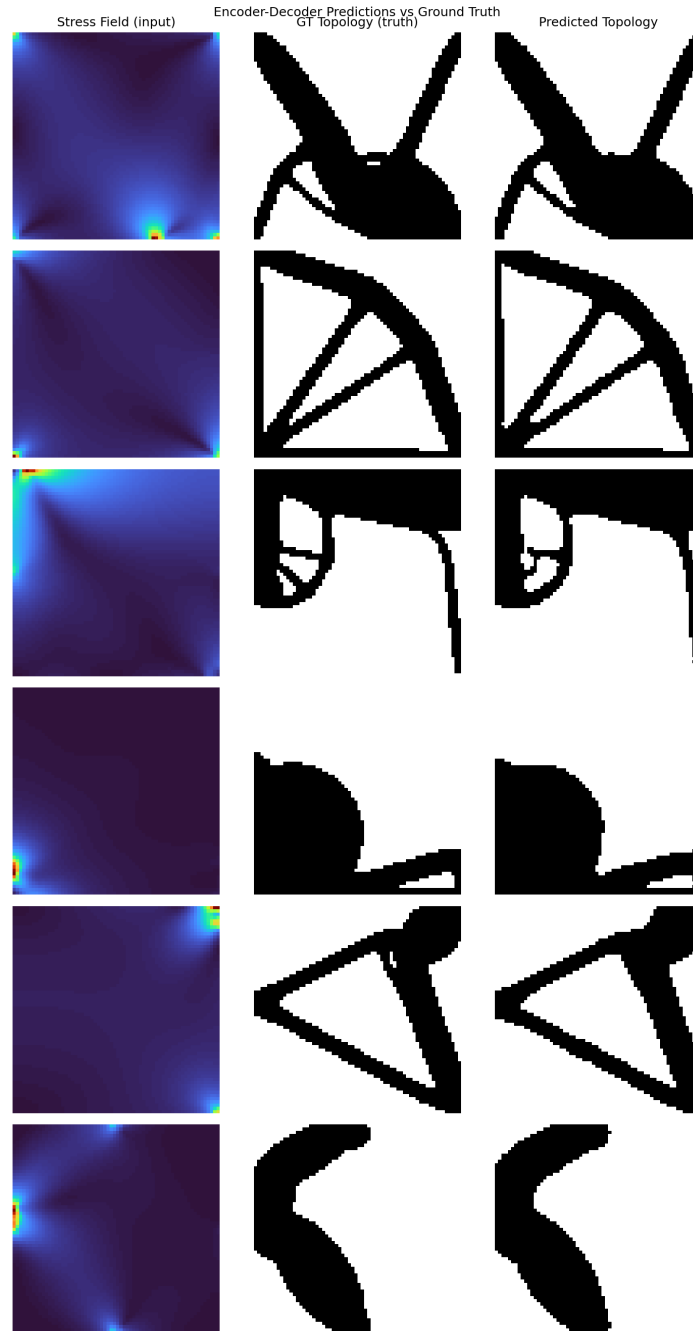


Figure 6.2: Binary topology predictions of the final network on the six fixed evaluation samples. Left: the von Mises stress field supplied as one of the five input channels. Centre: the SIMP ground-truth topology. Right: the network prediction. The global structural type is reproduced on every sample; the residual errors are confined to the thinnest members and the smallest voids, where the eight by eight bottleneck limits the recoverable spatial detail.

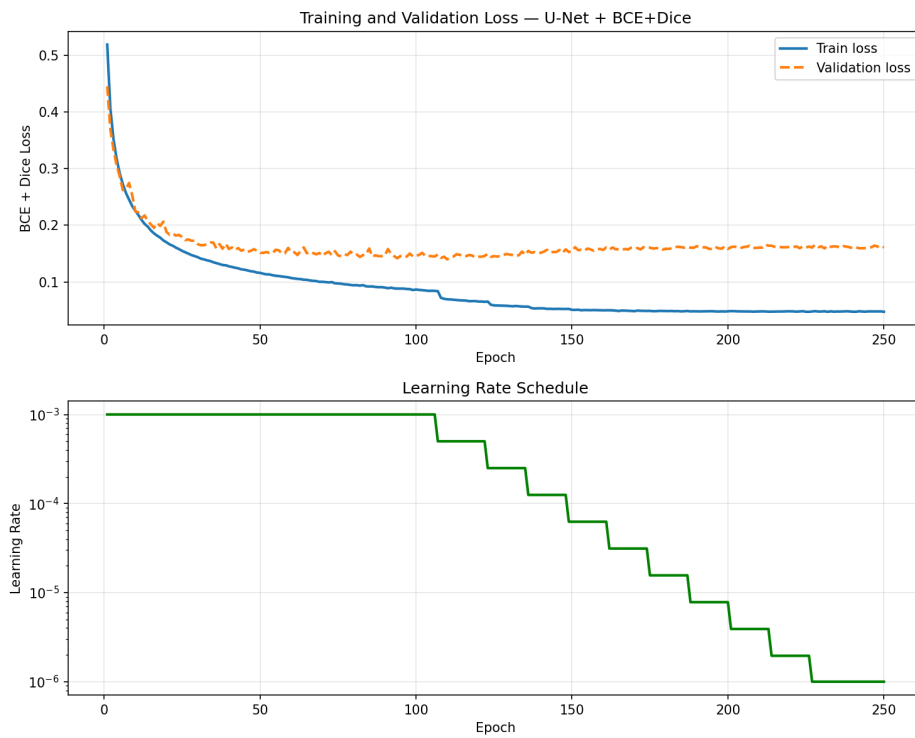


Figure 6.3: Training history of the final binary network. Top: the training and validation loss over the full schedule; after an initial common descent the validation loss settles onto a shallow plateau while the training loss continues to fall, indicating mild overfitting. Bottom: the learning-rate schedule, halved by the plateau scheduler each time the validation loss stalls. The reported model is the best-validation checkpoint, not the final epoch.

6.6 Extensions

With the baseline established, three further lines of investigation were pursued. Two of them, alternative loss functions and data augmentation, set out to push past the residual failures of Section 6.5 and did not; their chief value is the negative result they establish, that the binding limit is the architecture and not the loss or the data, and they are summarised briefly here with their full detail deferred to Appendix A. The third, the lattice fine-tuning study, is the transfer experiment for which the chapter was built, and it is treated in full.

6.6.1 Loss-function and augmentation studies

The residual failures of the baseline are at thin members and small voids, where the structure is almost entirely boundary. Two loss functions were tried in the hope of sharpening exactly there. The first weights the cross-entropy more heavily on the boundary pixels of the target, on the reasoning that those pixels carry the most shape information; the second, the same-class neighbour penalisation of Valverde et al. [123], modifies each pixel's logit by its worst-classified same-class neighbour, so that a pixel cannot be credited until its neighbours improve, a scheme aimed directly at broken members. Both degraded the result rather than improving it. The boundary weighting drew the network's effort onto boundary correctness at the expense of overall accuracy, and the neighbour penalisation, applied at full strength, produced gradients that interacted poorly with the overlap term and prevented confident classification. This outcome is the one anticipated above: where a member is so thin that the whole of it is boundary, adding loss there gives the network no new capacity to resolve it, because the missing capacity is spatial resolution at the bottleneck, which a loss function cannot supply.

Data augmentation was the second avenue, and here the constraint is the physics of the dataset rather than the architecture. As noted in Section 6.2, the applied load never points downward, so only a horizontal mirror keeps an augmented sample within the dataset's convention: a horizontal mirror maps the load angle to its supplement, which negates the horizontal load component and leaves the non-negative vertical component unchanged, whereas a vertical mirror or a rotation would produce a downward load that the network never sees at validation. Even restricted to the horizontal mirror, augmentation did not help. Three variants were tried, a stochastic per-sample flip, a statically doubled dataset, and a flip restricted to the boundary-condition configurations whose mirror image also exists in the data; all three left the validation loss slightly improved yet the pixel accuracy on the six evaluation samples slightly worse. That divergence is itself instructive. The aggregate validation loss, averaged over three thousand samples, can move in the opposite direction to the qualitative accuracy on specific cases, so that an apparent gain in the average masks a loss where it is most visible. A contributing cause was identified: only twelve of the forty-two boundary-condition configurations have a mirror image within the dataset, so an indiscriminate flip presents the network with configurations it never encoun-

Table 6.2: Summary of the extensions that did not improve on the baseline, by mean pixel accuracy over the six evaluation samples. None exceeds the baseline; the full per-sample results and loss formulations are in Appendix A.

Study	Variant	Mean pixel accuracy
Baseline	none	98.37%
Loss function	boundary-weighted (strong)	97.83%
	boundary-weighted (mild)	98.25%
	same-class neighbour penalisation	96.98%
Augmentation	horizontal flip, stochastic	97.75%
	horizontal flip, static doubled	98.22%
	horizontal flip, in-distribution	97.98%

ters at validation, and the restricted variant, which recovered part of the loss, confirms that this out-of-distribution effect is real, though not the whole of the explanation. The summary figures for both studies are given in Table 6.2, and the per-sample results and the formulations of the two losses in Appendix A.

6.6.2 Lattice fine-tuning and the transfer experiment

The closing study tests the hypothesis of Section 6.1: that a network trained on the binary task has learned structural knowledge it can carry to the lattice task. It does so by comparing two routes to a lattice density predictor, one that begins from the trained binary network and one that begins from nothing, under otherwise identical conditions, so that any difference is attributable to the initialisation alone.

6.6.2.1 The lattice dataset

The lattice task needs its own labels, which no public dataset provides, so a dataset of 2500 lattice optimisations was generated with the lattice engine of Section 4.5. To keep the comparison clean, the load and boundary-condition configurations were not generated independently but sampled, under the same fixed seed used for the train and validation split, from the same TopoDiff specification used for the binary task, so that the lattice inputs follow exactly the binary input distribution and the study tests adaptation of the output, not generalisation to new physics. Each configuration was written as an Abaqus model with a total applied load of three hundred newtons, optimised to convergence by the lattice engine, and the best-iteration density map retained. The generation was carried out on the FEUP supercomputer, on which a single optimisation takes about 8 minutes; run 24 at a time across its 64 cores, the full set of 2500 completed in roughly 14 hours. The samples were split 90:10 into 2250 for training and 250 for validation.

The target is the lattice density field, mapped to the same convention as the binary

target so that the network’s output need not be reinterpreted,

$$y_i = 1 - \frac{\rho_i}{100}, \quad (6.5)$$

where ρ_i is the relative density of element i in per cent and y_i the target, so that a fully dense element maps to zero, exactly as solid mapped to zero in the binary task, and a void element to one. One input channel is treated differently, and the decision is deliberate. The lattice engine takes no prescribed volume fraction, so the volume-fraction input channel, which for the binary task carried the SIMP material budget, has no counterpart in the lattice problem. Rather than feed the network a value that no longer means anything, the channel is zeroed, on the reasoning that retaining it would supply the pre-trained encoder, which had learned to respect the volume-fraction input strongly, with a now-conflicting cue that would mislead more than it informs. The consequence of removing it is taken up below, since it bears directly on how hard the lattice task is.

6.6.2.2 Fine-tuned and from-scratch training routes

The fine-tuned route, denoted here the fine-tuned network, starts from the trained binary network of Section 6.5. Its encoder and bottleneck, about 70% of the parameters, are frozen, and only the attention gates, decoder and output convolution are left free to adapt. The reasoning is threefold: the encoder has learned to extract structural features from thirty thousand examples and that knowledge applies directly to the lattice task; with only 2 250 training samples, unfreezing the encoder would risk overwriting that knowledge with noise; and it is the decoder that must change most, since it must now produce a graded density rather than a near-binary mask. The learning rate is reduced tenfold, to 10^{-4} , to preserve the pre-trained weights, and the loss is the mean squared error of Section 6.4. The from-scratch route is identical in every hyperparameter, the same learning rate, schedule, loss and data split, but begins from a random initialisation with all layers free, so that the only variable between the two is whether the network starts from the binary representation or from nothing.

6.6.2.3 Convergence and generalisation results

The two routes are compared in Table 6.3, their training histories in Figure 6.4, and their predictions, alongside the two optimisers’ outputs, in Figure 6.5. The clearest signal is at the very start of training. The fine-tuned network begins at a mean density error of 19.3% and falls to 14.4% within ten epochs, before training has had time to rewrite any weights appreciably; the from-scratch network begins at 39.8%, more than twice as far out, and needs roughly 45 epochs to reach the level the fine-tuned network had at 10, a convergence about four and a half times slower. That gap at the first epoch is a direct measurement of what the pre-training is worth: the binary representation was immediately useful, which is to say that genuine transfer occurred, and the structural knowledge of where load paths

Table 6.3: The two routes to a lattice density predictor, under identical hyperparameters. The fine-tuned network starts from the trained binary network with a frozen encoder; the from-scratch network starts from a random initialisation. Mean density error is in density percentage points; the six-sample figure is the mean over the fixed evaluation set.

	Fine-tuned	From scratch
Best validation mean absolute error	11.15%	11.01%
Mean absolute error at epoch 1	19.3%	39.8%
Mean absolute error at epoch 10	14.4%	23.5%
Epochs to reach 15% error	~ 7	~ 45
Mean error on six evaluation samples	7.1%	7.8%
Final training mean squared error	0.033	0.007
Train-to-validation error ratio	$\sim 2:1$	$\sim 10:1$

run carried across to the new task.

What the transfer does not do is raise the accuracy ceiling. Both routes plateau at essentially the same place, a validation mean absolute error of about 11% and a mean error of about 7% on the six evaluation samples, because that ceiling is set by the size of the lattice dataset and not by the initialisation. Within that shared ceiling the fine-tuned network is, on balance, the better model. It generalises better, with a lower validation mean squared error, and it overfits far less: its training and validation errors stay within a factor of about two of each other, where the from-scratch network drives its training error down almost to zero while its validation error does not follow, a roughly tenfold gap that marks near-complete memorisation of the small training set. The frozen encoder, in other words, acts as a structural regulariser, constraining the network to interpretations consistent with its pre-trained understanding rather than letting it memorise. The from-scratch network edges ahead on only one figure, the aggregate validation mean absolute error, and by a tenth of a point, which is within the noise at this dataset size; on the six evaluation samples and on every measure of generalisation the fine-tuned network is ahead. The result is therefore not that one route is more accurate than the other, but that pre-training yields convergence speed and generalisation at a shared accuracy ceiling.

6.6.2.4 Origins of the accuracy ceiling

That the fine-tuning accelerates convergence but does not lift the accuracy ceiling has three causes acting together, and they are worth separating because they answer different questions. The first is the size of the lattice dataset. A set of 2 250 samples is too few to pin down the density mapping precisely, and the shared plateau of both routes, together with the near-total memorisation of the from-scratch route, is the clear signature of a data-limited regime; this sets the ceiling that both routes reach. The second is a genuine shift between the two optimisers' outputs. The SIMP labels of the binary dataset come from a volume-fraction-constrained compliance minimisation, while the lattice labels come

from a volume-fraction-free, utilisation-driven design, and for the same load case the two produce structurally different optima, as the lattice-engine column of Figure 6.5 makes plain. The magnitude of that difference is principally governed by the gap between the volume fraction the SIMP label was given and the fraction the lattice engine settled on by itself: where the two are close the topologies nearly coincide, and where they are far apart one optimiser fills a compact solid region while the other lays down a single slender load path. A secondary contribution comes from the differing objectives even at a matched fraction, since minimum compliance and full stress utilisation are not the same criterion. Pre-training can transfer only the part of the mapping the two optimisers share, the location of the principal load paths, which is why convergence is fast; it cannot transfer the part on which they genuinely disagree. The third cause is the zeroed volume-fraction input. Removing that channel, the right decision given that the lattice engine has no such input, leaves the network to infer the emergent converged fraction from the load and stress fields alone, a more underdetermined regression than the binary task, in which the fraction was supplied; and the frozen encoder, calibrated to a fraction input it no longer receives, is the most likely origin of the residual void haze seen in the fine-tuned predictions of Figure 6.5. The speed result and the bounded ceiling are thus two sides of the same fact: the shared structure transfers, and the divergent structure, capped by the scarce data, does not.

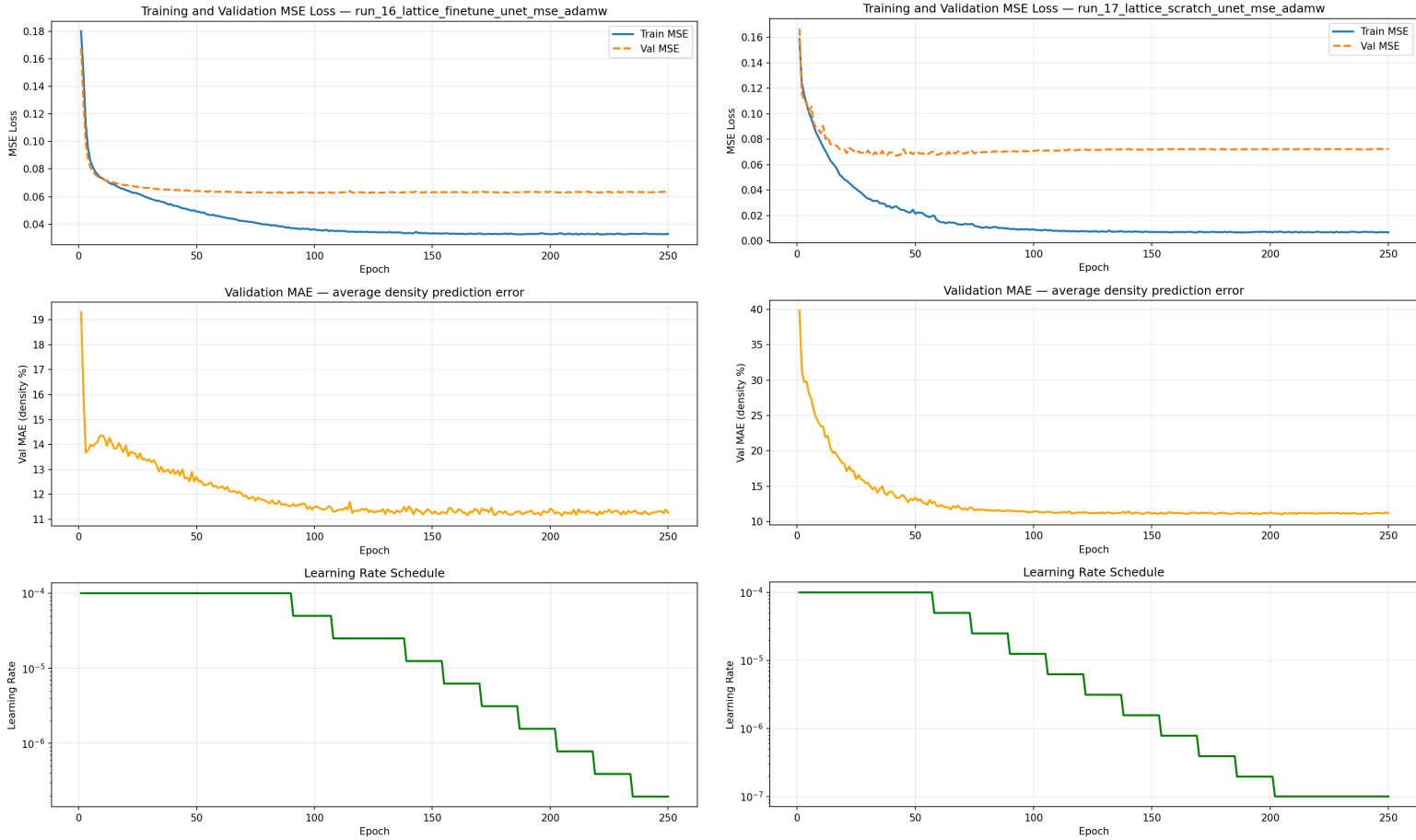
6.7 Discussion and Limitations

Taken together, the experiments of this chapter make a consistent case. A compact convolutional network, just under two million parameters and trained on commodity hardware, reproduces the solid-void optima of a verified dataset to about 98.5% pixel accuracy and returns a prediction in the time of a single forward pass, which is the order-of-minutes-to-milliseconds acceleration that motivated the study. The accuracy was won by architecture, the skip connections, batch normalisation and attention gates, far more than by loss design or data augmentation, both of which were tried and neither of which improved on the baseline; the residual errors are localised at the thinnest members and smallest voids and are a property of the bottleneck, not of the training. And the transfer experiment confirms the chapter’s central hypothesis in a qualified form: a network pre-trained on the binary task adapts to the lattice task about four and a half times faster and generalises markedly better than one trained from scratch, evidence that genuine structural knowledge transfers, even though the final accuracy is set not by the initialisation but by the scarcity of the lattice data and by the structural mismatch between the fraction-constrained solid-void labels and the fraction-free lattice engine, whose divergent optima the pre-training cannot reconcile.

The limitations are equally clear and are stated plainly. The eight by eight bottleneck is the binding constraint on binary accuracy; the thin members and small voids it cannot resolve would need a deeper or higher-resolution architecture, not a different loss, to recover.

The lattice dataset, at 2 250 samples, is an order of magnitude smaller than the binary one, and it is this scarcity, as much as the distribution shift between the two optimisers, that caps the lattice result; a substantially larger lattice dataset is the single change most likely to decide whether the pre-training advantage extends from convergence speed into final accuracy. The mean squared loss used for the lattice task penalises error uniformly and so yields blurred density boundaries, and in any case the lattice optimiser’s own output is close to binary in practice, with few elements at intermediate density, so the task is nearer to classification than to a true continuous regression. The whole pipeline is two-dimensional, and the extension to three dimensions carries the cubic growth in cost and the different structural behaviour noted in Chapter 2. Above all, a data-driven prediction carries no guarantee of physical validity, no assurance of equilibrium or connectivity, which is why the surrogate is positioned throughout as a complement to the verified optimiser and not a replacement for it, and why independent finite-element validation remains indispensable for any design it produces.

Several of these limitations point to natural continuations. A larger lattice dataset, a deeper bottleneck or dilated convolutions to widen the receptive field, and a topology-preserving loss such as the centreline-Dice of Shit et al. [124] to address the connectivity of thin members, are the obvious next steps for the prediction quality itself. For the transfer study specifically, the zeroing of the volume-fraction channel was a deliberate choice to avoid a conflicting cue, but a fine-tuning run that instead retains the channel and lets the encoder relearn its meaning under the lattice convention would test whether the input can be repurposed rather than discarded, and is a clean experiment to set against the present one. These directions are taken up, with the conclusions of the other two phases of the work, in the final chapter.



(a) Fine-tuned (frozen encoder).

(b) From scratch (random initialisation).

Figure 6.4: Training histories of the two lattice density predictors: the mean squared loss (top), the validation mean absolute error (middle), and the learning-rate schedule (bottom). The fine-tuned network **a** starts near 19% error and falls quickly, its training and validation losses staying close; the from-scratch network **b** starts near 40%, takes far longer to descend, and drives its training loss far below its validation loss, the signature of overfitting on the small dataset.

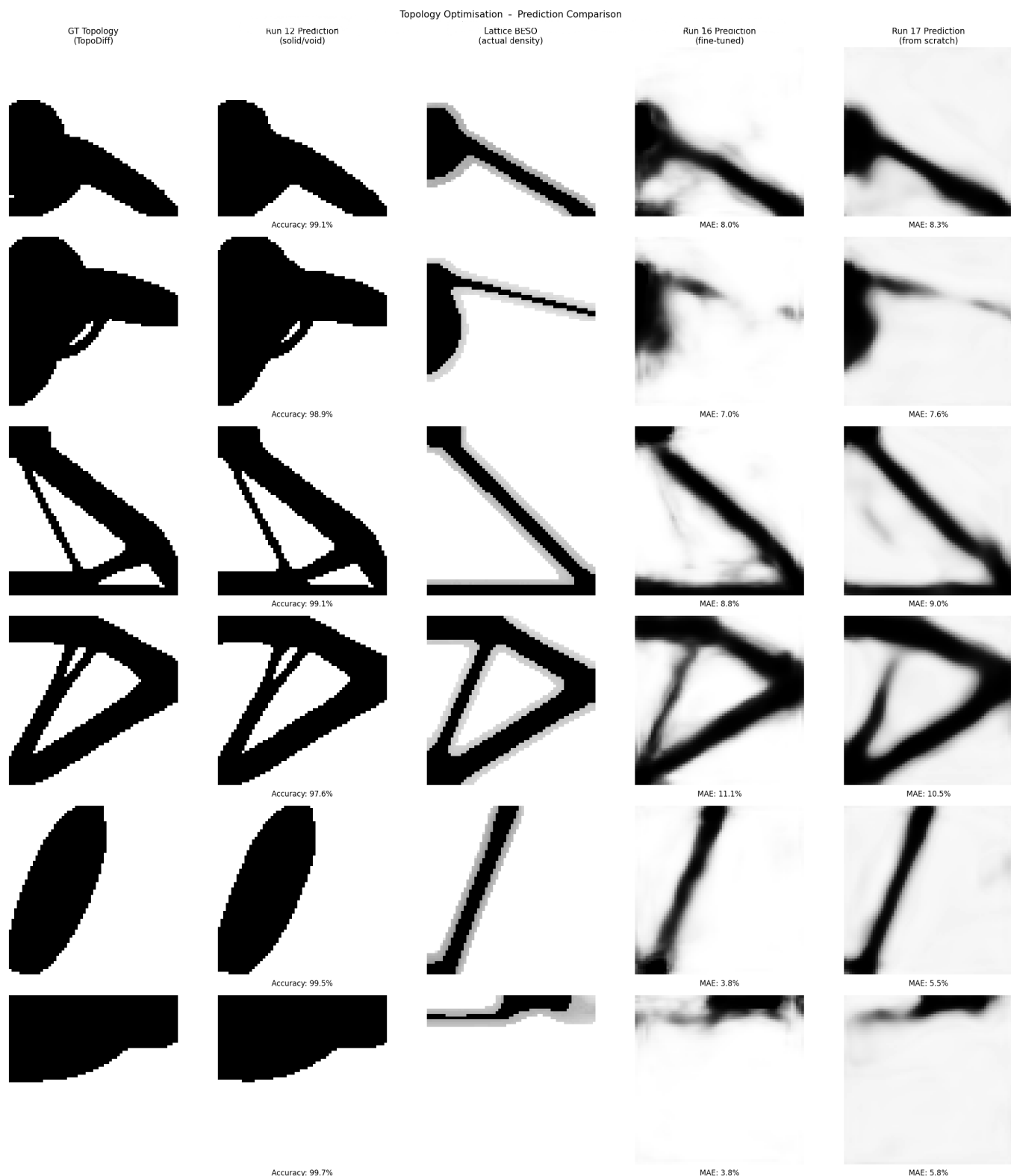


Figure 6.5: Prediction comparison on six samples. Columns, left to right: the SIMP ground-truth topology; the binary prediction; the lattice-engine target density; the fine-tuned lattice prediction; and the from-scratch lattice prediction. The binary prediction reproduces its ground truth closely, and both lattice predictions recover the load paths of theirs. The differences between the two optimisers' outputs, and the residual haze in the void regions of the fine-tuned prediction, are discussed in Section 6.6.2.4.

Chapter 7

Conclusions and Future Work

This dissertation set out to prove the union of topology optimisation and selective laser melting on a genuine engineering problem, and it pursued that aim across three phases: the construction, from first principles, of a bespoke optimisation framework; the application of that framework to a real, performance-critical component of the Adamastor Furia; and an exploratory study of whether a neural network can learn to reproduce an optimiser’s output. Each phase has been carried through to a result. This chapter draws those results together, states the limitations that qualify them, and sets out the work they leave open.

7.1 Conclusions

The first phase delivered the optimisation framework that is the methodological core of the work. It was built and developed from the ground up on a convergent, mesh-independent, soft-kill bidirectional evolutionary scheme, and it carries two distinct engines upon a single shared algorithmic core: a solid-void engine for classical topology optimisation and a lattice engine that distributes a graded relative density for mesoscale lightweighting. Because the framework is oriented throughout towards selective laser melting, it also provides an experimental thermal-proxy feature that biases the solid-void engine away from geometry left unsupported during the build, at a fraction of the cost of a coupled transient simulation. Verified against benchmark problems of known solution in Chapter 4, the framework was shown to be correct before it was ever applied to a real part, and it is this tool, and not any single result it produced, that constitutes the principal outcome of the dissertation.

The second phase applied that framework to the Furia’s roll-damper bracket, and it is against this concrete problem that the whole of the work is ultimately measured. The bracket, originally a 1.520 kg part machined from FR-3 steel, was redesigned as a 0.459 kg body intended for manufacture in AlSi10Mg by selective laser melting, a reduction of very nearly 70% in mass. The redesigned part meets its static requirement across its volume in all three load cases, holding below the 167 MPa admissible everywhere but for a single localised region at the pin bore in the governing right-roll case, which was shown in Section 5.6.2 to be an artefact of the load-introduction model rather than a true

overload of the material. Under a representative Spa-Francorchamps duty cycle the part is far from fatigue-limited, its Goodman life reaching some 1.42×10^7 laps, equivalent to roughly 9.75×10^7 km, at a safety factor of 1.75 on the cycle amplitudes (Section 5.6.3). The bracket is therefore governed by static strength, and specifically by the stress where its loads enter, which is what fixed the mass target where it was fixed.

Most of the mass was removed not by the optimiser directly, whose own volume saving was a modest 11%, but by the change of material it enabled. The eccentric, bending-dominated load path that had originally dictated steel was carried into a light alloy by placing it precisely along the line the bending demands, which is exactly the redistribution the method exists to perform; the optimisation and the material substitution are thus not two savings but one, since the substitution is only admissible because the optimisation makes the aluminium part strong enough. Two practical outcomes for Adamastor follow from the study. Because the geometry was fixed against the larger rear-axle roll range while being loaded with the higher front-axle damper forces, a single bracket serves both ends of the car, halving the number of distinct parts to qualify and build; and the optimised bracket is suitable to build in AlSi10Mg with T6 heat treatment, with the pin bore machined or reamed to its final fit after building, which is standard practice for a bore that accepts a pin and is the basis on which its fatigue strength was evaluated.

The third phase turned from the result of the methodology to its cost, and asked whether a neural network could stand in for the iterative optimiser. A compact convolutional encoder-decoder of the U-Net family, just under two million parameters and trained on commodity hardware, was shown in Section 6.5 to reproduce the solid-void optima of a verified public dataset to about 98.5% pixel accuracy, returning a prediction in the time of a single forward pass. The sharper question, of whether the representation such a network learns is transferable, was answered in qualified form by the fine-tuning study of Section 6.6: a network pre-trained on the binary task adapts to the harder, graded lattice task about four and a half times faster than one trained from nothing, which is genuine evidence that structural knowledge, the location of the principal load paths, transfers between the two optimisers. The transfer accelerates convergence but does not lift the final accuracy, which is set instead by the scarcity of the lattice data and by the mismatch between the volume fraction-constrained solid-void labels and the volume fraction-free lattice engine, whose divergent optima the pre-training cannot reconcile. The surrogate is, for that reason, a complement to the verified optimiser and not a replacement for it.

A single principle ran through all three phases and holds at their close. No design produced by any route was accepted on the strength of its method alone: the optimiser's output, the aluminium redesign, and any topology the network predicts were each held, or must each be held, to independent finite-element validation, and the manufacturing constraints of selective laser melting were treated throughout as real limits rather than idealised assumptions.

7.2 Limitations

The conclusions carry the caveats their idealisations imply, and these are stated plainly. On the application side, the optimisation model used, for reasons of cost, a coarse rocker whose displacement does not converge; this is admissible only because the evolutionary scheme is comparative rather than absolute, as the rank study of Section 5.3 established quantitatively, so that the topology is trustworthy while the absolute displacements are not. The localised exceedance at the pin bore was argued, not demonstrated, to be a load-introduction artefact, and a model that represents the bush and pin as a contact pair rather than as a coupling would replace that argument with a direct demonstration. The fatigue verdict rests, as it states, on the bore being finished by machining or reaming to its final fit. The lattice comparisons rest on the fast Experimental property estimate of Section 4.7.4 rather than on a homogenisation from first principles. And the whole of the validation is numerical: no part has yet been built or tested.

On the machine-learning side the bounds are equally clear, and are set out in full in Section 6.7. The eight by eight bottleneck of the network is the binding constraint on binary accuracy, and the thin members and small voids it cannot resolve would need a deeper or higher-resolution architecture, not a different loss, to recover. The lattice dataset, 2500 samples of which 2250 form the training split, is an order of magnitude smaller than the binary set, and it is this scarcity, as much as the distribution shift between the two optimisers, that caps the lattice result. The mean squared loss used for the lattice task penalises error uniformly and so blurs the density boundaries. The whole pipeline is two-dimensional. And above all, a data-driven prediction inherently carries no guarantee of physical validity, no assurance of equilibrium or connectivity, which is precisely why the surrogate is positioned as a complement to the verified optimiser and why independent finite-element validation remains indispensable for any design it produces.

7.3 Future Work

Several of these limitations point directly to their own continuations. For the bracket, the most immediate is to replace the coupling at the pin bore with a bush-and-pin contact pair, which would convert the artefact argument of Section 5.6.2 into a direct demonstration and confirm the static margin at the one location where it is in doubt; alongside it, the thickened boss and its reduced clearance should be verified against the as-built dimensions and the rocker tolerances before the part is committed. Where a further mass saving is sought, the framework's lattice engine offers a route not taken here: a graded lattice core within a solid skin, of the kind verified in Section 4.7.4, would trade a little of the discrete engine's efficiency for the manufacturability and multifunctionality of a cellular interior, and is a natural next step for a part whose static margins, outside the load-introduction regions, are comfortable. On the framework itself, the lattice property model would benefit from a first-principles homogenisation in place of the present fast estimate, and the constant-volume

refine phase, shown in Section 4.4.5 to improve specific stiffness even with the mitigation active, invites a variant run with the mitigation switched off, which would likely sharpen the build-phase result too. Above all of these stands the physical step the numerical study has earned: to build the optimised bracket in AlSi10Mg and to test it, so that the predicted static and fatigue behaviour is confirmed on real material.

The surrogate leaves a clear programme of its own. The single change most likely to matter is a substantially larger lattice dataset, since it is data scarcity, more than any choice of architecture, that now caps the lattice result and that will decide whether the pre-training advantage extends from convergence speed into final accuracy. Beyond it, a deeper or higher-resolution bottleneck, or dilated convolutions to widen the receptive field, would address the resolution limit that a different loss cannot; a topology-preserving loss such as the centreline-Dice of Shit et al. [124] would target the connectivity of the thinnest members directly; and a fine-tuning run that retains the volume-fraction channel, letting the encoder relearn its meaning under the lattice convention rather than discarding it, is a clean experiment to set against the present one. The extension of the whole pipeline to three dimensions, with the cubic growth in cost it entails, remains the largest and most consequential step, and the one that would bring the surrogate into the same dimensional setting as the framework it is meant to accelerate.

Each of these continuations sharpens rather than revises the central finding of the work. A bidirectional evolutionary framework, built from first principles and verified on problems of known solution, has taken an over-built steel bracket and returned a part very nearly 70% lighter in a printable aluminium alloy, shown to meet its static requirement and to be far from fatigue-limited; and a compact network has been shown to learn, and in part to transfer, the structural knowledge that optimiser embodies. What remains is to carry both from a validated numerical demonstration to built and tested hardware, which is the natural next stage of the work begun here.

Appendix A

Supplementary Results for the Machine Learning Surrogate

This appendix collects the detail deferred from Chapter 6: the full run history of the binary topology study, and the formulations and per-sample results of the loss-function and augmentation experiments summarised in Section 6.6.

A.1 Full binary run history

Table A.1 lists the runs of the binary topology study in sequence. The runs form a controlled progression in which each configuration differs from a previous one by a single change, which is what licenses the attribution of an accuracy increment to that change. The architecture milestones (skip connections, batch normalisation, attention gates) and the extended schedule are the refinements retained in the final network; the optimiser variants and the loss and augmentation experiments are the avenues that were tried and not retained. Accuracy is the mean pixel accuracy over the six fixed evaluation samples of Section 6.5.

Table A.1: Full run history of the binary topology study. ED denotes the plain encoder-decoder; U-Net adds skip connections; BN and AG denote batch normalisation and attention gates. Loss is either binary cross-entropy (BCE), the combined BCE and Dice loss of Equation (6.2), a boundary-weighted variant, or same-class neighbour penalisation (SCNP). Accuracy is the mean over the six evaluation samples.

Run	Architecture	Epochs	Loss	Augmentation	Note	Accuracy
01	ED	50	BCE	none	baseline	94.1%
02	ED	50	BCE	none	reduced batch size	96.0%
03	U-Net	50	BCE	none	skip connections	96.7%
04	U-Net	100	BCE	none	longer schedule	97.0%
05	U-Net	50	BCE+Dice	none	combined loss	96.75%
06	U-Net	100	BCE+Dice	none	combined loss, longer	97.3%
07	U-Net	100	BCE+Dice	none	stronger weight decay	97.2%
07b	U-Net	100	BCE+Dice	none	milder weight decay	97.75%
10	U-Net+BN	100	BCE+Dice	none	batch normalisation	98.07%
11	U-Net+BN+AG	100	BCE+Dice	none	attention gates	98.37%
12	U-Net+BN+AG	250	BCE+Dice	none	extended schedule	98.50%
13	U-Net+BN+AG	100	boundary	none	strong boundary weight	97.83%
13b	U-Net+BN+AG	100	boundary	none	mild boundary weight	98.25%
14	U-Net+BN+AG	100	SCNP	none	neighbour penalisation	96.98%
15a	U-Net+BN+AG	100	BCE+Dice	h-flip, stochastic	all configurations	97.75%
15b	U-Net+BN+AG	100	BCE+Dice	h-flip, static doubled	all configurations	98.22%
15c	U-Net+BN+AG	100	BCE+Dice	h-flip, in-distribution	mirrorable configurations	97.98%

A.2 Loss-function study

Two modifications to the combined loss of Equation (6.2) were tested, both aimed at the thin-member and small-void failures of the baseline.

Boundary-weighted loss. The first amplifies the cross-entropy on the boundary pixels of the target, on the reasoning that those pixels carry the most information about member geometry. A boundary map $b_i \in \{0, 1\}$ is computed per batch from the target, as the difference between a three by three dilation and erosion of the target mask, and the cross-entropy is reweighted,

$$\mathcal{L}_{\text{BCE}}^{\text{bw}} = \frac{1}{N} \sum_{i=1}^N (1 + \alpha b_i) \text{BCE}(\hat{y}_i, y_i), \quad (\text{A.1})$$

where α sets the boundary emphasis, so that an edge pixel contributes $(1 + \alpha)$ times the loss of an interior pixel; the Dice term is left unchanged. The strong variant used $\alpha = 4$ and the mild variant $\alpha = 2$. Both degraded the result, the stronger weighting more so, by drawing the network’s effort onto boundary correctness at the expense of overall accuracy.

Same-class neighbour penalisation. The second is the same-class neighbour penalisation of Valverde et al. [123], a modification applied to the logits before the cross-entropy. For each pixel it propagates the poorest-classified same-class neighbour within a three by

three window, so that a pixel’s logit is penalised by its worst same-class neighbour and the network cannot be credited for a pixel until its neighbours improve, a mechanism aimed directly at broken members and isolated false positives. Applied at full strength the penalisation produced gradients that interacted poorly with the overlap term and prevented confident pixel-level classification, and it gave the largest degradation of the three loss experiments.

The per-sample results of the loss study are given in Table A.2. The degradation is consistent across samples, and the conclusion is that for this architecture, dataset and training regime the combined cross-entropy and Dice loss is the best choice, because the residual failures are architectural and a loss function cannot supply the spatial resolution the bottleneck lacks.

Table A.2: Per-sample pixel accuracy of the loss-function study, against the baseline (Run 12). No variant exceeds the baseline on the mean.

Sample	Baseline	Boundary (strong)	Boundary (mild)	SCNP
0	98.6%	97.3%	97.8%	97.5%
1	97.9%	97.7%	97.6%	95.6%
2	97.7%	96.6%	96.9%	95.5%
3	99.0%	98.7%	98.9%	97.5%
4	98.6%	98.3%	98.9%	97.8%
5	99.2%	98.4%	99.4%	98.0%
Mean	98.50%	97.83%	98.25%	96.98%

A.3 Augmentation study

Admissible augmentation. The load convention of Section 6.2 restricts which geometric augmentations keep a sample within the data distribution. A horizontal mirror maps the load angle γ to its supplement $\pi - \gamma$. Because $\cos(\pi - \gamma) = -\cos \gamma$ and $\sin(\pi - \gamma) = \sin \gamma$, the horizontal load component is negated while the vertical component is unchanged, so the mirrored sample retains the non-negative vertical load that every sample in the dataset has; the stress and energy fields and the target topology are mirrored identically, so the augmented sample is a self-consistent, correctly labelled problem. A vertical mirror or any rotation, by contrast, sends $\sin \gamma$ to $-\sin \gamma$ and so produces a downward load that the dataset never contains. Only the horizontal mirror is therefore admissible, and the other augmentations were excluded on this basis.

Variants and the out-of-distribution effect. Three horizontal-mirror variants were tested: a stochastic per-sample flip applied with even probability each epoch; a static variant in which the training set was doubled by adding every flipped sample as a fixed extra; and an in-distribution variant in which the flip was applied only to samples whose

boundary-condition configuration has a mirror image elsewhere in the dataset. The last variant is motivated by a property of the data: of the forty-two boundary-condition configurations, only twelve have a mirror image within the dataset, so an indiscriminate flip presents the network, for thirty of the forty-two configurations, with a mirrored configuration it never encounters at validation. The in-distribution variant, restricted to the twelve mirrorable configurations, recovered part of the degradation seen in the stochastic variant, which confirms that this out-of-distribution effect is a real contributor; the residual degradation is attributed to the mild over-representation of those twelve configurations in the augmented set, which biases the network away from the full forty-two-configuration distribution.

The per-sample results are given in Table A.3. Across all three variants the aggregate validation loss improved while the pixel accuracy on the six evaluation samples worsened, the disagreement between aggregate and per-sample measures noted in Section 6.6. The conclusion is that horizontal-mirror augmentation, in any of these forms, does not benefit this network on this dataset at this scale, and that the binding limit on accuracy is the architectural bottleneck rather than the quantity of training data.

Table A.3: Per-sample pixel accuracy of the augmentation study, against the unaugmented baseline (Run 11). No variant exceeds the baseline on the mean.

Sample	Baseline	Stochastic	Static doubled	In-distribution
0	98.5%	97.9%	98.0%	97.9%
1	97.2%	96.3%	97.4%	96.4%
2	97.2%	96.6%	96.9%	97.2%
3	99.2%	98.6%	98.9%	99.1%
4	98.8%	98.2%	98.9%	98.2%
5	99.4%	98.9%	99.2%	99.1%
Mean	98.37%	97.75%	98.22%	97.98%

Appendix B

Source Code of the Optimisation Pipeline

The optimiser and its post-processing pipeline are implemented in six Python scripts, described below in the order in which a run passes through them. The complete, commented source is made available in a public repository rather than reproduced here, as the full listings would run to several hundred pages. The numerical methods behind the scripts are set out in Chapter 4 and their application to the suspension component in Chapter 5.

B.1 Overview of the scripts

The optimiser driver, `beso_main.py`, is the core of the framework. It runs inside the Abaqus Python environment and carries the BESO iteration loop: at each iteration it calls the solver on the current design, reads back the field results, evaluates the specific stiffness, filters the element sensitivities, and updates the solid set towards the target volume fraction. It checkpoints its state, so that a run can be resumed after an interruption or continued to a lower volume fraction without repeating the work already done.

The geometry dumper, `beso_geometry_dump.py`, is a helper called by the driver. Whenever a new best iteration is found it reads the node coordinates, the element connectivity and the list of solid elements from the current model and writes them to comma-separated files, giving the exporter a self-contained description of the geometry to be rebuilt.

The solid-void exporter, `beso_stl_generator.py`, reads the geometry written by the dumper and produces a watertight STL surface of the optimised part, suitable for inspection and for the additive manufacturing route.

The lattice driver, `beso_lattice_main.py`, is the lattice counterpart of `beso_main.py`. It follows the same iteration loop but acts on a graded density field rather than a solid-void one, using the homogenised Gibson–Ashby properties of the chosen triply periodic minimal surface morphology to relate the local density to stiffness and strength.

The lattice exporter, `beso_lattice_stl_generator.py`, converts the optimised density field into a graded lattice, applying the resolution law and the surface smoothing described in Chapter 4 before writing the STL.

Finally, the launcher, `beso_launcher.py`, ties the pipeline together. It runs a small local web server and serves a browser interface from which a run is configured, started and monitored, and in which the specific-stiffness history of several runs can be compared. The interface writes the configuration file that the driver and the exporters read, starts each of them as a subprocess, and streams their progress back to the browser so that a run can be followed live without returning to the terminal. Its purpose is to make the optimiser and the whole STL-generation pipeline usable by an operator who does not wish to edit code: the complete sequence, from model preparation to printable geometry, is driven from the interface.

Disclosure on the launcher. The browser interface of the launcher, that is, its HTML, CSS and client-side JavaScript, was developed with the assistance of large language models. The optimiser, the exporters and the numerical pipeline they drive are the author’s own work; the launcher places a convenience front-end over that pipeline and introduces no new method. It communicates with the scripts through the shared configuration file and their output logs, launches them as subprocesses, and reports the state of each run back to the interface. It is included in the repository because it is the practical means by which the pipeline is operated.

B.2 Availability

The complete source code, with comments, is available in a public repository at <https://github.com/migueldiaslima/beso-topology-optimisation>, under the file names listed in Table B.1. The repository contains the optimisation and post-processing code only; the component-specific data belonging to Adamastor are not distributed with it.

Table B.1: The six source files of the optimisation pipeline and their roles. All are provided in the `src/` directory of the repository.

File	Role
<code>beso_main.py</code>	Solid-void optimiser driver (BESO loop, Abaqus)
<code>beso_geometry_dump.py</code>	Best-iteration geometry export to CSV
<code>beso_stl_generator.py</code>	Solid-void STL exporter
<code>beso_lattice_main.py</code>	Lattice optimiser driver (graded density)
<code>beso_lattice_stl_generator.py</code>	Lattice (TPMS) STL exporter
<code>beso_launcher.py</code>	Web-based launcher and run monitor

Bibliography

- [1] Mohammad Mahdi Behzadi and Horea T. Ilies. GANTL: Toward practical and real-time topology optimization with conditional generative adversarial networks and transfer learning. *Journal of Mechanical Design*, 144(2):021711, 2022. doi: 10.1115/1.4052757.
- [2] François Mazé and Faez Ahmed. Diffusion models beat GANs on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9108–9116, 2023. doi: 10.1609/aaai.v37i8.26093.
- [3] Peter Mercelis and Jean-Pierre Kruth. Residual stresses in selective laser sintering and selective laser melting. *Rapid Prototyping Journal*, 12(5):254–265, 2006.
- [4] Fabio Distefano, Salvatore Pasta, and Gabriella Epasto. Titanium lattice structures produced via additive manufacturing for a bone scaffold: A review. *Journal of Functional Biomaterials*, 14(3):125, 2023. doi: 10.3390/jfb14030125.
- [5] GE Aerospace. Transformation in 3D: how a walnut-sized part changed the way GE aviation builds jet engines. <https://www.ge.com/news/reports/transformation-3d-walnut-sized-part-changed-way-ge-aviation-builds-jet-engines>, 2018. Accessed 2 July 2026.
- [6] TIME. This company is using 3D printing and AI to make sports cars. <https://time.com/7023380/czinger-3d-print-ai-sports-car/>, 2024. Accessed 2 July 2026.
- [7] Adamastor. Furia: Portugal’s first hypercar. <https://www.adamastor.com.pt/>, 2026. Accessed 4 July 2026.
- [8] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.
- [9] Martin Philip Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989.
- [10] Ming Zhou and George I. N. Rozvany. The COC algorithm, part II: Topological,

- geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1–3):309–336, 1991.
- [11] George I. N. Rozvany, Ming Zhou, and Torben Birker. Generalized shape optimization without homogenization. *Structural Optimization*, 4(3–4):250–254, 1992.
 - [12] Martin Philip Bendsøe and Ole Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9–10):635–654, 1999.
 - [13] Ole Sigmund. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.
 - [14] Ole Sigmund and Joakim Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1):68–75, 1998.
 - [15] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1–2):227–246, 2003.
 - [16] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, 2004.
 - [17] Yi Min Xie and Grant P. Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49(5):885–896, 1993.
 - [18] Yi Min Xie and Grant P. Steven. *Evolutionary Structural Optimization*. Springer, London, 1997.
 - [19] Ming Zhou and George I. N. Rozvany. On the validity of ESO type methods in topology optimization. *Structural and Multidisciplinary Optimization*, 21(1):80–83, 2001.
 - [20] Osvaldo M. Querin, Grant P. Steven, and Yi Min Xie. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations*, 15(8):1031–1048, 1998.
 - [21] Xiaodong Huang and Yi Min Xie. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43(14):1039–1049, 2007.
 - [22] Xiaodong Huang and Yi Min Xie. *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. John Wiley & Sons, Chichester, 2010.
 - [23] David Brackett, Ian Ashcroft, and Richard Hague. Topology optimization for additive

- manufacturing. In *Proceedings of the Solid Freeform Fabrication Symposium*, pages 348–362, Austin, Texas, 2011.
- [24] Andrew T. Gaynor and James K. Guest. Topology optimization considering overhang constraints: eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5):1157–1172, 2016.
- [25] Matthijs Langelaar. Topology optimization of 3D self-supporting structures for additive manufacturing. *Additive Manufacturing*, 12:60–70, 2016.
- [26] Matthijs Langelaar. An additive manufacturing filter for topology optimization of print-ready designs. *Structural and Multidisciplinary Optimization*, 55(3):871–883, 2017.
- [27] Xiaoping Qian. Undercut and overhang angle control in topology optimization: a density gradient based integral approach. *International Journal for Numerical Methods in Engineering*, 111(3):247–272, 2017.
- [28] Grégoire Allaire, Charles Dapogny, Rafael Estevez, Alexis Faure, and Georgios Michailidis. Structural optimization under overhang constraints imposed by additive manufacturing technologies. *Journal of Computational Physics*, 351:295–328, 2017.
- [29] Grégoire Allaire and Lukas Jakabčín. Taking into account thermal residual stresses in topology optimization of structures built by additive manufacturing. *Mathematical Models and Methods in Applied Sciences*, 28(12):2313–2366, 2018.
- [30] Takao Miki and Takayuki Yamada. Topology optimization considering the distortion in additive manufacturing. *Finite Elements in Analysis and Design*, 193:103558, 2021.
- [31] Grzegorz Misiun, Emiel van de Ven, Matthijs Langelaar, Hubert Geijselaers, Fred van Keulen, Ton van den Boogaard, and Can Ayas. Topology optimization for additive manufacturing with distortion constraints. *Computer Methods in Applied Mechanics and Engineering*, 386:114095, 2021.
- [32] Jeroen Pellens, Geert Lombaert, Martine Michiels, Tom Craeghs, and Mattias Schevenels. Topology optimization of support structure layout in metal-based additive manufacturing accounting for thermal deformations. *Structural and Multidisciplinary Optimization*, 61:2291–2303, 2020.
- [33] Ana Pais, Jorge Lino Alves, Renato Natal Jorge, and Jorge Belinha. Multiscale homogenization techniques for TPMS foam material for biomedical structural applications. *Bioengineering*, 10(5):515, 2023. doi: 10.3390/bioengineering10050515.

- [34] Lorenzo Dall’Ava, Harry Hothi, Johann Henckel, Anna Di Laura, Paul Shearing, and Alister Hart. Comparative analysis of current 3D printed acetabular titanium implants. *3D Printing in Medicine*, 5(1):15, 2019. doi: 10.1186/s41205-019-0052-0.
- [35] Bugatti Automobiles. World premiere: brake caliper from 3-D printer. <https://newsroom.bugatti.com/press-releases/world-premiere-brake-caliper-from-3-d-printer>, 2018. Accessed 2 July 2026.
- [36] TCT Magazine. McLaren W1 supercar features Divergent 3D printed suspension. <https://www.tctmagazine.com/mclaren-w1-divergent-3d-printed-suspension-supercar/>, 2025. Accessed 2 July 2026.
- [37] S. Mantovani, G. A. Campo, and M. Giacalone. Steering column support topology optimization including lattice structure for metal additive manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(21):10645–10656, 2020. doi: 10.1177/0954406220947121.
- [38] Lawrence Butcher. Powertrain focus: Czinger 21C, the 3D printed hypercar. Professional Motorsport World. <https://www.pmw-magazine.com/features/powertrain-focus-czinger-21c-the-3d-printed-hypercar.html>, 2022. Accessed 3 July 2026.
- [39] M. Seabra, J. Azevedo, A. Araújo, L. Reis, E. Pinto, M. Leite, and A. Deus. Selective laser melting (SLM) and topology optimization for lighter aerospace components. *Procedia Structural Integrity*, 1:289–296, 2016. doi: 10.1016/j.prostr.2016.02.039.
- [40] Emanuele Ghio and Emanuela Cerri. Additive manufacturing of AlSi10Mg and Ti6Al4V lightweight alloys via laser powder bed fusion: A review of heat treatments effects. *Materials*, 15(6):2047, 2022. doi: 10.3390/ma15062047.
- [41] LEAP 71. LEAP 71 announces successful manufacturing validation of massive rocket engine component with Nikon SLM Solutions. <https://leap71.com/2025/11/06/nikon-slm-leap71/>, 2025. Accessed 2 July 2026.
- [42] Relativity Space. Relativity space achieves major design and production milestones for Terran R. <https://www.relativityspace.com/press-release/2025/3/7/rycbar123>, 2025. Accessed 2 July 2026.
- [43] VoxelMatters. Code-first approach by LEAP 71 targets full-scale manufacturing of rocket engines. VoxelMatters. <https://www.voxelmatters.com/code-first-approach-by-leap-71-targets-full-scale-manufacturing-of-rocket-engines/>, 2025. Accessed 3 July 2026.
- [44] Pratt & Whitney. How Pratt & Whitney is 3D printing military engines. <https://www.prattwhitney.com/en/newsroom/news/2024/07/24/additive-manufacturing>, 2024. Accessed 2 July 2026.

- [45] Tobias Maconachie, Martin Leary, Bill Lozanovski, Xuezhe Zhang, Ma Qian, Omar Faruque, and Milan Brandt. SLM lattice structures: Properties, performance, applications and challenges. *Materials & Design*, 183:108137, 2019. doi: 10.1016/j.matdes.2019.108137.
- [46] Asliah Seharing, Abdul Hadi Azman, and Shahrum Abdullah. A review on integration of lightweight gradient lattice structures in additive manufacturing parts. *Advances in Mechanical Engineering*, 12(6):1687814020916951, 2020. doi: 10.1177/1687814020916951.
- [47] Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. Infill optimization for additive manufacturing—approaching bone-like porous structures. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1127–1140, 2018. doi: 10.1109/TVCG.2017.2655523.
- [48] Anders Clausen, Niels Aage, and Ole Sigmund. Exploiting additive manufacturing infill in topology optimization for improved buckling load. *Engineering*, 2(2):250–257, 2016. doi: 10.1016/J.ENG.2016.02.006.
- [49] Olivier Pantz and Karim Trabelsi. A post-treatment of the homogenization method for shape optimization. *SIAM Journal on Control and Optimization*, 47(3):1380–1398, 2008. doi: 10.1137/070688900.
- [50] Jeroen P. Groen and Ole Sigmund. Homogenization-based topology optimization for high-resolution manufacturable microstructures. *International Journal for Numerical Methods in Engineering*, 113(8):1148–1163, 2018. doi: 10.1002/nme.5575.
- [51] Jun Wu, Ole Sigmund, and Jeroen P. Groen. Topology optimization of multi-scale structures: a review. *Structural and Multidisciplinary Optimization*, 63(4):1455–1480, 2021. doi: 10.1007/s00158-021-02881-8.
- [52] Ifeanyichukwu Echeta, Xiaobo Feng, Ben Dutton, Richard Leach, and Samanta Pivano. Review of defects in lattice structures manufactured by powder bed fusion. *The International Journal of Advanced Manufacturing Technology*, 106:2649–2668, 2020. doi: 10.1007/s00170-019-04753-4.
- [53] M. Dallago, S. Raghavendra, V. Luchin, G. Zappini, D. Pasini, and M. Benedetti. The role of node fillet, unit-cell size and strut orientation on the fatigue strength of Ti-6Al-4V lattice materials additively manufactured via laser powder bed fusion. *International Journal of Fatigue*, 142:105946, 2021. doi: 10.1016/j.ijfatigue.2020.105946.
- [54] Oraib Al-Ketan and Rashid K. Abu Al-Rub. Multifunctional mechanical metamaterials based on triply periodic minimal surface lattices. *Advanced Engineering Materials*, 21(10):1900524, 2019. doi: 10.1002/adem.201900524.

- [55] Rebekka V. Woldseth, Niels Aage, J. Andreas Bærentzen, and Ole Sigmund. On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, 65(10):294, 2022. doi: 10.1007/s00158-022-03347-1.
- [56] Seungyeon Shin, Dongju Shin, and Namwoo Kang. Topology optimization via machine learning and deep learning: a review. *Journal of Computational Design and Engineering*, 10(4):1736–1766, 2023. doi: 10.1093/jcde/qwad072.
- [57] Ivan Sosnovik and Ivan Oseledets. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4):215–223, 2019. doi: 10.1515/rnam-2019-0018.
- [58] Yonggyun Yu, Taeil Hur, Jaeho Jung, and In Gwun Jang. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*, 59(3):787–799, 2019. doi: 10.1007/s00158-018-2101-5.
- [59] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. arXiv:1312.6114.
- [60] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2672–2680, 2014.
- [61] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020.
- [62] Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3):031715, 2021. doi: 10.1115/1.4049533.
- [63] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.
- [64] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.
- [65] Aaditya Chandrasekhar and Krishnan Suresh. TOuNN: Topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, 63(3):1135–1149, 2021. doi: 10.1007/s00158-020-02748-4.

- [66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 3320–3328, 2014.
- [67] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.
- [68] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1–2):81–93, 1938.
- [69] Stephen P. Timoshenko and James N. Goodier. *Theory of Elasticity*. McGraw-Hill, New York, 3rd edition, 1970.
- [70] ISO/ASTM. ISO/ASTM 52900:2021 additive manufacturing – general principles – fundamentals and vocabulary, 2021. International Standard, 2nd edition.
- [71] Silvia Vock, Burghardt Klöden, Alexander Kirchner, Thomas Weißgärber, and Bernd Kieback. Powders for powder bed fusion: a review. *Progress in Additive Manufacturing*, 4(4):383–397, 2019.
- [72] Saad A. Khairallah, Andrew T. Anderson, Alexander Rubenchik, and Wayne E. King. Laser powder-bed fusion additive manufacturing: physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Materialia*, 108:36–45, 2016.
- [73] Jamison L. Bartlett and Xiaodong Li. An overview of residual stresses in metal powder bed fusion. *Additive Manufacturing*, 27:131–149, 2019.
- [74] Tarasankar DebRoy, H. L. Wei, J. S. Zuback, T. Mukherjee, J. W. Elmer, J. O. Milewski, A. M. Beese, A. Wilson-Heid, A. De, and W. Zhang. Additive manufacturing of metallic components – process, structure and properties. *Progress in Materials Science*, 92:112–224, 2018.
- [75] Daniel Thomas. *The development of design rules for selective laser melting*. PhD thesis, University of Wales Institute, Cardiff, 2009.
- [76] Chun Kit Sit, Louis N. S. Chiu, Yunlong Tang, and Aijun Huang. Towards supportless laser powder bed fusion: fundamental understanding of the formation of 1st layer overhang through parameter optimisation and in-situ high-speed thermal imaging. *Virtual and Physical Prototyping*, 18(1):e2231906, 2023. doi: 10.1080/17452759.2023.2231906.
- [77] Dirk Herzog, Vanessa Seyda, Eric Wycisk, and Claus Emmelmann. Additive manufacturing of metals. *Acta Materialia*, 117:371–392, 2016.

- [78] Alborz Shokrani. Cost modelling for powder bed fusion and directed energy deposition additive manufacturing. *Journal of Manufacturing and Materials Processing*, 8(4):142, 2024.
- [79] Ralph I. Stephens, Ali Fatemi, Robert R. Stephens, and Henry O. Fuchs. *Metal Fatigue in Engineering*. John Wiley & Sons, New York, USA, 2nd edition, 2000.
- [80] O. H. Basquin. The exponential law of endurance tests. *Proceedings of the American Society for Testing and Materials*, 10:625–630, 1910.
- [81] John Goodman. *Mechanics Applied to Engineering*. Longmans, Green and Co., London, United Kingdom, 1899.
- [82] C. R. Söderberg. Factor of safety and working stress. *Transactions of the American Society of Mechanical Engineers*, 52:13–28, 1939.
- [83] Masanori Matsuishi and Tatsuo Endo. Fatigue of metals subjected to varying stress. In *Proceedings of the Kyushu Branch of the Japan Society of Mechanical Engineers*, pages 37–40, Fukuoka, Japan, 1968.
- [84] S. D. Downing and D. F. Socie. Simple rainflow counting algorithms. *International Journal of Fatigue*, 4(1):31–40, 1982. doi: 10.1016/0142-1123(82)90018-4.
- [85] ASTM International. ASTM E1049-85(2017): Standard practices for cycle counting in fatigue analysis. Technical report, ASTM International, West Conshohocken, PA, USA, 2017.
- [86] Arvid Palmgren. Die lebensdauer von kugellagern. *Zeitschrift des Vereines Deutscher Ingenieure*, 68(14):339–341, 1924.
- [87] Milton A. Miner. Cumulative damage in fatigue. *Journal of Applied Mechanics*, 12(3):A159–A164, 1945. doi: 10.1115/1.4009458.
- [88] Stefano Beretta, Luca Patriarca, Mohammadmahdi Gargourimotlagh, Andrew Hardaker, David Brackett, Shirin Salimian, Johannes Gumpinger, and Tommaso Ghidini. A benchmark activity on the fatigue life assessment of AlSi10Mg components manufactured by L-PBF. *Materials & Design*, 218:110713, 2022. doi: 10.1016/j.matdes.2022.110713.
- [89] Gianni Nicoletto. Influence of rough as-built surfaces on smooth and notched fatigue behavior of L-PBF AlSi10Mg. *Additive Manufacturing*, 34:101251, 2020. doi: 10.1016/j.addma.2020.101251.
- [90] Roland Rennert, Eckehard Kullig, Michael Vormwald, Alfons Esderts, and Michael Luke. *Analytical Strength Assessment of Components: Made of Steel, Cast Iron and Aluminium Materials. FKM Guideline*. VDMA Verlag, Frankfurt am Main, Germany, 7th edition, 2020.

- [91] Felix Sajadi, Jan-Marc Tiemann, Nooshin Bandari, Ali Cheloe Darabi, Javad Mola, and Siegfried Schmauder. Fatigue improvement of AlSi10Mg fabricated by laser-based powder bed fusion through heat treatment. *Metals*, 11(5):683, 2021. doi: 10.3390/met11050683.
- [92] Maedeh Amirpour, Dayna Cracknell, Amirali Amirian, and Arvin N. Alipour. Hybrid 3D printing of fluid-filled lattices for biomedical applications: a review. *The International Journal of Advanced Manufacturing Technology*, 136(10):4083–4105, 2025. doi: 10.1007/s00170-025-15079-9.
- [93] V. S. Deshpande, M. F. Ashby, and N. A. Fleck. Foam topology: bending versus stretching dominated architectures. *Acta Materialia*, 49(6):1035–1040, 2001. doi: 10.1016/S1359-6454(00)00379-7.
- [94] V. S. Deshpande, N. A. Fleck, and M. F. Ashby. Effective properties of the octet-truss lattice material. *Journal of the Mechanics and Physics of Solids*, 49(8):1747–1769, 2001. doi: 10.1016/S0022-5096(01)00010-2.
- [95] Lorna J. Gibson and Michael F. Ashby. *Cellular Solids: Structure and Properties*. Cambridge University Press, Cambridge, UK, 2 edition, 1997.
- [96] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.
- [97] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.
- [98] Shuai Zheng, Zhenzhen He, and Honglei Liu. Generating three-dimensional structural topologies via a U-Net convolutional neural network. *Thin-Walled Structures*, 159:107263, 2021. doi: 10.1016/j.tws.2020.107263.
- [99] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016. doi: 10.1109/3DV.2016.79.
- [100] Xiaofeng Guo, Xiaoyang Zheng, Xinyu Yang, and Yong Yi. Mechanical behavior of TPMS-based scaffolds: a comparison between minimal surfaces and their lattice structures. *SN Applied Sciences*, 1(10):1145, 2019. doi: 10.1007/s42452-019-1167-z.
- [101] Rashid K. Abu Al-Rub, Dong-Wook Lee, Kamran A. Khan, and Anthony N. Palazotto. Effective anisotropic elastic and plastic yield properties of periodic foams derived from triply periodic Schoen’s I-WP minimal surface. *Journal of Engineering Mechanics*, 146(5):04020030, 2020. doi: 10.1061/(ASCE)EM.1943-7889.0001759.

- [102] Dong-Wook Lee, Kamran A. Khan, and Rashid K. Abu Al-Rub. Stiffness and yield strength of architected foams based on the Schwarz Primitive triply periodic minimal surface. *International Journal of Plasticity*, 95:1–20, 2017. doi: 10.1016/j.ijplas.2017.03.005.
- [103] Diab W. Abueidda, Rashid K. Abu Al-Rub, Ahmed S. Dalaq, Dong-Wook Lee, Kamran A. Khan, and Iwona Jasiuk. Effective conductivities and elastic moduli of novel foams with triply periodic minimal surfaces. *Mechanics of Materials*, 95: 102–115, 2016. doi: 10.1016/j.mechmat.2016.01.004.
- [104] Bharath Bhushan Ravichander, Shweta Hanmant Jagdale, and Golden Kumar. Surface morphology, compressive behavior, and energy absorption of graded triply periodic minimal surface 316L steel cellular structures fabricated by laser powder bed fusion. *Materials*, 15(23):8294, 2022. doi: 10.3390/ma15238294.
- [105] James S. Bellino and Mukul Kumar. Investigating property-porosity relationships for micro-architected lattice structures. *Scientific Reports*, 16(1):5521, 2026. doi: 10.1038/s41598-026-35262-6.
- [106] Z. Hashin and S. Shtrikman. A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids*, 11(2):127–140, 1963. doi: 10.1016/0022-5096(63)90060-7.
- [107] T. Mori and K. Tanaka. Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metallurgica*, 21(5):571–574, 1973. doi: 10.1016/0001-6160(73)90064-3.
- [108] J. D. Eshelby. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the Royal Society of London A*, 241(1226):376–396, 1957. doi: 10.1098/rspa.1957.0133.
- [109] Reza Razani. Behavior of fully stressed design of structures and its relationship to minimum-weight design. *AIAA Journal*, 3(12):2262–2268, 1965.
- [110] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, Computer Graphics, pages 351–358, 1995. doi: 10.1145/218380.218473.
- [111] H. G. von Schnering and R. Nesper. Nodal surfaces of Fourier series: fundamental invariants of structured matter. *Zeitschrift für Physik B Condensed Matter*, 83(3): 407–412, 1991. doi: 10.1007/BF01313411.
- [112] William E. Lorensen and Harvey E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. doi: 10.1145/37402.37422.

- [113] William L. Oberkamp and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, UK, 2010. ISBN 978-0-521-11360-1.
- [114] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, New York, 4th edition, 2002. ISBN 978-0-471-35605-9.
- [115] Diogo da Fonseca Araújo Matos. Rolling damper, respective suspension system with asymmetric rolling movement and symmetric vertical movement, mounting method thereof. European Patent EP4147888B1, Irreverentdragon Lda. <https://patents.google.com/patent/EP4147888B1>, 2024. Granted 29 May 2024. Accessed 4 July 2026.
- [116] Ramada Aços. FR-3: Alloyed construction steel, technical data sheet. https://www.ramada.pt/en/products/steels/alloyed-construction-steel/fr-3_.html, 2024. Accessed 4 July 2026.
- [117] EOS GmbH. EOS aluminium AlSi10Mg: Material data sheet. EOS GmbH Electro Optical Systems. <https://www.eos.info/>, 2022. Accessed 4 July 2026.
- [118] Canopy Simulations Ltd. Canopy platform: Dynamic lap simulation. Canopy Simulations Ltd., Hampshire, United Kingdom, 2024. Available at <https://www.canopysimulations.com>.
- [119] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456. PMLR, 2015.
- [120] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning where to look for the pancreas. In *Medical Imaging with Deep Learning (MIDL)*, 2018. arXiv:1804.03999.
- [121] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- [122] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6980.
- [123] Juan Miguel Valverde, Dim P. Papadopoulos, Rasmus Larsen, and Anders B. Dahl. Towards high-quality image segmentation: Improving topology accuracy by penalizing neighbor pixels. Preprint, 2026. Venue and identifier to be confirmed.

- [124] Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H. Menze. clDice – a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16560–16569, 2021. doi: 10.1109/CVPR46437.2021.01629.