Buckling of Thin Perforated Plates



Elenor Imogen Naraidoo

Pembroke College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2026

Acknowledgements

Since first entering Oxford's Engineering Science Department in 2022, there have been countless 5am mornings (although I never quite got the hang of rowing), minutes spent crying (thank you Janet and Ada for nursing me with wine in the evening), calories burned (thank you to Matt for your brilliant gym classes, where I met lovely Emma, Daisy and Kexin), and jars of peanut butter consumed (I have eaten at least 10.8kg over the duration of my doctorate). In addition to my amazing friends at Pembroke College (Thomas, Abi, Alex, Dan, and Clem, to name a few), these people and things have made the past 3.5 years fly by.

Above all, I must thank my fantastic fairy god mother supervisor, Barbara. You are my role model; you made me feel that I *can*. You gave me your undivided attention whenever I needed help, you always had time and patience for me (despite doing a gazillion other things at once). If you were not involved with my thesis, it would have been a much gloomier struggle. I am so excited for the next chapter of my life with you, and I will be forever grateful for all of the opportunities you have made possible.

My professional gratitude extends to Mark, Nick and Paul at the University of Bath. Your passion for structural engineering was infectious; I would not be writing these words without you. For your guidance and the endless time you had for me, Burak, I would still be doing those compression tests today, if not for your help. Usman and Xiongfeng, thank you for dropping everything, and helping me with my simulations when things got a bit dire. Zhong, thank you for your lectures, and patience with my questions. Taniya, thank you to your crash-course in PINNs, and Ioannis, thank you for taking a chance on me, I really enjoyed working with you. Most importantly, my doctorate would not have been possible without the Athena Swan Engineering Science Studentship.

On a more personal level, thank you to my incredibly handsome, kind and selfless boyfriend. Pav, you have such a big heart, and you have taught me what truly matters in life. Thank you to my beautiful and intelligent little sister, Scarlett. You light me up, and I miss you every day. You don't know it yet, but you are going to do something so incredible with your life; your greatest enemy is only yourself. Finally, thank you to my wonderful parents, Lucy and Marcus. You gave me a brilliant happy childhood, and a fantastic education; you embraced every interest I wanted to pursue. Your love, patience and support for this weird and nerdy daughter of yours lights the fire behind everything I do.

Abstract

Thin perforated plates are used in beams, columns, rack sections and cladding, and are vulnerable to buckling under uniform compression. Buckling is dangerous as it can occur suddenly, however current methods used to design buckling-resistant perforated structures are limited by their complexity or accuracy or range of applicability. Design standards demand simple and intuitive methods, yet such methods are either outdated or only applicable to a narrow domain. Outside these bounds, finite element (FE) analysis can be used at significant computational cost, requiring the geometry to be discretised, and boundary conditions to be applied with considerable modelling judgement. Subsequently, structural optimisation is slow, repetitive, and difficult to implement efficiently in practice. To address these limitations, this research develops a series of semi-empirical and semi-analytical methods, alongside a new optimisation framework.

The numerical studies used to derive many of the expressions developed in this thesis are validated by a novel series of uniform compression tests on perforated aluminium plates with various perforation patterns. Digital image correlation is used to capture the initial geometric imperfection to inform geometrically and materially nonlinear FE models, replicating the tests until failure.

To improve applicability, a simple rule-based classification algorithm is proposed, to determine the local buckling mode governing failure. Accompanying this, a unified set of semi-empirical expressions is established, predominantly controlled by the number of rows of perforations. The failure mode of plates with tall holes is distinct, governed particularly by the spacing between perforations, and is classified and assessed separately. The ultimate buckling capacity of perforated plates is controlled by a more complex geometric variable, which can distinguish between two failure modes, and is used to propose a Direct Strength Method expression. For most of the semi-empirical expressions developed, a reliability-based model factor is also determined, permitting direct implementation into design standards.

To improve generality, semi-analytical methods are developed to predict the elastic buckling capacity of perforated plated across a broader domain than empirical expressions. A mathematical formulation based on Rayleigh's energy method is first derived to evaluate plates with perforations of non-uniform size and distribution, reducing reliance on FE procedure. In practice, FE is required when a design falls outside the scope of standards; however, as such analyses are heavily user-dependent and time-consuming, FE cannot be efficiently implemented in optimisation routines. Artificial Intelligence (AI) therefore

offers a new paradigm for structural design, assessment and optimisation. This thesis develops a novel physics-informed neural network (PINN), which can determine the elastic buckling capacity of a plate with a single eccentric perforation, and can obtain accurate results immediately. With further development, the PINN presents a powerful and reliable alternative to conventional numerical analysis, with the potential to significantly accelerate structural workflows.

Finally, an intuitive optimisation framework is developed, enabling the arrangement of perforations in a buckling plate to be optimised without relying on FE. Using the semi-empirical expressions established in this thesis, a stochastic shape annealing optimisation framework efficiently explores the design space, and can identify geometries that outperform those found with a deterministic FE-based process. This approach leads to novel perforation layouts, which can exceed the capacity of traditional designs. This optimisation framework is simple, interpretable, and readily applicable; it is implemented as an open-source plug-in to maximise its impact in engineering design.

Altogether, this work establishes a unified framework for the analysis, design, and optimisation of perforated plates subject to buckling. These methods perform accurately within known domains, reduce our reliance on numerical methods, and enable more efficient structural design.

Contents

List of Figures **xi**

List of Tables **xviii**

List of Abbreviations **xxii**

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Thesis Overview . . . 6
- 1.3 Important Concepts . . . 8
  - 1.3.1 Elastic Buckling Behaviour . . . 8
  - 1.3.2 Perforation Geometry . . . 9
  - 1.3.3 Slenderness and Post-Critical Behaviour . . . 11
- 1.4 Statement of Authorship . . . 12

2 Literature Review **15**
- 2.1 Methods for the Prediction of Buckling Capacity . . . 15
  - 2.1.1 Parametric Numerical Studies . . . 15
  - 2.1.2 Semi-Analytical Formulations . . . 16
  - 2.1.3 Energy-Based Methods . . . 20
  - 2.1.4 Physics-Informed Neural Networks . . . 21
  - 2.1.5 Overview . . . 23
- 2.2 Observed Buckling Behaviour of Perforated Plates . . . 28
  - 2.2.1 Perforation Size and Shape . . . 28
  - 2.2.2 Perforation Location . . . 30
  - 2.2.3 Slenderness of a Perforated Plate . . . 32
  - 2.2.4 Local Buckling Modes . . . 33
- 2.3 Optimisation . . . 34
- 2.4 Experiment Design . . . 36
- 2.5 Design Methods in Current Standards . . . 37
  - 2.5.1 The Direct Strength Method (DSM) . . . 37
  - 2.5.2 The Effective Width Method (EWM) . . . 40
- 2.6 Code Constraints and Geometric Limits . . . 41

2.6.1 Sections with Large Web Openings [6] . . . 41
2.6.2 Rack Sections and Perforated Cold-Formed Members [93] . . . 42

3 Elastic Buckling of Slender Steel Plates with Perforations 43
3.1 Commentary . . . 43
3.2 Introduction to Plate Buckling Theory . . . 44
3.3 Calibration of Finite Element Model . . . 46
3.4 Parametric Assessment . . . 48
3.5 Modified Energy Prediction . . . 52
3.6 Discussion and Conclusion . . . 55

4 Local Buckling Load of a Perforated Plate, a Computational Study 58
4.1 Commentary . . . 58
4.2 Introduction . . . 59
4.3 Research Objective and Hypotheses . . . 60
4.4 Finite Element Model . . . 61
4.4.1 Modelling Conditions . . . 61
4.4.2 Parametric Assessment . . . 63
4.5 Local Buckling Mode Algorithm . . . 65
4.6 New Prediction Method . . . 69
4.6.1 An Expression for plates with $n_r > 1$ . . . 70
4.6.2 An Expression for plates with $n_r = 1$ . . . 74
4.7 Prediction Formulae for $\widehat{N}_{cr,h}$ . . . 76
4.8 Conclusion . . . 79

5 An Experimental, Numerical and Analytical Study of Local Buckling in Perforated Plates 81
5.1 Commentary . . . 81
5.2 Introduction . . . 83
5.3 Experiment Design . . . 85
5.3.1 Load Frame . . . 85
5.3.2 Digital Image Correlation . . . 87
5.3.2.1 Speckle Pattern and Subset Size . . . 88
5.3.2.2 Hardware Setup . . . 88
5.3.2.3 Software Setup . . . 90
5.3.3 Specimen . . . 90
5.3.3.1 Material Properties . . . 90
5.3.3.2 Dimensions . . . 91
5.3.3.3 Perforation Patterns . . . 92
5.3.3.4 Initial Imperfection . . . 95

5.3.4 Issues with Experimental Testing . . . 99
5.4 GMNIA ABAQUS Model . . . 102
5.4.1 Convergence Study . . . 102
5.4.2 Boundary Conditions . . . 103
5.4.3 Loading and Imperfection Modelling . . . 104
5.4.4 Validation and Accuracy . . . 106
5.5 Results . . . 108
5.5.1 Load Displacement Curves . . . 108
5.5.2 Deformation . . . 114
5.5.2.1 Elastic Failure Mode . . . 114
5.5.2.2 Plastic Failure Mode . . . 116
5.5.3 Influence of Perforations on Performance . . . 118
5.6 Prediction of the Buckling Capacity . . . 120
5.7 Conclusion . . . 123

6 The Direct Strength Method for Plates with Perforations 125
6.1 Commentary . . . 125
6.2 Introduction . . . 126
6.3 Applicability of Existing DSM Equations to Experimental Results . . . 127
6.4 New DSM Equations for Regular and Staggered Perforations . . . 131
6.4.1 GMNIA Model . . . 131
6.4.2 Study Domain . . . 135
6.4.2.1 Regular Perforations . . . 136
6.4.2.2 Staggered Perforations . . . 137
6.4.3 Numerical Results . . . 138
6.4.4 New DSM Approach . . . 143
6.5 Final Comparison and Reliability Factor . . . 145
6.6 Conclusion . . . 149

7 The Prediction and Classification of Local Buckling for Plates with Tall Perforations 151
7.1 Commentary . . . 151
7.2 Introduction: Positioning the Problem . . . 153
7.3 Finite Element Model . . . 155
7.4 Prediction for Elastic Buckling Load . . . 157
7.4.1 Designation 1: Out-of-Plane Plate-to-Hole elastic buckling load . . . 158
7.4.2 Designation 2: Out-of-Plane Hole-to-Hole elastic buckling load . . . 162
7.4.3 Designation 3: In-Plane elastic buckling load . . . 164
7.4.4 Classification of elastic buckling load . . . 166
7.5 Overall Performance . . . 167

7.6 Optimising Capacity . . . 170
7.6.1 Elastic Capacity . . . 170
7.6.2 Plastic Capacity . . . 173
7.7 Conclusion . . . 180

8 Shape Annealing Optimisation of Perforations in Rack Sections Subject to Buckling 182
8.1 Commentary . . . 182
8.2 Introduction . . . 184
8.3 Shape Annealing Procedure . . . 186
8.4 Derivation of Spacing Heuristics for Optimisation . . . 189
8.4.1 Column Spacing s_x . . . 190
8.4.2 Row Spacing s_y . . . 193
8.5 Shape Annealing Optimisation . . . 195
8.5.1 Comparison to a Deterministic Algorithm . . . 195
8.5.2 Optimisation of a Rack Section . . . 197
8.5.2.1 Case Study 1 - One Row of Perforations . . . 198
8.5.2.2 Case Study 2 - One Column of Perforations . . . 200
8.5.2.3 Case Study 3 - Multiple Rows and Columns of Perforations 203
8.5.3 Blended Geometries . . . 206
8.6 Conclusion . . . 210

9 A Physics Informed Neural Network for the elastic buckling load of a Perforated Plate 213
9.1 Commentary . . . 213
9.2 Introduction . . . 214
9.3 Literature Review . . . 215
9.3.1 Architecture . . . 215
9.3.2 Optimisation . . . 217
9.3.3 Collocation Point Sampling . . . 218
9.4 Mathematical Background to the Problem . . . 219
9.4.1 Strong-form . . . 219
9.4.2 Weak-form . . . 221
9.5 Surrogate Solution to the Problem . . . 222
9.5.1 Feed-Forward Neural Network (FNN) . . . 222
9.5.2 Physics-Informed Neural Network (PINN) . . . 222
9.5.2.1 PINN Architecture . . . 222
9.5.2.2 PINN Losses . . . 223
9.6 Unperforated Plate . . . 225
9.6.1 Strong-Form . . . 225

9.6.2 Weak-Form . . . 227
9.6.3 Oversampling . . . 228
9.7 Perforated Plate . . . 229
9.7.1 Mathematical Background . . . 229
9.7.2 PINN with One Input . . . 231
9.7.3 PINN with Four Inputs . . . 235
9.7.4 PINN with Many Inputs . . . 239
9.7.4.1 Grid Search . . . 239
9.7.4.1.1 Quantitative Analysis . . . 243
9.7.4.1.2 Qualitative Analysis . . . 243
9.7.4.1.3 Extended Training . . . 244
9.7.4.2 Testing . . . 248
9.8 Conclusion . . . 254

10 Conclusion **256**
10.1 Solution 1: Empirical, Standard-Ready Predictions . . . 257
10.2 Solution 2: Reliable Alternatives to Numerical Analysis . . . 263
10.3 Solution 3: Intuitive Optimisation . . . 265
10.4 Final remarks . . . 266

Appendices

A Mesh Convergence Study **268**

B Linear And Nonlinear ABAQUS Input File Creator **270**

C DIC Data ABAQUS Input File Creator **298**

D Code to Measure Geometric Imperfection **321**

E Load Frame Drawings **327**

F Nonlinear Model Setup **334**

G Mathematical Formulations **336**
G.1 Chapter 6 . . . 336
G.1.1 Modified Slenderness $\lambda_{l,m}$ . . . 336
G.1.2 Geometric Parameter Ω . . . 338
G.2 Chapter 7 . . . 339
G.2.1 Introduction . . . 339
G.2.2 Scale Factor ψ_1 . . . 341
G.2.3 Scale Factor ψ_2 . . . 342
G.2.4 Scale Factor ψ_3 . . . 343
G.2.5 Conclusion . . . 344

H Typical Rack Sections **345**

I Abalone **348**
I.1 Construct . . . 350
I.2 Constrain . . . 351
I.3 Optimise . . . 352
I.4 Undo . . . 354
I.5 Deconstruct . . . 355
I.6 Mesh . . . 355

J Physics-Informed Neural Network **357**
J.1 Main Code . . . 357
J.2 Functions . . . 366
J.2.1 Math Disp . . . 366
J.2.2 Parametric PINNS Functions . . . 370
J.2.3 Prediction . . . 381
J.2.4 ReLoBRaLo V2 . . . 385
J.2.5 Utilities . . . 388

References **393**

List of Figures

1.1 Structural members with perforations, adapted from [11–17] 3
1.2 Methods for the design of thin, perforated structural members. Methods in **bold** are introduced in this chapter, whereas methods in *italics* are introduced or developed later 4
1.3 The modeshape of a simply supported buckling plate at the critical elastic buckling load $N_x = N_0$ 8
1.4 Perforation geometry defined in this thesis 10
1.5 Ultimate load-displacement curves for slender and stocky plates. The 'perfect' elastic plate behaviour is shown in black. 11

2.1 Variables considered by [2] 18
2.2 The influence of perforation shape and location on $N_{cr,h}$ (high/low) for (a) a concentric perforation in $a/b = 1$, (b) a concentric perforation in $a/b = 2$, (c) a concentric perforation in $a/b = 3$, (d) a perforation in the 'end panel', (e) a concentric perforation as $a/b \to \infty$. The perforated plates highlighted with the bold box exhibit 'wavelength stiffening' 28
2.3 Stress distribution in plates uniformly compressed on the left and right edges. Rectangular and elliptical holes (a) superior stress distribution and (b) inferior stress distribution, reproduced from [39] 29
2.4 Areas of a buckling plate 30
2.5 Buckling coefficient (k) when the centre of a circular hole is located at a buckling minimum. Wavelength stiffening is observed where the modeshape either gains or loses a buckling halfwave. Reproduced with permission from [72] 31
2.6 Geometric slenderness (b/t) for a square plate with 25 perforations as the perforation ratio r_p varies. The slenderness limit is shown in black. Adapted with permission from [4] 33
2.7 Geometric slenderness (b/t) for plates of aspect 2 with different yield strengths f_y as the perforation size varies. The elastic zone is shaded. Adapted with permission from [34] 33
2.8 Local buckling modes of perforated plates (a) 'whole local buckling', (b) deformation dampened at a very large perforation, (c) 'unstiffened strip' buckling, (d) 'Euler strip' buckling 33

2.9 Geometry of sections with tall holes, modified from [6] 42

3.1 Local buckling manifestations of perforated plates (a) Whole Plate Buckling (WPB), (b) Lateral Perforation Buckling (LPB), (c) Unstiffened Strip Buckling (USB), (d) Euler Strip Buckling (ESB) 44
3.2 Simply supported boundary conditions and nodal loading simulated in ABAQUS 47
3.3 Regular variables examined 47
3.4 Accuracy of the prediction by (a) $N_{cr,S}$ [4] and (b) $N_{cr,SM}$ [3] in comparison to FE results for a regular array of perforations 49
3.5 Examples of examined plates with a regular arrangement of perforations . 49
3.6 Accuracy of the prediction $N_{cr,S}$ by [4] in comparison to FE results for an irregular array of perforations. Closer detail is provided on the right. . . . 51
3.7 Examples of examined plates with an irregular arrangement of perforations 51
3.8 Assumptions for the Modified Energy Prediction (a) The force distribution within an unperforated plate (b) The force distribution within a perforated plate (c) A length B'B' along the adjacent surface above the hole will displace by a function $w(x, y)$ (grey). The unperforated plate function $w_0(x, y)$ is assumed (red) 52
3.9 (a) Nomenclature of perforation co-ordinates, and (b) approximate uniform force function $F_{x,i}$ shown in red for each of the i holes 55
3.10 Accuracy of (a) the prediction $N_{cr,S}$ by [4], and (b) the Modified Energy Prediction $\widehat{N}_{cr,h}$ in comparison to ABAQUS results for all 806 parametric models 55

4.1 Local buckling modes of perforated plates 60
4.2 Simply supported boundary conditions and nodal loading simulated in ABAQUS 63
4.3 The load ratio of the perforated plate to the unperforated plate with respect to perforation ratio for all 870 parametric models 64
4.4 Distribution of load ratio for all 870 models in the dataset 64
4.5 (a) WPB involves the net plate area (b) LPB at the lateral portion between the perforation and plate edge (c) ESB at the transverse portions between perforations (d) USB at the transverse portion between the perforation and plate edge. 66
4.6 The algorithm used to determine the buckling mode corresponding to the geometry of a plate with an array of rectangular perforations. 'Yes' and 'No' are blue and red respectively. The perforation variables are defined in Figure 1.4 67

4.7 Relationship between elastic buckling load ratio and perforation ratio (r_p) for plates with $n_r > 1$. *Please note that these figures are taken from [23]; the reference to [26] in the legend is [4] in this thesis* 69
4.8 Relationship between elastic buckling load ratio and perforation ratio (r_p) for plates with $n_r = 1$ *Please note that these figures are taken from [23]; the reference to [26] in the legend is [4] in this thesis* 69
4.9 Relationship between the load ratio and τ for plates with $n_r > 1$ 72
4.10 Relationship between load ratio and τ for plates exhibiting WPB, INT or USB with $n_r = 1$ 74
4.11 Relationship between the load ratio and ω for plates exhibiting WPB, INT or USB with $n_r = 1$ 74
4.12 Relationship between predicted ($\widehat{N}_{cr,h}$) and FE ($N_{cr,FE}$) elastic buckling loads for all 775 considered geometries, including WPB, ESB, USB, INT, excluding LPB (points in the green region are conservative). 76
4.13 Relationship between predicted (N_{Ed}) and FE ($N_{cr,FE}$) elastic buckling loads for all 775 considered geometries, including WPB, ESB, USB, INT, excluding LPB (Figure 4.1). Points in the green region are conservative . . 78

5.1 Examples of perforated plates 84
5.2 Local buckling modes of perforated plates (a) Whole Plate Buckling, WPB (b) Lateral Perforation Buckling, LPB (c) Unstiffened Strip Buckling, USB (d) Euler Strip Buckling, ESB 84
5.3 Design of loading frame and grooves permitting simply supported boundary conditions 86
5.4 Hardware setup, with loading rig partially installed (actuator attachment and specimen absent), see Appendix E 89
5.5 True stress-strain curve 91
5.6 A selection of specimen with regular, staggered, patterned and irregular perforations 94
5.7 Example of skewed imperfect plate with reference mid-surface and three points in red 99
5.8 Example of good subset (Test 13, top); Example of poor subset (Test 11, bottom) 101
5.9 Aspect ratio of specimen lies near the halfwave-transition point 102
5.10 DuoDIC 3DPoint data used to create the ABAQUS mesh 103
5.11 Convergence study for S4R elements 103
5.12 Boundary conditions shown by double-headed arrows. Load is applied in the u-direction (blue) at a reference point (grey) coupled to the edge . . . 104
5.13 A comparison of different GMNIA models 106

5.14 Force-displacement relationship for the unperforated plate. *Please note that this figure is taken from [25], where Ref. [44] is Ref. [146] in this thesis* . . 108
5.15 Force-Displacement Plots (Tests 2-7) 111
5.15 Force-Displacement Plots (Tests 8-13) 112
5.15 Force-Displacement Plots (Tests 14-18) 113
5.16 Comparison of plastic deformation in the GMNIA and experimental results 117
5.17 Variables considered by [2] reproduced from [23] 121

6.1 An example of a plate with a regular arrangement of perforations (Test E) 128
6.2 A critical length and a critical width in a staggered arrangement of perforations (Test I) 128
6.3 A critical length and a critical width in a staggered arrangement of perforations (Test J) 128
6.4 Nominal ultimate capacity determined with current DSM approaches compared to experimental result 130
6.5 Ramberg-Osgood model fitted to measured data 132
6.6 Examples of GMNIA models examined in this study 135
6.7 Relevant variables 136
6.8 Staggered arrangements of perforations examined 137
6.9 Load-displacement curves 138
6.10 Relationship between capacity ratio and perforated slenderness ($\lambda_{l,h}$), for plates with a regular arrangement of perforations. The points are coloured by their local plate buckling mode (see Figure 4.1) 139
6.11 Relationship between capacity ratio and perforated slenderness ($\lambda_{l,h}$), for plates with a staggered arrangement of perforations. The points are coloured by their staggered arrangement (see Figure 6.8) 139
6.12 Relationship between capacity ratio and modified slenderness ($\lambda_{l,m}$), for plates with a regular arrangement of perforations. The points are coloured by their local plate buckling mode (see Figure 4.1) 140
6.13 Relationship between capacity ratio and modified slenderness ($\lambda_{l,m}$), for plates with a staggered arrangement of perforations. The points are coloured by their staggered arrangement (see Figure 6.8) 140
6.14 The behaviour and stress distribution in perforated plates at failure 142
6.15 A suite of DSM curves with different κ values for all 300 models 145
6.16 Accuracy of the prediction for all GMNIA and experimental results (labelled) 145

7.1 Modes of failure in webs with tall perforations, adapted from [119] 153
7.2 Model for buckling of web between (a) widely spaced rectangular openings and (b) closely spaced circular openings, adapted from [6] 154

7.3 Nodal constraints and loading in FE model. Nodes are restrained in the coloured direction shown. Loads are applied at reference points and constrained to nodes along the plate width 156
7.4 Relationship between $N_{cr,FE}/N_0$ and S_{xo}/b with the best-fit curves shown for four values of l_y/b 158
7.5 The elastic buckling coefficient $k_{f,1}$ corrected for the limiting ratio $S_{xo,T}$ . . 161
7.6 Equation 7.3 plotted as the side aspect ratio varies 162
7.7 Computational results reproduced from [35] 162
7.8 Simply supported plate buckling curves for a plate of length a, and width b 163
7.9 Proposed buckling curves for plates exhibiting Designation 2 buckling . . . 163
7.10 Lattice-like Buckling 165
7.11 The relationship between the capacity predicted with Equation 7.8 and the spacing s_x depending on n_c 165
7.12 Accuracy score for 25 random seeds 166
7.13 Example of results for a random seed of 25 166
7.14 Neural network for out-of-plane buckling mode classification 167
7.15 Confusion matrix for parametric database in Table 7.8 168
7.16 Relationship between the predicted capacity and true capacity predicted with equations proposed in Table 7.7, for models examined in Section 7.5 . 168
7.17 Perforated-to-unperforated elastic buckling load ratio as perforation ratio varies 169
7.18 $k_f = \min(k_{f,1}, k_{f,2})$ for $a = 200$mm, $b = 100$mm, $l_x = 10$mm, $s_x = 10$mm (a) $n_c = 2$, (b) $n_c = 6$. 170
7.19 $k_{f,1}$ - to - $S_{xo,i}/b$ relationship as s_x is varied 171
7.20 Confusion matrix for parametric database in Section 7.6.1 (Table 7.9) . . . 172
7.21 Relationship between the predicted capacity and true capacity predicted with equations proposed in Table 7.7 for models examined in Section 7.6.1 172
7.22 Group I (perforated plates with $N_{u,FE} < 7000$ N) 175
7.23 Group II (perforated plates with $7000 \leq N_{u,FE} < 12000$ N) 176
7.24 The ratio of $\min(N_{cr,FE}, N_{u,FE})$ to N_0 of plates exhibiting localised failure (circles) and global failure (triangles) form two clusters 178
7.25 The plastic-to-elastic capacity ratio of models exhibiting localised failure (circles) and global failure (triangles) 178
7.26 The different out-of-plane buckling designations 178

8.1 Rack frame assembly with perforated column highlighted, and storage access shown with the arrow 185
8.2 Probability of acceptance for $\phi_{i+1} = 0.9\phi_i$ as step number (x) varies 187

8.3 The initial state could be transformed to one of four final states depending on whether rule application violates geometric constraints. The number of columns is n_c and the number of rows is n_r . . . 188
8.4 The nodal distance is shown by the arrow for the 1st and 7th hole with a centre at $c_{x,i}$. The location of the nodes for a plate buckling into 6 halfwaves are $c_{0,j}$ . . . 190
8.5 The relationship between capacity ratio and $\sum |c_{x,i} - c_o|$ for models examined in Section 8.4.1 . . . 192
8.6 The relationship between capacity ratio and s_y for models examined in Section 8.4.2 . . . 194
8.7 Depiction on how an initially less efficient path could terminate more efficiently 196
8.8 The most efficient result found with the deterministic approach . . . 197
8.9 Results found in Case Study 1 . . . 200
8.10 'Best' and 'worst' prediction in the result pool as l_x/l_y varies . . . 202
8.11 Efficiency of best result in Case Study 2 as l_x/l_y varies . . . 202
8.12 How the sigmoid factor λ on the objective function varies with n_c (Equation 8.4) . . . 203
8.13 Efficiency given by Equation 8.2 for the best results found in Case Study 3, as l_x/l_y varies . . . 205
8.14 Efficiency of best result in Case Study 3 as l_x/l_y varies . . . 206
8.15 The optimal result found with reciprocal and exponential cooling for (a) $l_x/l_y = 0.4$ and (b) $l_x/l_y = 0.8$ . . . 206
8.16 Efficiency of all Case Studies (CS) found with FE, plotted with respect to average hole aspect ratio . . . 208

9.1 Strong-form results for forward PINN . . . 225
9.2 Strong-form results for inverse PINN . . . 226
9.3 Strong-form inverse PINN . . . 226
9.4 Weak-form forward PINN . . . 228
9.5 Sampling collocation points in a square plate with loading on the left and right edges . . . 229
9.6 Example of Sobol sampling into 5 intervals ($I_n = 5$), with the points along the hole border identified in red . . . 230
9.7 Modeshape predicted by PINN trained on one input . . . 237
9.8 Modeshape predicted by PINN trained on four inputs . . . 238
9.9 Training after 500 epochs - Test 13 (Table 9.6) . . . 243
9.10 Training after 500 epochs - Test 17 (Table 9.6) . . . 243
9.11 Training after 500 epochs - Test 21 (Table 9.6) . . . 244
9.12 Training after 500 epochs - Test 22 (Table 9.6) . . . 244
9.13 Log loss as the number of training epochs increases . . . 245

9.14 Predictions for the modeshape on examples from the training dataset (Table 9.6). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$) 246
9.15 Predictions for the modeshape on examples from the validation dataset with smaller holes (Table 9.7). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$) 247
9.16 Predictions for the modeshape on examples from the validation dataset with larger holes (Table 9.7). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$) 248
9.17 The modeshape of the three best predictions in Test Set 1 (Table 9.9) . . . 250
9.18 The modeshape of the three worst predictions in Test Set 1 (Table 9.9) . . 251
9.19 Three best predictions in Test Set 2 (Table 9.10) 253
9.20 Three worst predictions in Test Set 2 (Table 9.10) 253

A.1 Mesh convergence study for S8R5 elements (Chapter 3 and Chapter 4) . . 269

G.1 The final expressions for f_1 and f_2, parametrised by ψ_i, as given in Chapter 7 340
G.2 The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.2 . . . 341
G.3 The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.3 . . . 342
G.4 The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.4 . . . 343
G.5 Examples of f_2 with different expressions for ψ_i 344

I.1 Abalone component network pipeline 349
I.2 Abalone construct component 350
I.3 Abalone constrain component 351
I.4 Abalone optimise component 352
I.5 Abalone undo component 354
I.6 Abalone deconstruct component 355
I.7 Abalone mesh component 355
I.8 An example of a perforated plate brep after optimisation with Abalone . . 356

List of Tables

2.1 Advantages and disadvantages of various published methods 24
2.3 Dimensional limits imposed by [6] on web openings 41

3.1 Domain of variables examined – regular arrangement of perforations (see Figure 3.3) 48
3.2 Domain of variables examined – irregular arrangement of perforations (see Figure 3.3) 48

4.1 Domain of variables examined in parametric study (perforation variables defined in Figure 1.4) 63
4.2 Functions $f(X)$ and $g(x_i)$ where i takes values from 1 to n for plates with $n_r > 1$ 71
4.3 R^2 test performance for plates with $n_r > 1$, assessed for # models exhibiting each mode 72
4.4 Functions $f(X)$ and $g(x_i)$ where i takes values from 1 to n for plates with $n_r = 1$ 75
4.5 R^2 test performance for plates with $n_r = 1$, assessed for # models exhibiting each mode 75
4.6 Mean (μ), COV and R^2 test performance for the elastic buckling load of a perforated plate found with FEM to that found with various methods, for different subsets of n parametric models, and different local buckling modes (Figure 4.1) 77
4.7 The factor γ determined for each mode with n models at the 5% and 1% significance level 78

5.1 True stress-strain values 91
5.2 Perforation parameters for each test (relevant variables defined in Figure 5.6) 93
5.3 Description of observed buckling deformation 96
5.4 Percentage pixel error 100
5.5 Predicted deformation mode, critical and ultimate capacity of tests and FE models 114
5.6 Accuracy of elastic buckling load predictions, and efficiency 122

6.1 Experimentally-found ultimate strengths (in Newtons) of tested aluminium perforated plates . . . 128
6.2 True stress-strain values . . . 132
6.3 Material parameters considered in FE models . . . 133
6.4 Control parameters considered in FE models . . . 133
6.5 Time incrementation parameters considered in FE models . . . 134
6.6 Default field parameters considered in FE models . . . 134
6.7 Domain of variables examined in parametric study for regular perforations (see Figure 6.7 for the variable definitions) . . . 136
6.8 Domain of variables examined in parametric study for staggered perforations (see Figure 6.7 for the variable definitions) . . . 137
6.9 Elastic failure mode of models with a regular arrangement of perforations . 143
6.10 Optimised parameters for each subset . . . 143
6.11 Comparison of the accuracy of the modified approach by [5] ($N_{u,YR}$) and Naraidoo &Rossi ($\widehat{N}_{u,h}$, Equation 6.12), for plates with a regular (R) and staggered (S) arrangement of perforations . . . 146
6.12 The accuracy of Equation 6.12 in comparison to experimental tests for plates with a regular or staggered arrangement of perforations, conducted in [23] . . . 147
6.13 Coefficients used to determine the LRFD resistance factor Φ . . . 148

7.1 Material parameters considered in FE models, in accordance with [24] . . . 155
7.2 Control parameters considered in FE models . . . 156
7.3 Description of observed buckling deformation . . . 157
7.4 Domain of variables examined in Section 7.4.1 (perforation variables summarised in Figure 1.4) . . . 158
7.5 Domain of variables examined in Section 7.4.2 (perforation variables summarised in Figure 1.4) . . . 162
7.6 Domain of variables examined in Section 7.4.3 (perforation variables summarised in Figure 1.4) . . . 164
7.7 Summary of equations for each buckling mode . . . 167
7.8 Domain of variables examined in section 7.5 (perforation variables summarised in Figure 1.4) . . . 168
7.9 Domain of variables examined in Section 7.6.1 (perforation variables summarised in Figure 1.4) . . . 172
7.10 Observed failure and relative elastic buckling load . . . 173
7.11 Geometries with least and greatest $N_{u,FE}/N_{cr,FE}$ (shown in increasing order) 179
7.12 Geometries with least and greatest min($N_{cr,FE}$, $N_{u,FE}$) (shown in increasing order) . . . 179

8.1 Domain of variables examined to determine optimal spacing s_x (relevant parameters summarised in Figure 1.4) . . . 190
8.2 Heuristic rules for s_x (relevant parameters summarised in Figure 1.4) . . . 191
8.3 Domain of variables examined to determine optimal spacing s_y (relevant parameters summarised in Figure 1.4) . . . 193
8.4 Heuristic rules for s_y (relevant parameters summarised in Figure 1.4) . . . 193
8.5 Domain of variables examined in Section 8.5.1 (relevant parameters summarised in Figure 1.4). Units of mm. . . . 196
8.6 The most efficient results found with shape annealing . . . 196
8.7 Dimensions of rack section web to optimise . . . 198
8.8 Geometric constraints on the rack section web to optimise (relevant parameters summarised in Figure 1.4) . . . 198
8.9 Additional geometric constraints imposed on optimisation in Case Study 1 (Section 8.5.2.1) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm) . . . 199
8.10 Additional geometric constraints imposed on optimisation in Case Study 2 (Section 8.5.2.2) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm) . . . 201
8.11 Additional geometric constraints imposed on optimisation in Case Study 3 (Section 8.5.2.3) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm) . . . 204
8.12 Blended geometries in Case Study 4 . . . 209

9.1 Four possible PINN variants . . . 223
9.2 Geometries considered in a study on the PINN with one input, and variables are summarised in Figure 1.4. All dimensions in mm; the measurement origin is at the bottom left corner of the plate . . . 232
9.3 A comparison of the buckling modeshape found using FE and PINN with one input . . . 233
9.4 PINN trained with one input . . . 237
9.5 PINN trained with four inputs . . . 238
9.6 Data used to train the many-input PINN (variables summarised in Figure 1.4) 240
9.7 Data used to validate the many-input PINN (variables summarised in Figure 1.4) . . . 241
9.8 Grid search results for PINNs with T_{layers} and $T_{neurons}$ . . . 242
9.9 Test set of models more similar to the training domain (Test Set 1) (variables summarised in Figure 1.4) . . . 250
9.10 Test set of models less similar to the training domain (Test Set 2) (variables summarised in Figure 1.4) . . . 252

10.1 Proposed semi-empirical standard-ready design equations (refer to referenced equation for undefined variables) 261

G.1 Values used to parametrise the scale factors ψ in Chapter 7 339
G.2 Values used to parametrise the scale factors ψ in Section G.2.2 341
G.3 Values used to parametrise the scale factors ψ in Section G.2.3 342
G.4 Values used to parametrise the scale factors ψ in Section G.2.4 343

List of Abbreviations

a . . . . . Length of a plate

b . . . . . Width of a plate

D . . . . Plate flexural constant

E . . . . Young's Modulus

$E_{0.2}$ . . . Secant modulus at 0.2% proof strain

e_x . . . . The average eccentricity of all perforations about the centre of the plate, in the direction of the plate length

e_y . . . . The average eccentricity of all perforations about the centre of the plate, in the direction of the plate width

f_u . . . . Net section ultimate strength

f_y . . . . Net section yield strength

$k_{f,1}$ . . . Plate buckling coefficient for tall perforations buckling with LPB designation 1

$k_{f,2}$ . . . Plate buckling coefficient for tall perforations buckling with LPB designation 2

$k_{f,3}$ . . . Plate buckling coefficient for tall perforations buckling with LPB designation 3

L_{xo} . . . Length of the perforated region, in the direction of the plate length

L_{yo} . . . Height of the perforated region, in the direction of the plate width

l_x . . . . Length of a perforation in the plate length direction

l_y . . . . Height of a perforation in the plate width direction

M_{Ed} . . . Design value of bending moment, i.e., the effects of actions calculated under design conditions [1]

N_0 . . . . Local elastic buckling capacity of an unperforated plate

$N_{cr,B}$ . . Nominal local elastic buckling capacity of a perforated plate, according to [2]

$N_{cr,d}$ . . . Distortional elastic buckling capacity

$N_{cr,e}$ . . . Global elastic buckling capacity

$N_{cr,ESB}$. Local elastic buckling capacity when the Euler Strip Buckling mode is identified

$N_{cr,FE}$ . . Local elastic buckling capacity of a perforated plate, found using finite element analysis

$N_{cr,h}$ . . . Local elastic buckling capacity of a perforated plate

$\widehat{N}_{cr,h}$ . . . Local elastic buckling capacity of a perforated plate, found using proposed formulae or methods

$N_{cr,INT}$. Local elastic buckling capacity when an INTeraction of buckling modes is identified

$N_{cr,LPB}$. Local elastic buckling capacity when the Lateral Perforation Buckling mode is identified

$N_{cr,l}$ . . . Local elastic buckling capacity

$N_{cr,l,k}$ . . Nominal local elastic buckling capacity of a perforated plate with modified buckling coefficient, according to [3]

$N_{cr,l,tr}$ . . Nominal local elastic buckling capacity of a perforated plate with reduced thickness, according to [3]

$N_{cr,S}$ . . . Nominal local elastic buckling capacity of a perforated plate, according to [4]

$N_{cr,SM}$ . . Nominal local elastic buckling capacity of a perforated plate, according to [3]

$N_{cr,USB}$. Local elastic buckling capacity when the Unstiffened Strip Buckling mode is identified

$N_{cr,WPB}$. Local elastic buckling capacity when the Whole Plate Buckling mode is identified

N_{Ed} . . . Design value of axial force [1]

$N_{u,d}$ . . . Distortional ultimate capacity

$N_{u,e}$ . . . Global ultimate capacity

$N_{u,exp}$ . . Local ultimate capacity determined experimentally

$N_{u,FE}$ . . Ultimate elastic buckling capacity of a perforated plate, found using finite element analysis

$N_{u,h}$ . . . Ultimate buckling capacity of a perforated plate

$\widehat{N}_{u,h}$ . . . Local ultimate buckling capacity of a perforated plate, found using proposed formulae or methods

$N_{u,l}$ . . . Local ultimate capacity

$N_{u,SM}$ . . Ultimate capacity of a plate, predicted by [3]

$N_{u,YR}$ . . Ultimate capacity of a plate, predicted by [5]

N_{Rd} . . . Design resistance to axial force, i.e., capacity of the member in axial compression [1]

$N_{w,Ed}$ . . Design value of axial force in the web of a member [6]

$N_{wp,Ed}$ . . Design value of axial force in the web-post of a member [6]

N_x . . . . The uniform compression force over the plate width, with a load axis parallel to the plate length

$N_{y,net}$ . . The net section yield capacity

n_c . . . . Number of perforations along the plate length (i.e., the number of columns)

n_r . . . . Number of perforations along the plate width (i.e., the number of rows)

r_p . . . . The perforation ratio i.e.,the gross area of perforations divided by the gross plate area

s_x . . . . Length between two adjacent perforations (measured parallel to the plate length)

$S_{xo,i}$ . . . Length between the loaded edge and the nearest perforation edge (measured parallel to the plate length); '$i = 1, 2$' for each end of the plate

s_y . . . . Width between two adjacent perforations (measured parallel to the plate width)

$S_{yo,i}$ . . . Width between the unloaded edge and the nearest perforation edge (measured parallel to the plate width); '$i = 1, 2$' for each side of the plate

t . . . . . Thickness of a plate

U_0 . . . . Internal strain energy of an unperforated plate

V_{Ed} . . . Design value of shear force [1]

$V_{wp,Ed}$ . . Design value of shear force in the web-post of a member [6]

W_0 . . . . External work done on an unperforated plate

w_0 . . . . Buckling modeshape of an unperforated plate

$\gamma_{M,1}$ . . . Partial material factor for resistance of members to instability [1]

ϵ . . . . . Strain

ϵ_u . . . . Ultimate strain

λ_c . . . . Global slenderness

λ_d . . . . Distortional slenderness

λ_l . . . . Local slenderness

$\lambda_{l,0}$ . . . Local slenderness of an unperforated plate

$\lambda_{l,h}$ . . . Local slenderness of a perforated plate

$\lambda_{l,m}$ . . . Modified local slenderness

ν . . . . . The Poisson's ratio

σ . . . . Stress (i.e., force per unit area)

σ_0 . . . . Critical elastic buckling stress of an unperforated plate

σ_{cr} . . . . Critical elastic buckling stress

Ω . . . . A variable used to characterise the local ultimate failure of perforated plates

1 Introduction

1.1 Background

Thin steel plates are employed in many structural forms; including bridge girders, webs and flanges of beams, columns and rack sections, or architectural cladding. In many instances, weight reduction or access for services and maintenance demand the inclusion of perforations within the member (Figure 1.1). Thin steel plates are susceptible to failure by buckling, which is a sudden and dangerous instability phenomenon which limits the geometric design of thin-walled plated elements subject to uniform compression.

At the instant of buckling, a structure deforms into a modeshape which mathematically satisfies the eigenvalue problem of non-zero deflection. A modeshape is characterised by buckling waves; the wavelength depends on the type of buckling, which in turn depends on the section geometry. Thin-walled sections typically exhibit global, distortional or local buckling, with elastic capacities of $N_{cr,e}$, $N_{cr,d}$ and $N_{cr,l}$ respectively (ordered by reducing buckling wavelength). Distortional and local buckling involve significant rotational distortion at the internal fold lines of a member, with the addition of translational distortion for distortional buckling. Conversely, global buckling does not involve distortion of the cross section. This thesis focuses on local buckling where the length of the buckling halfwaves depends on the cross-section dimensions, rather than the member length. Since its wavelength is small relative to the cross-section width, local buckling is difficult to

mitigate and therefore must always be considered [7]. Local elastic buckling is well-established in the domain of unperforated plates; the critical buckling mode depends only on the material properties (Young's Modulus, E, Poisson's ratio ν, plate flexural constant D), plate geometry (plate length a, plate width b, plate thickness t), loading and boundary conditions, which dictate the number of buckling half-waves over the plate length (m) [8–10].

On the one hand, stiffeners can be added to improve the buckling resistance, which increases material consumption and structural mass. On the other hand, perforated members subject to compression promise reduced section weight, with little compromise to buckling resistance. Yet, perforated plate buckling is complex to model computationally, and no closed-form analytical solution can account for multiple perforations of different shapes, sizes, orientations and spacings.



(a) Examples of uniformly perforated sections [11–14] (clockwise from top-left)



(b) Examples of less uniformly perforated sections [15, 16] (left-to-right)



(c) Members with large openings for service integration [17]

Figure 1.1: Structural members with perforations, adapted from [11–17]

The main methods used to determine the buckling capacity of perforated plates currently employed by design standards and output by research publications are depicted in Figure 1.2, including those which this thesis develops or investigates. All existing methods are evaluated in detail in Chapter 2, which summarises the major outcomes and observations from published work on the buckling of perforated plates. Although this includes a discussion on perforations with various shapes, this thesis will focus on members with multiple rectangular perforations (which are typical of many structures, Figure 1.1).

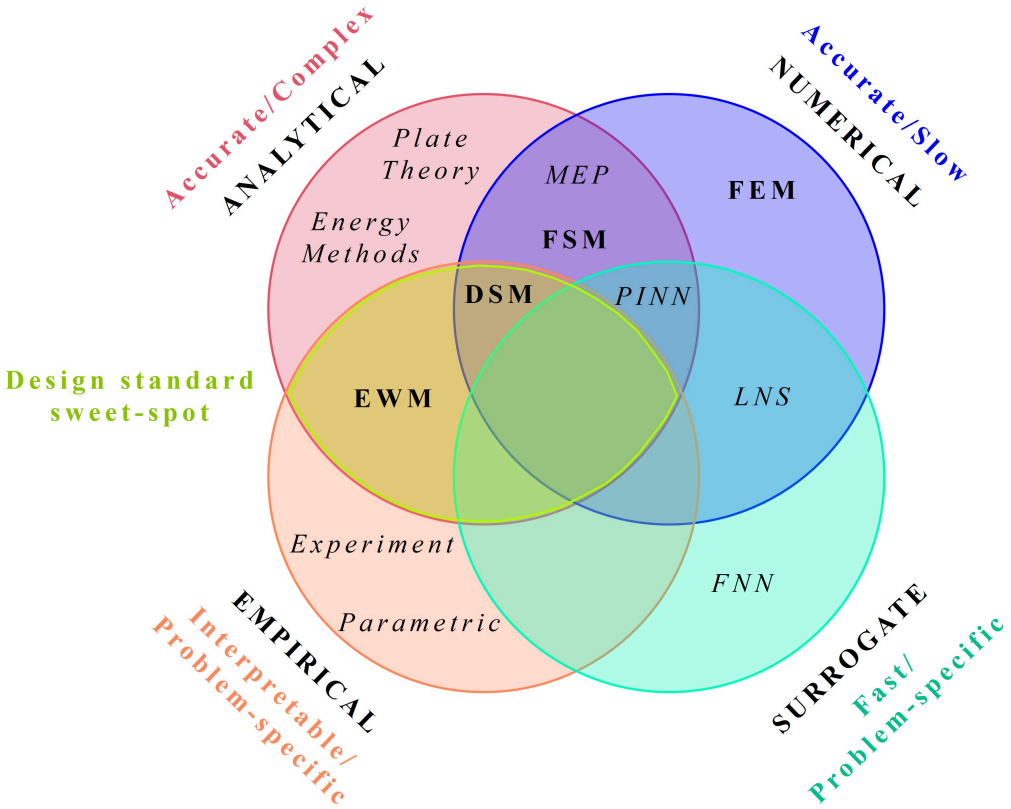


Figure 1.2: Methods for the design of thin, perforated structural members. Methods in **bold** are introduced in this chapter, whereas methods in *italics* are introduced or developed later

Most standards for the design of thin-walled sections employ the effective width method (EWM), which was formulated for codified use in the 1960s [7, 18, 19]. For example, EN 1993-1-5 [18] requires that either the EWM or the Reduced Stress Method (RSM) should be used to determine the buckling capacity, when the diameter of any unstiffened circular hole does not exceed $0.05b$ (where b is the width of the perforated plate) [18].

AISI S100-16 [7] also utilises the EWM to determine the postcritical buckling strength of uniformly compressed plates with circular or non-circular holes, updated to account for different perforation sizes, spacings and section depth [20, 21]. Additionally, [7] streamlines the limitations for applying the EWM with the Direct Strength Method (DSM). This method is not considered in pre-2016 North American Standard, or EN 1993-1-5 [18]. In fact, EN 1993-1-5 [18] has received no empirical, numerical or analytical update, despite more than 150 papers having been published on the buckling of perforated metallic plates or sections since the conception of the EWM.

Both [7] and [18] highlight that numerical methods typically provide superior approximations to the elastic buckling load of perforated members, which can also account for the beneficial restraint provided by any attached elements, and for details such as edge-stiffened holes. Numerical methods such as the finite element method (FEM), or finite strip method (FSM) are encouraged for all cases outside of the domain for which EWM and DSM equations apply [7, 18]. FSM is a semi-analytical method to determine the elastic buckling load of thin-walled sections, in which the cross-section is discretised into strips, and perforations are represented by a net cross section-weighted average thickness. The FSM is used by [7, 22] to supplement DSM for perforated sections to predict the local, distortional, and global buckling modes.

Although FEM and FSM can be significantly more accurate than the semi-analytical EWM and DSM, they are also much more computationally expensive. FSM has a limited ability to model holes (particularly when not arranged uniformly in rows and columns); and it can lead to artificially high predictions for $N_{cr,l}$. The FEM is user-dependent; sensitive to mesh density, element type, and boundary conditions, and its outputs are often difficult to cross-check and verify.

1.2 Thesis Overview

There are three main problems with the design and assessment of perforated plates subject to buckling, which this thesis will address. Firstly (i) design standards are driven by risk and safety; to permit engineers to apply safety factors and cross-check their calculations, design methods for the buckling capacity of perforated plates should be semi-empirical. Despite extensive research over the past half-century, developing methods that are safe, accurate, and broadly applicable remains a challenge. Secondly (ii) as perforated plate buckling is too complex to solve analytically, and often computationally expensive to solve numerically, engineers are limited in their ability to explore and adopt novel structural topologies. To enable more efficient and informed structural design, semi-analytical methods are required, which improve computational efficiency without compromising accuracy. In parallel with this, (iii) the optimisation of perforated plated members remains reliant on repetitive and time-consuming parametric FE studies. Semi-empirical optimisation, enabled through heuristic search processes, provides an intuitive solution by efficiently exploring large design spaces to identify favourable perforation arrangements.

In Chapters 4, 6 and 7 the first problem above is addressed. By first categorising the different local buckling modes with an interpretable design algorithm, Chapter 4 proposes a series of empirical equations to determine $N_{cr,l}$ for a perforated plate. This set of equations is compared to existing empirical methods, and a statistical safety factor enables standard-ready design [23]. Chapter 6 assesses the ultimate local buckling capacity $N_{u,l}$ for plates with an array of regular or staggered perforations. An empirical expression is developed to accommodate two distinct types of ultimate failure, and comparisons are again made to existing methods [24]. Chapter 7 systematically classifies and predicts buckling behaviour for uniformly compressed plates with tall perforations. This research is particularly relevant to EN 1993-1-13 [6], which provides design equations and protocol for beams with large web openings, omitting columns with large web openings under axial compression. Finding that typically $N_{u,l} < N_{cr,l}$ for such plates, the study in Chapter 7 evaluates how the buckling performance can consequently be maximised. Chapter 5 provides experimental validation to the conclusions made in Chapters 4-7, and makes

a significant contribution to the identification of local buckling modes for perforated plates in the absence of FE simulations [25].

Chapters 3 and 9 address the second problem above by establishing and evaluating different semi-analytical methods. Both chapters use the Rayleigh quotient; the method presented in Chapter 3 modifies it together with an approximate redistribution function [26], and the method presented in Chapter 9 employs it as the physics loss residual in a physics-informed neural network (PINN). The method presented in Chapter 3 performs comparably to empirical methods, and could become a faster alternative to FE with further development [26]. However, the method presented in Chapter 9 shows greater promise, within the domain on which the PINN is trained. As a mesh-free approach, the PINN avoids many of the challenges associated with FE spatial discretisation, mesh generation, distortion, and refinement. The result is a powerful tool that can compute an accurate prediction for $N_{cr,l}$ in a fraction of the time it takes for an FE simulation to be meshed and run. In addition, the PINN developed in Chapter 9 can determine an accurate visual representation of the elastic buckling modeshape for plates with eccentric perforations, unlike any published work.

Chapter 8 addresses the third problem above by implementing 'shape annealing' optimisation for rack section design. This rule-based procedure overcomes the complexities associated with numerical optimisation of perforated plates, while being particularly intuitive and user-friendly.

Finally, Chapter 10 concludes this work, outlines its implications for structural design, and summarises how the methods developed in this thesis address the three critical problems.

1.3 Important Concepts

1.3.1 Elastic Buckling Behaviour

For an unperforated plate, the critical elastic buckling load (N_0) is the capacity at which a plate becomes unstable when behaving 'perfectly' under small deflection theory [18]. Kirchhoff–Love thin-plate theory models thin plates by assuming that (i) normals to the mid-surface remain straight and normal to the deformed mid-surface and experience no extension, (ii) transverse shear strains are negligible, and (iii) the plate thickness is small compared with its in-plane dimensions. In its classical form, the theory is typically combined with the assumption of linear elastic material behaviour [27]. Under these assumptions, the three-dimensional elasticity problem reduces to a fourth-order differential equation of the transverse displacement modeshape $w(x, y)$ which governs bending and buckling of thin plates.

$$D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = q(x,y) + N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + 2N_{x,y}\frac{\partial^2 w}{\partial x \partial y} \tag{1.1}$$

Where D is the flexural rigidity, q is the transverse (z-direction) load and w is the z-direction displacement. The normal forces per unit length in the x, y and xy (shear) directions are N_x, N_y and N_{xy} respectively.

Timoshenko & Gere [8, 9] showed that a Fourier series of sinusoidal functions satisfies the boundary conditions for a simply-supported plate, displayed in Figure 1.3.

$$w(x,y) = \sum_m^\infty \sum_n^\infty \sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right) \tag{1.2}$$

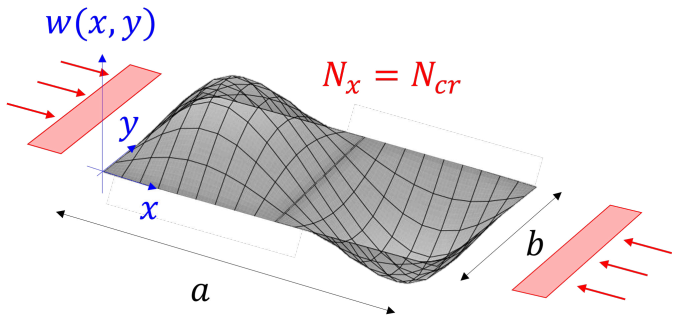


Figure 1.3: The modeshape of a simply supported buckling plate at the critical elastic buckling load $N_x = N_0$

For a simply-supported plate subjected to increasing x-directional compression, the pre-buckling response is characterised by a trivial equilibrium state in which the plate remains planar ($w = 0$). At a critical load N_0, this equilibrium loses stability, and the governing equations admit a non-trivial solution corresponding to out-of-plane deformation. Substituting Eq. 1.2 into Eq. 1.1 yields a linear bifurcation eigenvalue problem, from which the classical buckling load is obtained. The resulting solution demonstrates that elastic buckling is governed not only by the material properties E and ν, but also by the plate geometry through the aspect ratio a/b, geometrical slenderness b/t, and the associated buckling mode shape (where m is the number of halfwaves over the plate length and n is the number of halfwaves over the plate width).

$$N_0 = bt\sigma_0, \tag{1.3a}$$

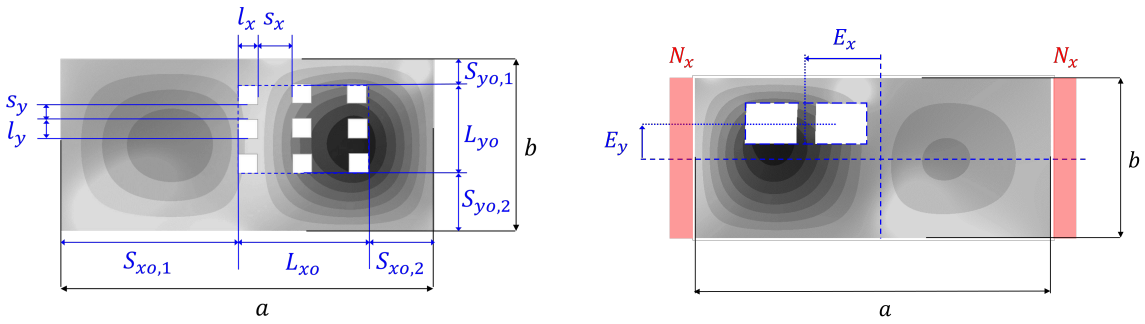
$$\sigma_0 = k\frac{\pi^2 E}{12(1-\nu^2)}\left(\frac{t}{b}\right)^2, \tag{1.3b}$$

$$k = \left(\frac{mb}{a} + \frac{a}{mb}\right)^2, \tag{1.3c}$$

where the critical elastic buckling stress is σ_0. However, as discussed in Section 1.1, the elastic buckling load of a perforated plate (termed $N_{cr,h}$ herein) cannot be determined in closed form.

1.3.2 Perforation Geometry

This thesis considers a 'regular', 'staggered' or 'irregular' arrangement of rectangular perforations. This arrangement is characterised by how holes perforate a 'region', defined with dimensions L_{xo} and L_{yo} (Figure 1.4(a)), and perforations are arranged in n_r rows and n_c columns at some average eccentricity about (i.e., offset from) the plate centroid (E_x, E_y, Figure 1.4(b)). The ratio of these eccentricities relative to the plate dimensions are given as $e_x = E_x/a$ and $e_y = E_y/b$. Distances from this region to the edges are denoted $S_{xo,i}$ and $S_{yo,i}$ (Figure 1.4(a)), with S_{xo} and S_{yo} being the smaller distance. Figure 1.4(a) also defines the perforation spacing (s_x, s_y) and size (l_x, l_y). The perforation ratio r_p is the ratio of the total perforation area to the gross (unperforated) plate area. Unless otherwise stated, this thesis considers loading N_x to be applied uniformly over the plate width b, as shown in Figure 1.4(b).



(a) Regular variables examined

(b) Eccentricity with uniform compression over the plate width, shown in red (N_x)

Figure 1.4: Perforation geometry defined in this thesis

1.3.3 Slenderness and Post-Critical Behaviour

Following elastic buckling, a plate may continue to sustain additional load through post-critical behaviour. In this regime, the plate has buckled out-of-plane, but retains load-carrying capacity. The maximum load attained is termed the ultimate capacity (defined in this thesis as $N_{u,h}$ for perforated plates).

According to [18], slender plates with a relative slenderness $\lambda_P > 1.22$ possess a post-critical reserve of strength, allowing them to resist loads beyond the elastic buckling load ($N_{cr,l}$) and reach an ultimate capacity ($N_{u,l}$, where $N_{u,l} > N_{cr,l}$). Conversely, stocky plates with a relative slenderness $\lambda_P \leq 0.673$ are fully effective and possess little or no post-buckling reserve, such that yielding occurs before significant buckling develops and $N_{u,l} < N_{cr,l}$.

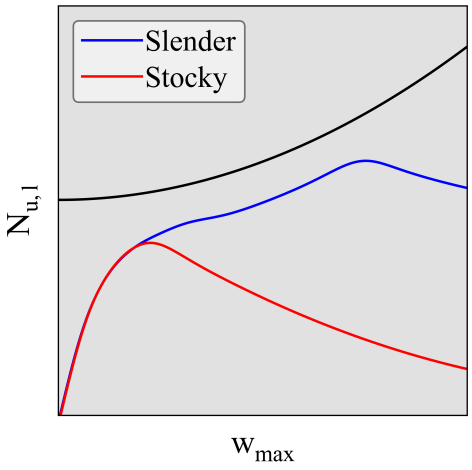


Figure 1.5: Ultimate load-displacement curves for slender and stocky plates. The 'perfect' elastic plate behaviour is shown in black.

The strength reserve of slender plates arises from a redistribution of in-plane membrane stresses following buckling. As deformation increases, stresses concentrate within effective load-carrying regions of the plate, while other regions unload. The ultimate load is reached when this redistribution causes yielding within parts of the plate. Since yielding is typically localised, failure is generally governed by an interaction between buckling and plasticity. In contrast, stocky plates reach yield before the elastic buckling load is attained, and therefore collapse primarily through plasticity rather than buckling (Figure 1.5).

The post-buckling stress redistribution observed in slender plates formed the basis of the effective-width method (EWM) developed by von Kármán and Winter [28]. The EWM approximates the non-uniform stress distribution in a buckled plate by replacing the full plate width with an equivalent effective width, b_{eff}, that carries a uniform compressive stress. The effective width is calculated using empirical expressions, of which Eqs. 5.4 are a commonly adopted form. The EWM forms the basis of local buckling design provisions in many structural design standards [7, 18, 19], and is discussed further in Chapter 2.

$$b_{eff} = \rho b, \tag{1.4a}$$

$$\rho = \sqrt{\frac{1}{\lambda_p}}\left(1 - 0.22\sqrt{\frac{1}{\lambda_p}}\right) \tag{1.4b}$$

1.4 Statement of Authorship

This is an 'integrated thesis' thus Chapters 3-9 represent a stand-alone series of papers, of which Chapter 3, 4, 5 and 6 are published [23–26]. For efficiency and to avoid repetition, the literature review from each of these papers is consolidated into Chapter 2, and the abstract of each paper is integrated into the relevant commentary section.

All papers are co-authored by Prof. Barbara Rossi and myself, Chapter 3 with the addition of Prof. Zhong You, and Chapter 8 with the addition of Dr. Ioannis Mirtsopoulos at ETH Zurich. I, Elenor Naraidoo (EN), am responsible for the core ideas, conceptualisation, design, data collection, interpretation of results, and techniques employed in each paper or chapter. I am the first author in the listed published work, and independently wrote the first draft of each. As my primary supervisor, Barbara Rossi (BR) provided guidance and direction on my technical work, and provided feedback and proposed modifications in subsequent drafts. The contribution made by the additional authors Zhong You (ZY) and Ioannis Mirtsopoulos (IM) is outlined below.

- Chapter 3: Elastic Buckling of Slender Steel Plates with Perforations. Authored by EN, BR, ZY. Published in the Proceedings of the Nordic Steel Conference, Luleå, 2024 [26]. ZY provided guidance on the mathematical theory underpinning this work. Please note that all graphs were re-formatted according to the common style of graphs in this thesis.

- Chapter 4: Local Buckling Load of a Perforated Plate - A Computational Study. Authored by EN, BR. Published in Thin-Walled Structures, 2025 [23].
- Chapter 5: An Experimental, Numerical and Analytical Study of Local Buckling in Perforated Plates. Authored by EN, BR. Published in Thin-Walled Structures, 2026 [25].
- Chapter 6: The Direct Strength Method for Plates with Perforations. Authored by EN, BR. Published in the Journal of Structural Engineering, 2026 [24].
- Chapter 7: The Prediction and Classification of Local Buckling for Plates with Tall Perforations. Authored by EN, BR. Submitted to Engineering Structures.
- Chapter 8: Shape Annealing Optimisation of Perforations in Rack Sections Subject to Buckling. Authored by EN, BR, IM. Preparing for submission in Advanced Engineering Informatics.

  IM took charge in translating the python script I independently authored into C#. This was necessary to design Abalone [29, 30](Appendix I), a shape-annealing optimisation plug-in for Grasshopper [31]. This intellectual property is protected by copyright with the University of Oxford (and thus no code is provided in the Appendix). IM provided no input on the manuscript preparation, and very minor input on the investigation undertaken in Chapter 8.
- Chapter 9: A Physics Informed Neural Network for the Elastic Buckling Capacity of a Perforated Plate Authored by EN, BR. Preparing for submission in Computer Methods in Applied Mechanics and Engineering.

As this is an integrated thesis, each chapter may appear more 'self contained' than those in a traditional thesis. For example, the same figure may appear more than once, and certain sub-chapters are similar (e.g. those describing FE studies). I have attempted to improve the coherence of this thesis by cross-referencing the relevant chapter in papers where I cite my published work, and by ensuring that all abbreviations and notation are consistent between chapters. Furthermore, the commentary section at the start of each

chapter aims to contextualise and link each chapter together. Please be aware that the order by which the papers were published dictated the order of the chapters. Finally, please note that extracts from my published work [23–26] are used to compose parts of this chapter, and the chapter which follows.

2
Literature Review

2.1 Methods for the Prediction of Buckling Capacity

2.1.1 Parametric Numerical Studies

Many authors parametrically assess the buckling load of perforated sections with commercial FE software. Some authors use these results to present design charts within a given domain [32, 33], or discuss observations qualitatively without providing an expression for the critical elastic buckling load of a perforated plate, based on the parameters they each assess [34–40]. However most authors undertake parametric assessments to derive empirical relations, or suggest modifications to existing methods. In [32], the authors determine design charts and formulae for the critical buckling capacity of trapezoidal sheeting perforated with a regular triangular or quadratic array of circular perforations by undertaking a parametric FE analysis. Shanmugam, Thevendran & Tan (1999) [33] propose design formulae and charts for the ultimate capacity of thin square plates with a single concentric square or circular perforation subject to different boundary and loading conditions. Adany (2017) [41] investigates the application of the constrained finite element method (cFEM) for the buckling analysis of thin-walled members with holes. The author numerically investigates the influence of holes of various shapes and sizes on the global, distortional and local buckling modes. Modal decomposition with cFEM can determine hole-induced changes, where otherwise the effect of holes on the buckling behaviour can

make it "all but impossible to reasonably assess the change of the critical load due to holes" [41]. Guo & Yao (2021) [42, 43] undertake both FE and physical experiments on sections with a single concentric circular or rectangular perforation, to derive closed-form expressions for local buckling according to the perforation size by undertaking a regression analysis. The authors use their findings to propose modifications to effective width formulae.

Scheperboer, et al. (2016) [4] examine the buckling of square plates with staggered circular perforations, and propose that elastic buckling load is a quadratic function of hole area and unperforated plate buckling load. In their examination, they find that for the same total perforation area, plates with more holes have lower a buckling load. Thus, they use parametric models with the most perforations to derive the equation's coefficients with the least square error method:

$$\frac{N_{cr,S}}{N_0} = 1.8(r_p)^2 - 2.1r_p + 1 \tag{2.1}$$

In their expression, the critical elastic buckling load of a perforated plate ($N_{cr,h} = N_{cr,S}$) is a function of the critical elastic buckling load of an equivalent unperforated plate (N_0), and the ratio of the area of holes to the area of the unperforated plate (i.e., the perforation ratio r_p, see Section 1.3.2). As it expresses critical elastic buckling behaviour so effectively, Equation 2.1 is compared to many of the methods developed throughout this thesis.

2.1.2 Semi-Analytical Formulations

Semi-analytical predictions are achieved by considering a simplified mathematical problem, in conjunction to FE analysis. Paik & Thayamballi (2018) [44] examine how the Johnson-Ostenfeld equations can be used to determine the critical perforated plate buckling capacity. The Johnson-Ostenfeld formulation is adapted from Johnson's guide [45] to determine the critical buckling capacity of any (stocky or slender) column. If the critical elastic buckling load is less than half of the yield strength, the buckling capacity is limited by Euler's hyperbola. Otherwise, a modification is made to Euler's buckling strength prediction, for elasto-plastic failure. Although developed for columns, [44] consider these formulations to predict the critical buckling capacity of both perforated and unperforated plates, with the elastic buckling strength of a plate replacing the Euler's column capacity. The authors

identify that the Johnson-Ostenfeld formulation greatly underpredicts the elastic buckling load for plates with a large perforation. Furthermore, the authors find that the critical buckling capacity of plates with small perforations is significantly underestimated [44]. Recommending not to use the Johnson-Ostenfeld formulation for the critical buckling prediction of a plate which undergoes elasto-plastic failure, the authors present formulae in terms of the ultimate load capacity [44]. These equations are applicable only to rectangular plates with a single concentric perforation, or multiple perforations spaced very far apart.

Moen & Schafer (2009) [2] established a prediction for the buckling of a rectangular plate with a large aspect ratio, and a single row of rectangular holes with a corner radius. They define two possible buckling modes; plate buckling away from the hole, or plate buckling at the unstiffened strip above or below the hole (or an interaction of the two) [2]. They consider buckling of the former type to be equal to the unperforated plate buckling load N_0, and buckling of the latter type ($N_{cr,h} = N_{cr,B}$) as follows (with variables described in Figure 2.1).

$$N_{cr,B} = N_{cr,u,net}\left(1 - \frac{l_y}{b}\right), \tag{2.2a}$$

$$N_{cr,u,net} = \min(N_{cr,1}\,,\, N_{cr,2}), \tag{2.2b}$$

$$N_{cri} = k_i \cdot \frac{\pi^2 E}{12(1-\nu^2)} \cdot \left(\frac{t}{S_{yo,i}}\right)^2, \tag{2.2c}$$

$$k_i = \begin{cases} 0.425 + \dfrac{0.2}{\left(\frac{l_x}{S_{yo,i}}\right)^{0.95} - 0.6} & \text{if } \frac{l_x}{S_{yo,i}} \geq 1, \\ 0.925 & \text{otherwise} \end{cases} \tag{2.2d}$$

where $i = 1$ or 2, l_y is the perforation height, l_x is the perforation length, E is the Young's modulus, ν is the Poisson's ratio, and $S_{yo,i}$ is the spacing between the perforation and the unloaded edge (Figure 2.1). Equation 2.2 is compared to the methods developed in later chapters.

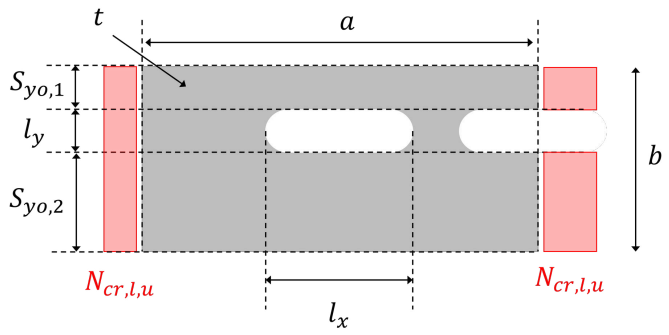


Figure 2.1: Variables considered by [2]

In comparison to FE results for simply supported plates, [2] find that their prediction is both accurate and conservative when failure manifests as unstiffened strip buckling (Section 2.2.4), however the capacity may be underpredicted by over 50% for a line of perforations transversely offset from the longitudinal axis, subject to uniaxial compression [2]. Therefore, the authors impose that $e_y b \leq 0.15 l_y$.

Minera, et al. (2021) [46] establish the Localised Nominal Stiffness method (LNS) to determine the sensitivity of a localised region of a structure to stiffness variation (i.e., the introduction of a hole). The result is a sensitivity contour map which informs the user where best to locate a small number of small holes.

Smith & Moen (2014) [3] developed a semi-numerical approach to determine the local buckling capacity $N_{cr,h}$ of plates with multiple rows and columns of rectangular perforations:

$$N_{cr,h} = \min(N_{cr,l,k}, N_{cr,l,tr}) \tag{2.3}$$

Where $N_{cr,l,tr}$ is approximated with the Finite Strip Method (FSM) using a reduced cross-sectional thickness t_r, expressed in terms of the Poisson's ratio (ν), plate length (a) and width (b), number of perforations along the member length (n_c) and member width (n_r), perforation length (l_x) and width (l_y) (see Figure 1.4), and perforation constants α and β. These constants depend on the number of buckling halfwaves across the plate length (m) and plate width (n) respectively. This reduced thickness is given as [3]:

$$t_r = t\left(1 - \frac{n_c n_r (l_x l_y + \nu l_y \alpha + \nu l_x \beta + \alpha\beta)}{ab}\right)^{0.5}, \tag{2.4a}$$

$$\alpha = \frac{a}{m\pi}\sin\left(\frac{m l_x \pi}{a}\right), \tag{2.4b}$$

$$\beta = \frac{b}{n\pi}\sin\left(\frac{n l_y \pi}{b}\right) \tag{2.4c}$$

The local buckling capacity $N_{cr,l,k}$ is a modification of the unperforated plate capacity by some buckling coefficient k_h [3]:

$$N_{cr,l,k} = \frac{k_h \pi^2 E}{12(1-\nu^2)} \cdot \left(\frac{t^3}{b^2}\right), \tag{2.5a}$$

$$k_h = 4\left(1 - \frac{n_c n_r (l_x l_y - \nu l_y \alpha X - \nu l_x \beta Y + \alpha X \beta Y)}{ab}\right), \tag{2.5b}$$

$$X = \begin{cases} -1 & \text{if } \frac{n_c b}{a} = \frac{1}{1}, \frac{1}{3}, \frac{1}{5} \\ 0 & \text{otherwise} \end{cases} \tag{2.5c}$$

$$Y = \begin{cases} -1 & \text{if } n_r = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.5d}$$

Although this semi-numerical aspect adds complexity, this approach is more applicable to plates with multiple rows of perforations than expressions discussed previously. However, it cannot be used for a staggered arrangement of perforations [3].

In [47–52] the authors used the isoparametric spline finite strip method (ISFSM) to predict the elastic buckling of uniformly compressed plates with tall perforations of various shapes. Traditional FSM assumes a sinusoidal function over the length of the member, and represents perforations by a reduced thickness term. For reasons discussed, this limits the accuracy of FSM for perforated plates. However, ISFSM replaces the prescribed sinusoidal longitudinal functions with spline interpolation functions, and uses isoparametric geometry mapping to represent the perforated geometry more accurately. Although the study domain is small, [47, 48] found that ISFSM can accurately capture multiple modes of buckling, for less computational effort than FE. Later, accurate buckling predictions were again found for the ultimate capacity, when the method was extended to include both material inelasticity and geometric nonlinearity [49–52]. Yet despite these reports of high accuracy, ISFSM has not been widely investigated and adopted. Classical FSM is appealing as it is simple, fast,

and well-embedded in existing workflows; while FE is appealing because it is versatile, generalisable and has very mature commercial implementation. ISFSM is both not simple enough, and not powerful enough when compared to each of these established methods.

2.1.3 Energy-Based Methods

This section is extracted from [26], thus pertains particularly to Chapter 3, where a modified energy prediction is developed. The Rayleigh quotient, Rayleigh-Ritz energy solution and virtual work are energy methods, employed by several authors for the problem of plate buckling [3, 53–55]. Energy methods involve approximating the displacement function as a sum of functions satisfying kinematic boundary conditions, on the basis of the Lagrange-Dirichilet theorem [45]. Most of the methods discussed until now necessitate some simplification of plate buckling, expressing design equations in terms of a single or minority of the parameters involved, and consequently sacrificing some accuracy. However, energy methods can be applied to a larger domain. Their accuracy depends on good knowledge of the displacement function, finding an upper-bound solution. Authors disagree as to the correct application of these approaches for perforated plates. For example, [53] account for the 'highly nonuniform' force redistribution within a plate with a single concentric hole in the calculation of the external work, whereas [54] demonstrate mathematically that such force inhomogeneity does not directly affect the calculation. In similar disagreement, [3] conservatively consider the gross plate area (i.e., an unperforated plate) in their calculation for the external work done. Yet, all authors consider the net plate region for the calculation of the internal energy [3, 53–55]. Each of these authors acknowledge that the modeshape of the buckled perforated plate will be different to that of the unperforated plate, and must be approximated in order to determine the buckling capacity accurately. Osgouei, et al. (2020) [55] compare the accuracy of a range of kinematically admissible displacement functions to determine which can find the most accurate buckling capacity prediction. Djeloseciv, et al. (2013)[53], Gracia & Rammerstorfer (2019) [54], and Smith & Moen (2014)[3] represent the displacement function in terms of 'amplitude factors' which they determine by conducting a Rayleigh-Ritz assessment. Gracia & Rammerstorfer (2019) [54] only guarantee high accuracy with a semi-analytical assessment, relying on

FE analysis to determine a multivariable polynomial approximating the modeshape of the critical eigenvector. The authors develop a sensitivity contour map according to the vulnerability of a localised region to strain energy variation, similar to [46]. Their map shows areas where the introduction of a small perforation can increase the elastic buckling load of a square plate [54].

2.1.4 Physics-Informed Neural Networks

A Physics-Informed Neural Network (PINN) is a neural network trained to solve supervised learning tasks while respecting any given physical law, described by general non-linear partial differential equations (PDEs) [56]. They are a 'mesh-free' technique that finds solutions to PDEs without directly solving the governing equations, using a neural network's function approximation property [57]. This is achieved by enforcing a physics-informed loss term, which acts to penalise unacceptable solutions. After training, the PINN can produce an estimated high-fidelity solution for a given point in the integration domain (i.e., at 'collocation' points). Raissi, et al. (2019)[58] introduced a unified PINN framework for (i)'forward' and (ii)'inverse' problems, where the network solves a PDE (i) given parameters and (ii) without parameters, which are learned from data.

PINNs hold great potential for structural engineering problems, which are typically described by PDEs that can be too complicated to be solved analytically. Yet, as PINNs are less than a decade young, very few authors employ PINNs to assess structural behaviour, and their application to buckling structures was only developed in the past few years.

Kapoor, et al. (2024) [59] and Fuh-Gwo, et al. (2020) [60] develop a PINN framework for structural beam systems using classical Euler-Bernoulli beam theory. Kapoor, et al. (2024) [59] also use Timoshenko beam theory to develop nondimensionalised PDEs, which are used as residual terms in the PINN loss. The authors develop a PINN which can accurately predict out-of-plane displacements and cross-sectional rotations, and can infer unknown model parameters even with noisy data [59].

Fuh-Gwo, et al. (2020) [60] develop a multi-layer feed-forward artificial neural network to solve the beam vibration problem for both forward and inverse problems. The authors

demonstrate that a PINN can overcome overfitting issues associated with lack of training data, specifically for applications in structural health monitoring (SHM).

Al-Adly & Kripakaran (2024) [61] and Bischof & Kraus (2025) [62] use the Kirchhoff-Love differential plate equation as the governing PDE loss for a PINN used to measure the deflection of a slab or plate. With SHM intentions, [61] examine different uniformly distributed loads (UDL) and boundary conditions, finding that their PINN can predict deflections and moments with a coefficient of variation error of 20% and 13% respectively, which they state indicate strong potential for generalisation, and demonstrate superior predictive capabilities outside the range of training data [61].

Bastek & Kochmann (2023) [63] develop a PINN to solve the small-strain response of arbitrarily shaped shell structures using Naghdi shell theory. The authors find that the strong form of these equations (i.e., the PDE residual at collocation points) leads to poorer convergence than the weak form loss (i.e., the variational PDE residual - the total potential energy functional) [63].

Guo, et al. (2025) [64] Liu, et al. (2024) [65] and Liu, et al. (2025) [66] are some of the first to develop a PINN for the problem of buckling structures. Motivated by the high computational cost of numerical approaches, and their requirement for specialised expertise for model setup and result interpretation, [64] propose a PINN for predicting the compressive buckling failure of laser-welded stiffened plates. In comparison to FE analysis, the PINN more accurately predicts the failure load of experimental test results [64].

Liu, et al (2024, 2025) [65, 66] investigate different strategies to incorporate physics-informed features into an artificial neural network to predict the buckling capacity of cylindrical shells. Using experimental data to train and test their PINN, the authors find that it outperforms purely data-driven models [65, 66].

Rather than enforcing a strong-form PDE loss, [63, 67, 68], introduce the weak-form loss which requires fewer residual terms and avoids higher-order differential calculations than the strong-form loss. Wang & Thai (2025)[69] use the variational energy (weak-form) loss to develop a PINN that can accurately predict mid-plane displacements, stress and moment distributions of laminated composite plates under bending, for different ply-directions and boundary conditions [69].

For eigenvalue problems, [68, 70] reformulate the variational principle for the smallest eigenvalue as the Rayleigh quotient (Section 2.1.3). Rowan, et al. (2025)[70] demonstrate that the weak-form loss simultaneously accelerates computation using automatic differentiation, and expands the class of admissible solutions by reducing continuity requirements. Accompanied with Gram-Schmidt orthogonalisation, [70] demonstrate that the Rayleigh quotient can efficiently find multiple eigenvalues and eigenfunctions for domains with various topologies.

Finally, Nabian, Gladstone and Meidani (2021)[71] develop an algorithm for piecewise constant approximation of loss on a sample domain, where loss is evaluated at groups of collocation points in cells. The authors investigate the performance of a PINN for a linear elasticity problem of a 2D plate with multiple holes, using plane stress equations. The authors find that their loss or error-aware sampling approaches improve the efficiency of PINN training and accelerate convergence [71].

2.1.5 Overview

The previous sections detailed how authors have implemented the four broad methods depicted in Figure 1.2 (numerical, analytical, empirical, or surrogate). The advantages and disadvantages of particularly relevant methods are summarised in Table 2.1.

Table 2.1: Advantages and disadvantages of various published methods

Author	Method	Summary	Perforation Geometry		Pros.	Cons.
			Arrangement	Shape		
ref.	*Empirical (E), Analytical (A), Numerical (N), Surrogate (S)*	*Summary of method*	*Single, concentric (S), Multiple, one row ($n_r = 1$), Multiple, multiple rows ($n_r > 1$), Multiple, staggered array (M), Multiple, irregular array (I), Multiple, any arrangement (A), Not applicable (N/A)*	*Square (S), Circular (C), Rectangle with $l_x/l_y > 1$ (RL), Rectangle with $l_x/l_y \leq 1$ (RH), Any (A), Not applicable (N/A)*	*Pros*	*Cons*

Author	Method	Summary	Perf. Geom.		Pros.	Cons.
[2]	A	Determine two buckling failure modes. Balance a uniform distribution of forces in the plate plane to determine the net elastic buckling load.	$n_r = 1$	RL	Accurate for a concentric row of small or large holes with $l_x/l_y > 1$.	Significantly unconservative when the row of perforations is eccentric.
[3]	A/N	Use the Rayleigh-Ritz energy solution to propose a reduced thickness for FSM, giving $P_{cr,l,tr}$. Assume that the external work equals that of an unperforated plate.	$n_r > 1$	RL	When compared to the elastic buckling load of 54 rack section geometries determined with FE, the formulation exhibits good accuracy with little variance.	Overly conservative as the net web area reduces. Unconservative for rack sections with a large net web area. Cannot be used for rows of staggered perforations.
[4]	E	Conduct a parametric assessment to determine a quadratic expression for $N_{cr,h}$ in terms of only r_p and N_0.	M	C	Simple standard-ready and material-independent.	No validation dataset. Narrow domain of investigation.

Continued on next page

Author	Method	Summary	Perf.	Geom.	Pros.	Cons.
[44]	A/E	Modify Johnson-Ostenfeld equations for perforated plates.	S	C	None; assert that the Johnson-Ostenfeld formulation inadequately predicts $N_{cr,h}$.	Significantly unconservative for slender plates with a large perforation; greatly overly conservative for stocky plates with a small perforations.
[46]	S/N	Introduce the LNS method, resulting in a map of how $N_{cr,h}$ is sensitive to each region in a plate. Insensitive regions are used to direct perforation location.	A	A	Can determine perforation designs that are superior to naïve designs with the same perforation area.	Computationally expensive and time-consuming. Sensitivity map does not capture force redistribution benefits for plates with tall perforations.
[53]	A	Plate is composed of sections, acted on by non-uniform forces. The total internal strain energy is the sum of the strain energy of the sections of plate. The potential of all sections of the plate is minimised to determine $N_{cr,h}$.	S	C S RL RH	Strong agreement with FE solution.	Very complex, inapplicable outside of study domain.
[54]	A/N	Develop a buckling sensitivity map by minimising the Rayleigh quotient, and approximate the buckling modeshape with a Ritz-ansatz. Show that inhomogeneous force redistribution does not affect the external work calculation. Internal and external energies are the difference between the internal and external energies of the unperforated plate and the hole region.	A	A	Fully analytical for plates with small holes, and results agree with FE solutions.	Semi-analytical when the modeshape greatly differs to the unperforated plate (e.g. large holes). A smooth Ritz-ansatz approximation to the modeshape must be found with multivariable regression.

Continued on next page

Author	Method	Summary	Perf.	Geom.	Pros.	Cons.
[55]	A/E	Internal strain energy of members with stiffened openings is the sum of modeshape deformation energy of the plate and stiffener. This modeshape is approximated with a polynomial formula. This is equated to the work done on the plate and stiffener.	S	RL	Buckling coefficient is a function of only geometry and material properties.	Very large error and variation when validated on 34 FE models.
[47]	A/N	Use ISFSM to determine the elastic buckling load of near-square plates with two tall perforations.	$n_r = 1$	A	High accuracy for comparably low computational overhead to FE	Cannot contend with simplicity of FSM and generality of FE, both of which are well-established methods.
[61]	S	Train a physics-informed neural network (PINN) with the Kirchhoff-Love differential plate equation as the physics loss.	N/A	N/A	Provides a framework for implementing Kirchhoff-Love plate theory in a PINN.	Narrow study; the PINN is not generalisable. High percentage error in comparison to FE. Do not investigate plate buckling.
[62]	S	Showcase the Relative Loss Balancing with Random Lookbacks (ReLoBRaLo) self-adaptive loss balancing scheme for PINNs on the Kirchhoff-Love plate bending equations.	N/A	N/A	Provides a framework for using ReLoBRaLo, and demonstrates that it can accurately solve the Kirchhoff-Love plate bending equations for a simply-supported square plate.	Only consider a uniformly distributed load (not buckling). PINN is therefore not generalisable for different inputs or case studies.
[64]	S	Train a PINN to predict $N_{cr,h}$ of stiffened plate structures, validated by experimental tests.	N/A	N/A	Generalisable across multiple inputs (e.g. different panel dimensions, spacing, material properties etc). Very high accuracy on a test set.	The PINN considers Euler-Bernoulli beam theory, a much simpler problem than Kirchhoff-Love plate bending theory.

Continued on next page

Author	Method	Summary	Perf.	Geom.	Pros.	Cons.
[65]	S	Train a PINN to determine $N_{cr,h}$ of axially compressed cylindrical shells.	N/A	N/A	The PINN is generalisable across multiple inputs, and superior when compared to a naïve feed-forward neural network.	Large variation in accuracy when compared to the experimental test set results.
[66]	S	Train a PINN to determine $N_{cr,h}$ of spherical shells under pressure.	N/A	N/A	The PINN is generalisable across multiple inputs.	Large variation in accuracy for both the testing and training datasets. When a safety factor is imposed, the maximum absolute error is 40%.
[70]	S	Demonstrate that the Rayleigh-quotient with a Gram-Schmidt orthogonalisation procedure can robustly and accurately solve eigenvalue problems when used as the physics residual in a PINN.	S	C	The PINN can be used on domains with complex geometries, including a 'donut' (perforated) case. For the perforated case, there is indistinguishable difference when compared to the FE result.	The PINN is retrained per-case (not generalisable across multiple inputs), and not demonstrated on test cases.
[71]	S	Showcase 'performance sampling' in a PINN for the plane stress elasticity problem of a perforated plate. The PINN can accommodate stress boundary conditions.	I	C	Accurately predicts the stress field and displacement when compared to FE results.	The PINN is retrained per-case (not generalisable across multiple inputs), and not demonstrated on test cases.

2.2 Observed Buckling Behaviour of Perforated Plates

The influence a single perforation has on $N_{cr,h}$ is summarised from [37, 53, 72] in Figure 2.2, explained in detail in the following sections. Note that the influence of multiple perforations on $N_{cr,h}$ is not represented diagrammatically here; it is more complex and under-researched, discussed in the following sections.

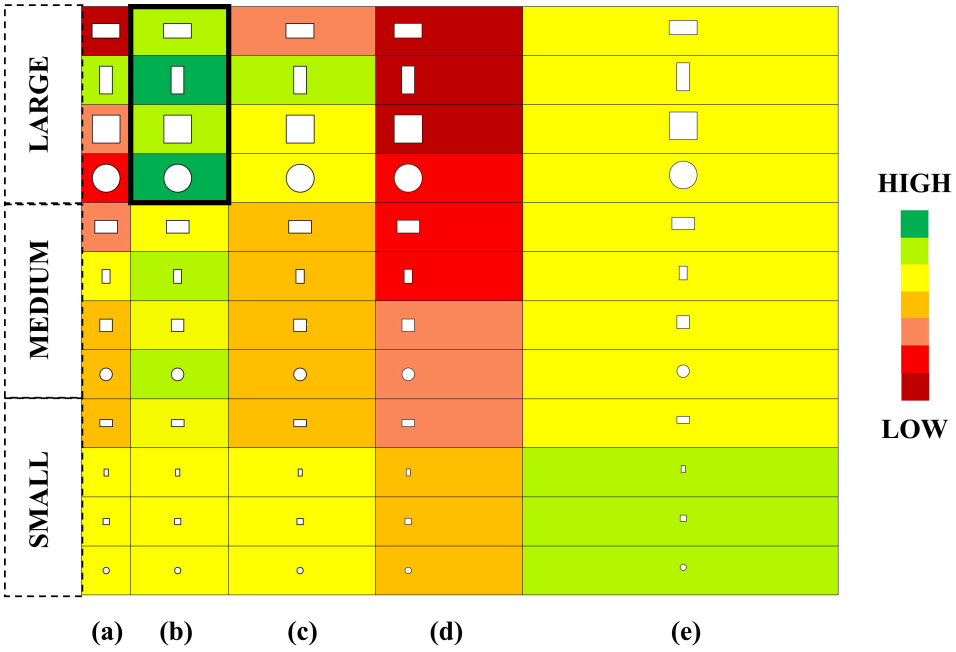


Figure 2.2: The influence of perforation shape and location on $N_{cr,h}$ (high/low) for (a) a concentric perforation in $a/b = 1$, (b) a concentric perforation in $a/b = 2$, (c) a concentric perforation in $a/b = 3$, (d) a perforation in the 'end panel', (e) a concentric perforation as $a/b \to \infty$. The perforated plates highlighted with the bold box exhibit 'wavelength stiffening'

2.2.1 Perforation Size and Shape

Vann (1971) [73] and Sweedan & El-Sawy (2011) [35], determined that the influence of a hole on the elastic buckling load of a simply supported square plate is twofold. The first influence is a reduction to the plate bending stiffness in the portion of the plate where the hole is located; the second influence is a redistribution of stresses away from the centre of the plate and towards the supported edges [35, 73]. The former effect dominates for small holes, typically causing buckling load to reduce; the latter effect dominates for large holes (with a diameter beyond 40-50% of the plate width), causing the buckling stress to increase relative to the unperforated case [35, 73, 74]. Note that this is not

necessarily indicative of an increased buckling capacity, which also depends on the plate axial stiffness [2], and on the location of the perforation.

Although perforation size dictates $N_{cr,h}$, [53] showed that its shape, position, and orientation are also influential. In addition to size, a rectangular perforation located at a buckling node has a greater redistribution effect [38, 72, 75]. Furthermore, a rectangular perforation oriented with its longer length parallel to the plate width (b) enables a greater $N_{cr,h}$ than one oriented orthogonally [37, 53, 72]. For a rectangular perforation located at the centre of a square plate, $N_{cr,h}$ is proportional to l_y/l_x [37].

However, $N_{cr,h}$ reduces as l_y increases for a circular or square perforation located at the centre of a square plate [38, 72, 74]. $N_{cr,h}$ reduces at a greater rate for plates with square holes, attributed to the difference in stress concentration (opposed to the difference in area) of the two perforation shapes [74].

More recently, [39] showed that both l_y/l_x and the hole shape influence stress distribution at failure, and dictate the elastic buckling load (Figure 2.3). The authors examined concentric rectangular, elliptical, hexagonal and diamond holes, finding that that the optimal l_y/l_x for one shape is different to the next; this governs whether the plate buckles elastically (l_y/l_x is small) or elasto-plastically (l_y/l_x is large).

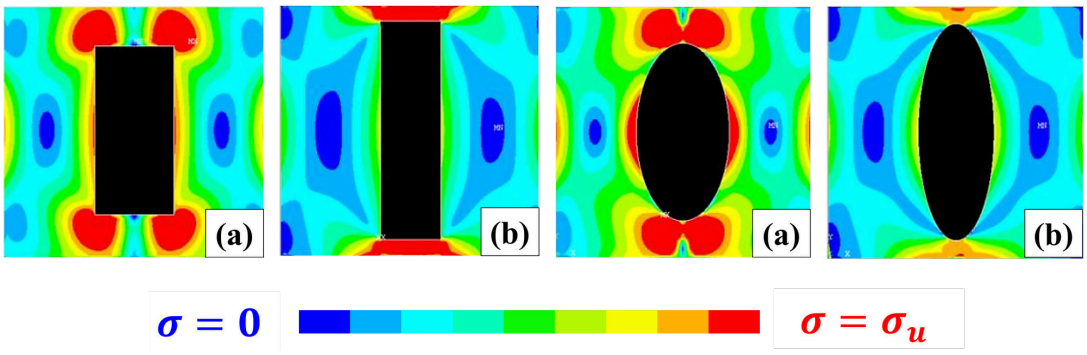


Figure 2.3: Stress distribution in plates uniformly compressed on the left and right edges. Rectangular and elliptical holes (a) superior stress distribution and (b) inferior stress distribution, reproduced from [39]

2.2.2 Perforation Location

Regardless of hole shape, $N_{cr,h}$ reduces as l_y increases in plates where $a/b > 1$, when holes are located within the first half of an 'end panel' (Figure 2.4). This provides the majority of the plate buckling resistance and is highly sensitive to stiffness variation [46, 72, 76].

Conversely, $N_{cr,h}$ may increase with l_y when the perforation is located at the buckling minimum (i.e., a 'nodal cross-line', Figure 2.4)[38, 72]. This is because a perforation here causes little reduction in plate bending stiffness, therefore the force-redistribution effect is always dominant. At a critical hole height, a perforation located here will trigger 'wavelength stiffening' (Figure 2.5).

For a perforation located at or near to a 'nodal cross-line' (Figure 2.4)[72] find that the local nodal plate bending stiffness reduces as l_x increases. When this bending stiffness reduces beyond a critical threshold, the buckling regions are shortened relative to their fundamental half-wavelength and the node moves longitudinally [72], a phenomenon Moen & Schafer (2009) coined 'wavelength stiffening' [2, 35]. This term is somewhat misleading as this effect is not always beneficial to the buckling behaviour of the plate, as shown in Figure 2.5. This increase in buckling stress does not indicate an increase in buckling capacity; a large perforation length reduces the axial stiffness of the plate [2], and a close perforation spacing is associated with a greater stress concentration [35] – effects which counteract the 'wavelength stiffening' phenomenon.

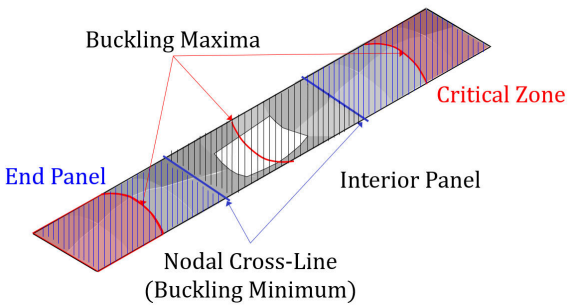


Figure 2.4: Areas of a buckling plate

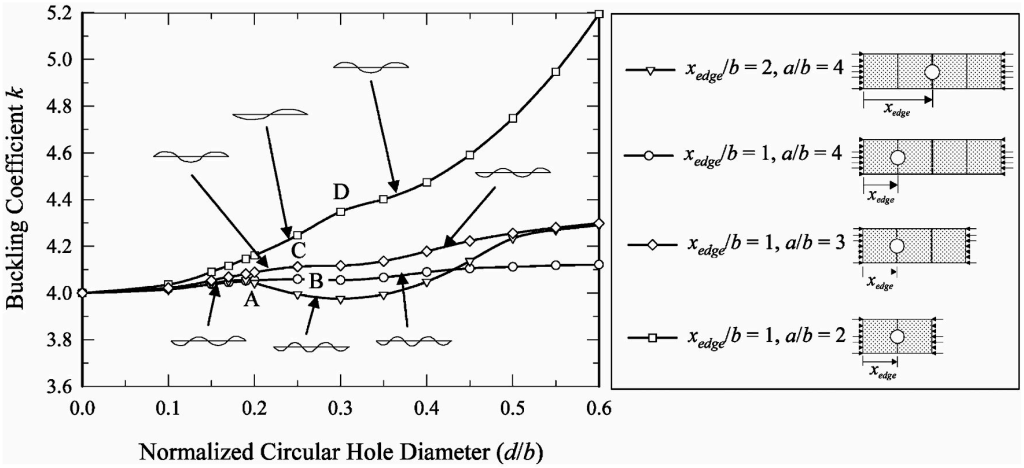


Figure 2.5: Buckling coefficient (k) when the centre of a circular hole is located at a buckling minimum. Wavelength stiffening is observed where the modeshape either gains or loses a buckling halfwave. Reproduced with permission from [72]

When a perforation is located in the 'interior panel' of a long plate, $N_{cr,h}$ can exceed N_0 [46, 72]. Using a 'sensitivity contour map' (Section 2.1.2), [46] find that as $a/b \to \infty$, the plate centre is almost unaffected by stiffness change, rendering it more optimal to locate perforations here [46]. For small holes, [46] determine that the introduction of a hole at the corners of a square plate is greatly weakening. For a square plate, the remainder of the sensitivity contour map somewhat mirrors the curvature of the buckled plate, i.e., the centre of the square plate has a greater curvature and thus is more sensitive to loss of stiffness [46].

Similarly, Gracia & Rammerstorfer (2019) [54] find that the centre of a square plate is vulnerable to the introduction of a hole. However, they find that the areas close to the centre of the plate edge are not equally sensitive to stiffness variation; a hole at the centre of the loaded edge can increase the plate buckling load whereas a similar hole at the centre of the unloaded edge cannot. The observations and methods presented by [46] and [54] are particularly good when the perforation is very small.

2.2.3 Slenderness of a Perforated Plate

The relative slenderness of an unperforated plate λ_p is given in [18] as the yield-to-elastic stress ratio $\lambda_p = \sqrt{\frac{f_y}{\sigma_{cr}}}$ (where $\sigma_{cr} = \sigma_0$ for an unperforated plate). However, the slenderness of a perforated plate depends on the number and shape of the perforations, and the perforation ratio (r_p) [4, 34, 36, 39]. Scheperboer, et al. (2016) [4] find that for the same perforation ratio, a plate with more perforations has a greater slenderness limit. For the same number of perforations, the slenderness limit increases with r_p. The authors determine that the reduction in elastic buckling strength of a perforated plate to that of an unperforated plate also depends on the hardening exponent of the material [4]. El-Sawy, Nazmy & Martini (2004) [34] use boundary curves to show how the elastic failure zone is related to the grade of steel, perforation size and aspect ratio for plates with a single perforation. The slenderness limit of a perforated plate depends on r_p and a/b. When $a/b = 1$, the slenderness limit is a minimum for $r_p = 0.2$, whereas the minimum coincides with that of an unperforated plate when $a/b = 2$.

Sweedan & El-Sawy (2011) [35] find that closely spaced circular holes in a row cause a great reduction to the longitudinal stress in the central strip of the web, and redistribution of stress near to the longitudinal edges; this is beneficial for closely spaced tall holes, but weakening for small holes. Widely spaced tall circular perforations can enable a greater elastic buckling load than the unperforated case [35]. However, when s_x is particularly large or l_x is small, $N_{cr,h}$ converges to a constant magnitude, close to that of the unperforated plate [2].

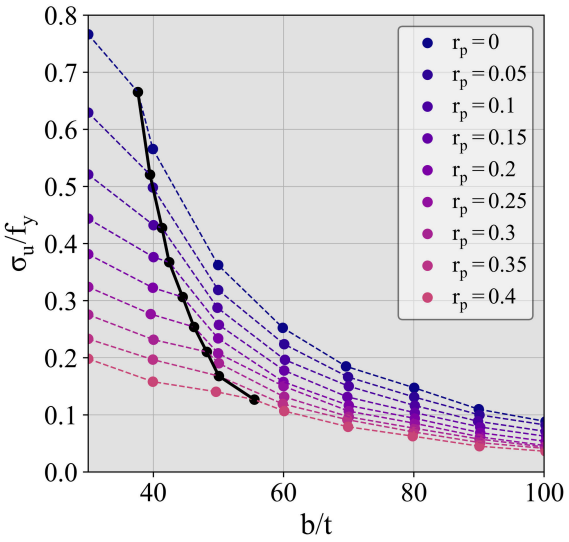


Figure 2.6: Geometric slenderness (b/t) for a square plate with 25 perforations as the perforation ratio r_p varies. The slenderness limit is shown in black. Adapted with permission from [4]

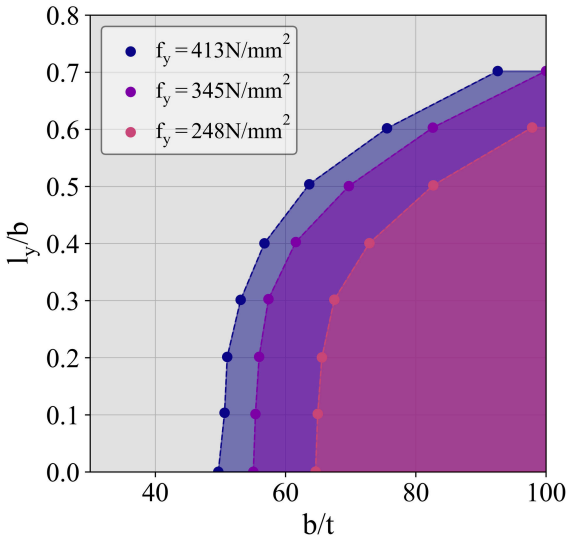


Figure 2.7: Geometric slenderness (b/t) for plates of aspect 2 with different yield strengths f_y as the perforation size varies. The elastic zone is shaded. Adapted with permission from [34]

2.2.4 Local Buckling Modes

Plates with very small perforations exhibit 'whole local buckling' or 'plate buckling with hole influence' where the modeshape is little disturbed from the unperforated local buckling modeshape, accommodated by the network of remaining material [2, 3, 42]. Plates with larger holes show local buckling at the 'unstiffened strip' between the perforation and supported edge [2, 3, 42]. When there are multiple rows of intermediately-sized holes, localised 'Euler Strip' buckling may develop between the perforations [3]; critical elastic buckling loads reduce the most when there are multiple rows of long perforations across the web-width [3].

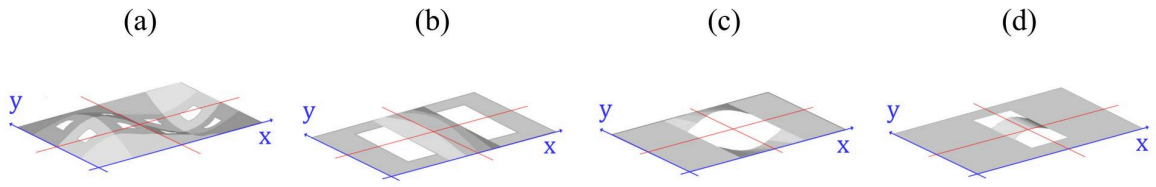


Figure 2.8: Local buckling modes of perforated plates (a) 'whole local buckling', (b) deformation dampened at a very large perforation, (c) 'unstiffened strip' buckling, (d) 'Euler strip' buckling

When the perforation height is very large, deformation is dampened at the perforation, and the section between perforations buckles [2, 35]. For plates with a single row of closely-spaced perforations of any size, [35] and [77] found that the regions closer to the loaded edges (i.e., within the 'critical zone', Figure 2.4) are most sensitive to buckling, and exhibit the greatest out-of-plane displacement, while buckling is dampened at the holes.

2.3 Optimisation

Authors employ either numerical or heuristic optimisation in structural design. Numerical topology optimisation distributes material within a prescribed design domain to optimise a structural objective, subject to design constraints [78]. However, for buckling-dominated structures, it may be difficult to formulate objective functions and constraints that adequately capture the complex interaction between geometry and instability. Furthermore, computational demand greatly increases with the number of design variables [79]. Alternative to such numerical optimisation procedures, heuristic optimisation uses rule-based and stochastic search strategies to find near-optimal solutions, overcoming these issues by incorporating randomness [80]. However, they are typically much slower to converge, and require substantial hyperparameter tuning.

Under these two categories, there are many well-established procedures for topology optimisation of structures, such as the 'bubble method' [81], 'level set method' [82], evolutionary algorithms [83], and 'Solid Isotropic Material Penalisation' (SIMP) [78]. SIMP is adopted by [84] to determine the optimal shape, number and placement of perforations in thin-walled beams. However, a mid-span stiffener is added after optimisation to achieve superior results to a traditional cellular beam [84].

Norato, et al. (2007) [85] present the 'topological-derivative' method, which explicitly indicates where a new hole will most improve the objective. However, as this method measures first-order sensitivity to an infinitesimal topological change, it can only be reliably used for members with small holes (similar to the LNS method by [46] and the Rayleigh quotient buckling sensitivity map introduced by [54]). Furthermore, all of these methods result in geometries akin to an irregular truss-like design, necessitating additive manufacturing [81–85].

Many optimisation studies for steel framed buildings use beam element discretisation, which cannot account for local buckling [86–90]. For example, buckling-specific procedures are examined by [91] and [92] for the optimisation of rack sections. Bonada, et al. (2023) [91] developed a genetic algorithm to optimise the cross section against global and distortional buckling using Generalised Beam Theory (GBT). Their method can find cross-sections with greater ultimate capacities while using less material, however it cannot be used for elastic local buckling. Similarly, [92] numerically optimise the cross section of a rack section to maximise the global buckling capacity, which is found with the formulae proposed in [93]. However, they do not optimise perforation geometry, or investigate other types of buckling failure.

Rather than creating a completely novel geometry subject to design constraints, [94] and [95] use heuristic optimisation procedures to select the best universal beam from design catalogues, subject to codified constraints. This does not allow complete exploration of the design space, and uses simplified codified expressions.

It follows that geometries obtained with numerical optimisation often require expensive additive manufacturing, or are inapplicable when perforations are large; and methods using beam element discretisation cannot account for local buckling. The latter is also an issue with some heuristic optimisation procedures, which are typically used to optimise existing section designs, subject to simplified calculations.

Shape annealing is a stochastic technique concerned with optimising the arrangement of shapes according to a 'rule-based language' specified by a 'shape grammar' i.e., an iterative application of shape transformation rules to a starting shape [96, 97]. This 'rule-based' approach involves the alteration or rearrangement of design components, rather than geometrical parameters and variable relationships [79]. As it is concerned with topological relationships, rule-based optimisation provides the designer with freedom to reset rules at a later step, thus permitting exploration and emergence [79]. Furthermore, shape annealing generates a configuration of geometries, rather than determining the values of variables in a fixed design configuration [96]; these properties render shape annealing particularly advantageous for the optimisation of perforated plates subject to buckling. The late 20^{th} century saw shape-annealing optimisation used predominantly for truss structures [98],

building on foundational work by [96] [99], and extended to domes and geodesic design [100]. However, research on the application of shape annealing for structural optimisation then stalled, and it has not been yet used for the optimisation of perforated sections.

2.4 Experiment Design

This section is extracted from [25]; forming a background to the study conducted in Chapter 5. Some authors conduct experimental investigations on columns with a linear arrangement of perforations in their webs to suggest whether current design code predictions are appropriate, see [76, 101–104]. Some of these studies have been used to inform modern standards (see Section 2.5). Simply supported boundary conditions provide a conservative baseline for the elastic buckling load of the webs of members. Simply supported edges are difficult to represent experimentally, regardless of whether a plate is perforated or not. In [105–107], authors achieved simply-supported boundaries for experimental tests on unperforated plates by various means. In each of the studies by [105–107], two edges were slotted within a round bar enabled to rotate by sets of roller bearings, and the remaining two edges were supported with either a square groove [107], knife-edge support [105], or discrete finger supports [106]. Square grooves restrict rotation so are arguably not pinned. Finger supports are most representative of perfectly pinned conditions; however, these are difficult to manufacture. Knife-edge supports minimise frictional resistance but change the plate aspect ratio. In addition, it is ideal to use the same support type for all edges, and it would be difficult to use finger supports or a knife-edge condition for a loaded edge.

In [108] and [109], perforated plates with all four edges supported were tested. Neither author considers any of the edge conditions discussed above, but both apply a consistent support type along all four edges: [109] uses sets of needle bearings to provide pinned boundary conditions, and [108] considers an elliptical groove. Similarly, [110] applied compressive loading to two edges of a perforated thin steel plate using a loading band with an elliptical groove. Although not experimentally demonstrated, [9] suggests that V-notches permit complete freedom of edge rotation as closely as can be replicated under experimental conditions. However, they note that for plates with unrounded edges, any rotation will be accompanied by some slipping in the groove [109].

To mitigate the difficulties associated with simply supported plate boundaries, [111] consider rotational boundary conditions, arguing that these better reflect the behaviour of plates in maritime structures. The authors studied two loading configurations in which a panel with a large hole embedded within a girder functions as a perforated plate [111]. Many authors scale-down this arrangement to study the influence of small perforations in rack sections, assuming the web to be simply-supported [101] [76]. Most consider an arrangement where the rack section bares directly onto a platen and is unsupported longitudinally, assuming that the web of these members represents a simply-supported plate.

However, this has been shown to be inaccurate; [111] [104] reveal that the experimental setup exhibits a greater capacity than a computational model with perfectly pinned edges. Ketsi (2000) [104] summarises that 'the restraint given by the flanges [of open sections] is rather more rigid than the simply supported and local buckling coefficient value', and [35] write that 'the actual boundary conditions for a real web plate may be somewhere between the all-fixed and the all-simply supported extremes'. Hence the web of a structural section can be conservatively approximated as a simply supported plate, but a simply supported plate will be unconservatively approximated with the web of a structural section.

2.5 Design Methods in Current Standards

2.5.1 The Direct Strength Method (DSM)

This section underpins the study presented in Chapter 6 on DSM for plates with perforations, and is an extract from [24]. Developed at the end of the 20th Century, The Direct Strength Method (DSM) is the primary method used to determine the theoretical ultimate capacity of cold-formed structural steel members in American and Australian design standards [7, 22, 112]. It requires knowledge of the global, local and distortional elastic cross-sectional buckling capacities, $N_{cr,e}$, $N_{cr,l}$ and $N_{cr,d}$ respectively. $N_{cr,d}$ is typically determined with the numerical finite strip method (FSM), whereas $N_{cr,e}$ and $N_{cr,l}$ can be easily determined analytically. The nominal ultimate column capacity is thence the minimum of the global, local and distortional nominal ultimate buckling capacities

($N_{u,e}$,$N_{u,l}$ and $N_{u,d}$ respectively), determined with the following set of equations:

$$N_u = \min(N_{u,e}, N_{u,l}, N_{u,d}) \tag{2.6}$$

$$N_{u,e} = \begin{cases} 0.658^{\lambda_c^2} N_y & \text{if } \lambda_c \leq 1.5 \\ \dfrac{0.877 N_y}{\lambda_c^2} & \text{if } \lambda_c > 1.5 \end{cases} \tag{2.7}$$

$$N_{u,l} = \begin{cases} N_{u,e} & \text{if } \lambda_l \leq 0.776 \\ N_{u,e} \cdot \left(1 - 0.15 \left(\dfrac{1}{\lambda_l}\right)^{0.8}\right) \cdot \left(\dfrac{1}{\lambda_l}\right)^{0.8} & \text{if } \lambda_l > 0.776 \end{cases} \tag{2.8}$$

$$N_{u,d} = \begin{cases} N_y & \text{if } \lambda_d \leq 0.561 \\ N_y \cdot \left(1 - 0.25 \left(\dfrac{1}{\lambda_d}\right)^{1.2}\right) \cdot \left(\dfrac{1}{\lambda_d}\right)^{1.2} & \text{if } \lambda_d > 0.561 \end{cases} \tag{2.9}$$

Where the global, local and distortional slenderness are given by $\lambda_c = \sqrt{N_y/N_{cr,e}}$, $\lambda_l = \sqrt{N_{u,e}/N_{cr,l}}$ and $\lambda_d = \sqrt{N_y/N_{cr,d}}$ respectively. This series of simple empirical formulae are particularly intuitive, able to evaluate complex cross-sections much faster than tabulated approaches, making DSM particularly appealing for the design of cold-formed sections.

However, these equations cannot easily apply to perforated members, typically requiring a semi-analytical analysis [113]. Furthermore, [114] find that the DSM leads to poor predictions where used to design perforated steel rack columns when failure is governed by local buckling. They conclude that the DSM is suitable for determining the strength of perforated columns failing by distortional or global buckling, provided local buckling is not involved.

To extend the DSM equations to members with perforations, [113] consider Equation 2.7 and Equation 2.8 unchanged, with the nominal local buckling capacity limited to the net section yield strength $N_{y,net}$, and the critical elastic buckling loads ($N_{cr,e}$, $N_{cr,l}$, $N_{cr,d}$) determined including the influence of perforations. As described in [7], the analytical method presented in [115] is used to determine the global elastic buckling load ($N_{cr,e}$) for members with perforations. Concerning the local buckling prediction ($N_{cr,l}$) for members with perforations, [7] refers to research completed by [3] to determine the elastic buckling load of members with 'patterned holes', or it recommends the use of numerical methods

such as the FE method [7]. Patterned holes are defined as small and tightly spaced, in a regular array of rows and columns for the complete length of a member, and the corresponding elastic buckling load is found using Equation 2.3 [3].

As an alternative to FE, [2] assume $N_{cr,l}$ is min(N_0 , $N_{cr,B}$), where N_0 is the unperforated elastic critical buckling capacity, and $N_{cr,B}$ (Equation 2.2) is the local elastic buckling load of a perforated plate. Although this procedure mitigates the use of numerical methods, it does not account for plates containing a group of eccentrically located perforations, or multiple perforations arranged over the web.

Yao & Rasmussen (2017) [116] use FE to investigate perforated and unperforated cold-formed steel columns with varied dimensions, cross-sections, imperfections and material properties. The authors discuss the performance of C-sections, with perforations equally spaced at the web centreline. They use $N_{cr,B}$ to determine the perforated local elastic buckling load, and make the assumption that the local buckling capacity of an unperforated plate N_0 can be used in Equation 2.8 such that $\lambda_l = \sqrt{N_{u,e}/N_0}$. In [5], Yao & Rasmussen (2017) impose a penalty factor of 0.85 on Equation 2.6, raising the mean accuracy of their DSM expression for all failure modes from 0.993 to 1.154. They undertake a regression analysis to improve this penalty factor and optimise the mean accuracy and standard deviation across a range of different cross section shapes and buckling modes. The resulting method is suitable for use with both perforated and unperforated members. The authors conclude that the variation in their prediction for nominal buckling capacity is typically due to interaction of global, local and distortional modes. They conclude that perforations influence the ultimate capacity differently depending on the failure mode, cross section type, section dimensions and perforation dimensions; and further research to examine these variables is recommended by the authors [5].

2.5.2 The Effective Width Method (EWM)

The Effective Width Method (EWM) is a design approach for slender plates where post-buckling stress redistribution is represented by an effective width that carries uniform stress, accounting for elastic post-buckling reserve (Section 1.3.3). To use the EWM for plate buckling analysis, EN 1993-1-5 [18] clause 2.3 requires that the size of the unstiffened hole does not exceed 5% of the plate width, and if this limit is exceeded, such cases should be analysed with FE methods. Furthermore, EN 1993-1-5 [18] does not explicitly provide guidance on using the EWM to determine $N_{cr,h}$ for plates with multiple perforations.

Using the EWM, [19] provides specific criteria for different perforation sizes, spacings and arrangements permissible to limit their effect on the member's buckling resistance; the effective width of the flange or web is reduced relative to these geometries. However, the EWM assumes continuous, uniform plates with smooth post-buckling stress redistribution. Perforations disrupt these stress fields, introducing local stress concentrations where these assumptions quickly become inaccurate, particularly for plates with an irregular arrangement of perforations. Taranu, et al. (2025) [117] follow use EWM presented in [118], finding that it underestimates local buckling at perforations in cold-formed steel uprights by 15.8%.

2.6 Code Constraints and Geometric Limits

2.6.1 Sections with Large Web Openings [6]

EN 1993-1-13 [6] imposes geometric limits on the design of sections with tall perforations subject to flexure. It requires that unstiffened openings must comply with the limiting dimensions of Table 2.3, including the constraint that $S_{yo,min} = 0.125b$ for rectangular web openings. Additional geometrical restrictions are imposed by [119] who recommend that $S_{xo,min} \geq 0.5l_y$ for circular openings, and $S_{xo,min} = \max(l_x, b)$ for rectangular openings. These recommendations agree with [46],[72] and [38], who find that this region is particularly sensitive to stiffness variation; locating large openings here will cause a greater reduction to the elastic buckling load, regardless of hole shape, orientation or plate aspect ratio (Section 2.2.2).

Table 2.3: Dimensional limits imposed by [6] on web openings

Shape of Opening	Effective Opening Length A_{eff}	Max Opening Height $l_{y,max}$	Max Opening Length $l_{x,max}$	Min Edge-to-Edge Spacing s_x	Min Depth of Tee in Compression (mm)
Circular ('Cellular')	$0.7l_y$	$0.8b$	N/A	$0.1l_y$	$\max(t_f + r + 10, t_f + 30)$
Hexagonal ('Castellated')	$0.7l_y$	$0.75b$	$1.5l_y$	$0.25l_y$	$\max(t_f + r + 10, t_f + 30)$
Rectangular	l_x	$0.75b$	$2.5l_y$	$\max(0.5l_x, l_y)$	$\max(l_x/12, 0.1b)$
Elongated	$l_x - 0.3l_y$	$0.8b$	$3.0l_y$	$\max(0.25A_{eff}, 0.5l_y)$	$A_{eff}/12$
Sinusoidal ('Angelina')	$l_y + 0.5l_x$	$0.8b$	$5.0l_y$	$0.25l_y$	$A_{eff}/12$

Although [6] places no limit on the minimum opening height or length, studies find that closely spaced holes influence stress distribution (Section 2.2.2). Lawson & Hicks (2011) [119] acknowledges three sources of stress at the web-post; (i) longitudinal (horizontal) shear on the narrowest width, (ii) compression due to transfer of vertical shear and (iii) bending developed due to Vierendeel action. Lawson & Hicks (2011) [119] explains that buckling is influenced by a combination of all of these effects. This is reflected in [6] which specifies a prediction for the web-post buckling capacity ($N_{wp,Ed}$) under combined shear ($V_{wp,Ed}$) and moment ($M_{wp,Ed}$), and the web buckling capacity ($N_{w,Ed}$) under transverse shear force ($V_{w,Ed}$). However, [6] does not consider members buckling under uniform axial compression.

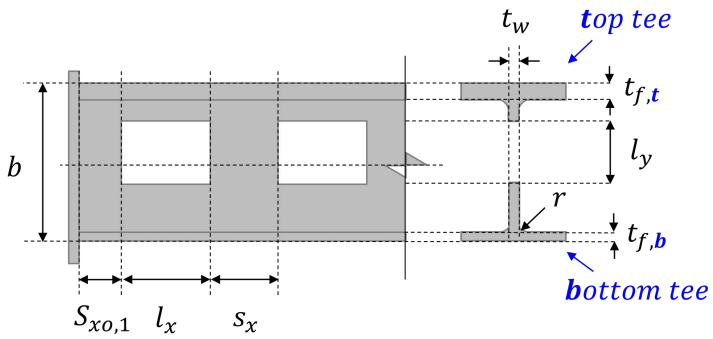


Figure 2.9: Geometry of sections with tall holes, modified from [6]

2.6.2 Rack Sections and Perforated Cold-Formed Members [93]

BS EN 15512 [93] is a European standard providing guidance for the design of rack sections, specifically how to account for the reduced member strength using the EWM. As detailed in Section 2.5.2, this approach is generally conservative as it cannot capture the redistribution behaviour (Section 2.2.1), and will always determine a reduced capacity as a direct consequence of reduced cross-sectional area. For members in tension, [93] restricts the spacing of two consecutive holes s_x and s_y by the minimum cross-sectional area of the perforated web A_{min}. This is given as the minimum section length reduced by holes at right angles to the direction of stress in the member, and constrained by:

$$N_{t,RD} = \frac{A_{min} f_y}{\gamma_{M0}} \tag{2.10}$$

Unlike members with large openings (Section 2.6.1), [93] makes no restriction on s_x and s_y for members in compression. It requires that the design resistance of perforated sections "shall be determined by tests or alternatively by a theoretical procedure. The theoretical procedure shall take rational account of the perforation together with local, global, and distortional buckling and imperfections. In the case of continuously perforated members, the theoretical procedure shall be validated by tests" [93]. BS EN 15512 [93] and ANSI MH16.1-2023 [120] propose that FE, FSM, or the buckling curves in [118] are suitable such procedures. If these buckling curves are considered, no provision is made for the perforations; the gross cross-section is assessed in order to find $N_{cr,e}$ rather than $N_{cr,l}$ [93].

3

Elastic Buckling of Slender Steel Plates with Perforations

3.1 Commentary

Section 1.2 proposed that semi-analytical methods could compute solutions with a greater computational efficiency than numerical methods, and are more broadly applicable than empirical approaches. Section 2.1.3 introduced the Rayleigh quotient and Rayleigh-Ritz energy methods, which are used by [3, 53–55] to evaluate perforated plate buckling. As closed-form PDE solutions are typically impossible to establish, energy methods are particularly advantageous for this problem as they work with admissible approximations to the displacement field, and are applicable to a very large geometric domain. None of the methods developed by [3, 53–55] assess plates with an irregular array of perforations, preventing them from completely supplementing numerical FE procedure, required in response to issue (ii) (Section 1.2). This chapter presents my examination into the modified energy prediction (MEP), an analytical method I developed as a surrogate to FE; the work is published in [26]. By considering Rayleigh's energy method, this chapter proposes a mathematical formulation to obtain the elastic buckling capacity of a slender plate with many perforations. Comparisons are made to finite element results, and alternative formulations. In addition to an existing empirical expression, the proposed method is

found to give accurate predictions for the elastic buckling capacity, except for plates with perforations of a large aspect ratio, or plates with one row of particularly tall perforations.

I conducted a mesh convergence study on S8R5 elements, which was not included as part of the conference paper, and is documented in Appendix A. Additionally, the parametric investigation undertaken in this chapter was automated with a meshing script I wrote in python. This mitigated time-consuming model creation and meshing in the graphic user interface (GUI), and was used to produce meshes with elements of a uniform aspect ratio, with regular sizes imposed by the user. This script received various amendments over the thesis duration, and the final version is provided in Appendix B.

3.2 Introduction to Plate Buckling Theory

Perforations equally spaced in a grid-like arrangement in a plate are considered as a 'regular array'. An 'irregular array' of perforations describes a random distribution of differently sized holes, or patterned arrangement that is not regular. Depending on various parameters (such as hole size, location, number), local buckling of a perforated plate may evolve as one of four possible manifestations (Figure 3.1).

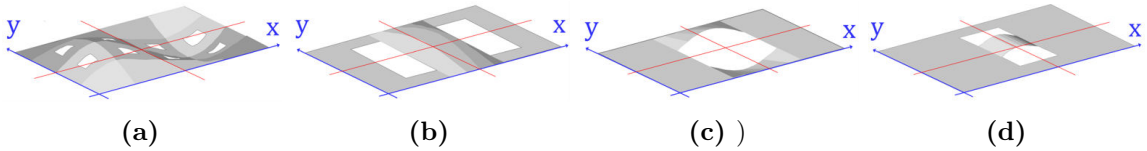


Figure 3.1: Local buckling manifestations of perforated plates (a) Whole Plate Buckling (WPB), (b) Lateral Perforation Buckling (LPB), (c) Unstiffened Strip Buckling (USB), (d) Euler Strip Buckling (ESB)

Timoshenko [8] derived an expression for the internal bending strain energy of an unperforated plate U_0, given the material properties (plate flexural constant D, and Poisson's ratio ν), in terms of the displacement function $w_0(x, y)$ over the plate's surface area S:

$$U_0 = \left(\frac{D}{2}\right) \int_S \left(\frac{\partial^2 w_0(x,y)}{\partial x^2} + \frac{\partial^2 w_0(x,y)}{\partial y^2} \right)^2 - 2(1-\nu) \left(\frac{\partial^2 w_0(x,y)}{\partial x^2} \cdot \frac{\partial^2 w_0(x,y)}{\partial y^2} - \left(\frac{\partial^2 w_0(x,y)}{\partial x \partial y} \right)^2 \right) dS \quad (3.1)$$

A rectangular plate with simply supported boundary conditions on all four sides is a lower bound approximation to the web of a cross section. Navier [10] determined an expression for the displacement function of a simply supported plate as a double Fourier series of sinusoidal terms, with amplitudes denoted A_{mn}. If small deflection theory is considered where axial effects are neglected, the surface of a buckled plate can be approximated by this function:

$$w_0(x,y) = \sum_{m=1,3,\ldots}^{\infty} \sum_{n=1,3,\ldots}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \tag{3.2}$$

Where a and b denote the length and width of the plate respectively. According to elastica theory, the shortening of an edge under uniform load N_x can be expressed in terms of this out of plane displacement. Accordingly, the external work done to buckle a plate (W_0) is determined:

$$W_0 = \frac{1}{2} \int_S N_x \left(\frac{\partial w_0(x,y)}{\partial x}\right)^2 dS \tag{3.3}$$

For a simply supported plate, the boundary conditions are known, and the governing differential plate equation can be solved. The critical state is obtained when all amplitude factors A_{mn} (Equation 3.2) are equal to zero, except one [9]. The same result is obtained when internal bending strain energy, (Equation 3.1), and external work, (Equation 3.3), are equated. The final expression for the local elastic buckling capacity of an unperforated plate N_0 depends on the plate's geometry (a,b,t), material properties (D, ν), and the number of longitudinal halfwaves (m). This expression is classically given in terms of the plate buckling coefficient (k):

$$N_0 = \frac{\pi^2}{b^2} \cdot \frac{Et^3}{12(1-\nu^2)} \cdot \left(\frac{mb}{a} + \frac{a}{bm}\right)^2 = \frac{\pi^2 D}{b^2} \left(\frac{mb}{a} + \frac{a}{bm}\right)^2 = k\frac{\pi^2 D}{b^2} \tag{3.4}$$

If defined only by the ratio b/t, a perforated plate may be defined as 'slender' yet undergo elasto-plastic failure. The slenderness depends additionally on the number, shape, size, location and arrangement of perforations [34, 121].

The objectives of this research are to answer the following questions. Firstly, can existing methods be used outside of the domain within which they were established, to accurately determine the critical elastic buckling capacity of perforated plates? Secondly, can a new solution based on the energy method give a better prediction for the critical elastic buckling capacity of perforated plates, for the entire domain investigated?

3.3 Calibration of Finite Element Model

To determine the nature of the relationship between the critical elastic buckling capacity of a perforated simply supported thin plate and the parameters discussed, performance is investigated with the finite element (FE) method. This is determined with the ABAQUS linear perturbation procedure [122], which solves the eigenvalue problem of the constitutive and geometric stiffness matrices. The material is steel with a Young's Modulus of 203.4 GPa, and a Poisson's ratio of 0.3. It is assumed to have isotropic properties and obey the stress-strain relationships for isotropic materials. The perforated plate buckles in the elastic regime, with slenderness determined in accordance with [4, 34]. The following limit ensures the plate is slender for all variables investigated in terms of the plate width (b) and thickness (t):

$$\min\left(\frac{b}{t}\right) = 100 \tag{3.5}$$

As the plate thickness is very small in comparison to the plate dimensions, the problem is adequately described by Kirchhoff shell theory, and thin shell quadratic elements with reduced integration (S8R5) are chosen. A structured mesh is used to negate transverse shear locking and allow for easier data management and thus faster convergence. It has been shown that a plate with a central circular perforation of diameter equal to the width of a square perforation is of greater capacity [42, 74]. This is supported in an assessment on uniaxially compressed square plates undertaken by Narayanan & Chow [109] and holds for eccentric perforations. Yu & Davis [74] attribute this to the difference in stress concentration, opposed to the difference in area of the two perforation shapes. Therefore, this research evaluates plates with an array of rectangular and square perforations, and assumes that the latter is a conservative representation of a plate with circular perforations. All perforations are oriented with their major and minor dimensions parallel to the axes of the global co-ordinate plane, which allows for rectangular FE elements everywhere. A limit is placed on the aspect ratio of elements in the mesh to further ensure accurate convergence, according to an investigation by Moen [77]:

$$\min\left(\frac{a_e}{b_e}\right) = 1, \max\left(\frac{a_e}{b_e}\right) = 1.5 \tag{3.6}$$

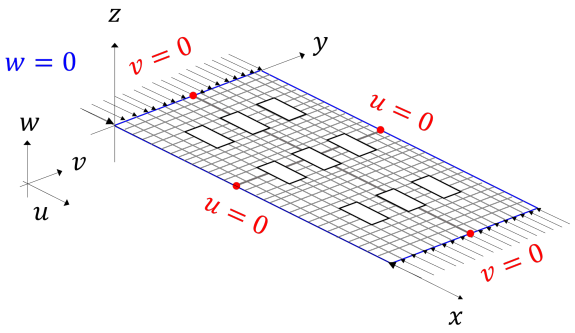


Figure 3.2: Simply supported boundary conditions and nodal loading simulated in ABAQUS

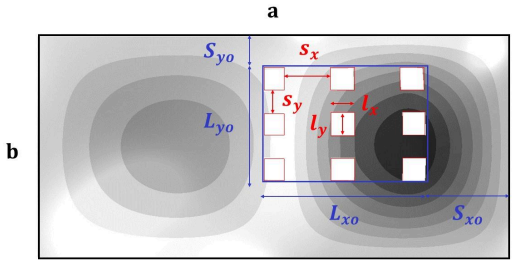


Figure 3.3: Regular variables examined

Where a_e and b_e represent the length and width of the finite element respectively. Simply supported boundary conditions are represented as shown in (Figure 3.2). This is similar to the conditions applied by Sweedan & El-Sawy [35] who examine a symmetrical distribution of circular perforations arranged along the length. Compressive uniform edge loading is simulated by consistent nodal loads, as considered by Moen & Schafer [2].

ABAQUS input files are generated with a custom-built python script where these meshing, loading and boundary conditions are prescribed to the model (Appendix B). The limit on the maximum element size is chosen by validating the model to a study conducted by Moen & Schafer [2], who examined 32 FE models of plates with rectangular holes with a corner radius. Despite the holes in this study being sharp-cornered, the results follow the same trend. The element size limit is defined as:

$$\max\left(\frac{b_e}{b}\right) = 0.02 \tag{3.7}$$

3.4 Parametric Assessment

Regular and irregular perforation arrangements are examined. Rectangular holes may perforate a rectangular area defined by L_{xo} and L_{yo}, which may be located eccentrically about the plate centroid (Figure 3.3). The closest distance of this region to the transverse and longitudinal edge is S_{xo} and S_{yo} respectively. Within this region, a regular arrangement of perforations is defined by rows and columns and the number of perforations in each are n_r and n_c respectively. The perforation spacing and size in each direction is also defined in Figure 3.3. The domain of these variables is summarised in Table 3.1 and Table 3.2 for regular and irregular perforations respectively.

Table 3.1: Domain of variables examined – regular arrangement of perforations (see Figure 3.3)

	a/b	b/t	n_r	n_c	S_{xo}/a	S_{yo}/b	L_{xo}/a	L_{yo}/b	s_x/a	s_y/b	l_x/a	l_y/b
Minimum	1.00	100	1.00	2.00	0.040	0.040	0.1	0.1	0.006	0.025	0.005	0.02
Maximum	4.00	100	5.00	30.0	0.40	0.40	0.90	0.90	0.70	0.80	0.33	0.80

Table 3.2: Domain of variables examined – irregular arrangement of perforations (see Figure 3.3)

	a/b	b/t	l_x/a	l_y/b
Minimum	1.00	100	0.009	0.02
Maximum	4.00	100	0.30	0.30

A total of 486 parametric models with regularly spaced perforations have been generated such that the variables are randomly selected from the domain using the python 'randrange' function [123]. The elastic buckling capacity determined with ABAQUS ($N_{cr,FE}$) is compared to the prediction given by [4] (Figure 3.4(a)) and [3] (Figure 3.4(b)) ($N_{cr,S}$ and $N_{cr,SM}$ respectively). The ratio of the FE result to that predicted by either author for the perforated plate elastic buckling capacity is given on the y-axis. The perforation ratio (r_p) is the ratio of the total hole area to the gross plate region (Section 1.3.2). Note that each prediction is applied outside of its established domain. Scheperboer, et al. (2016) [4] examine staggered circular perforations in a square plate, and Smith & Moen (2014) [3] examine a regular array of rectangular perforations symmetric about the mid-length and mid-width of the plate.

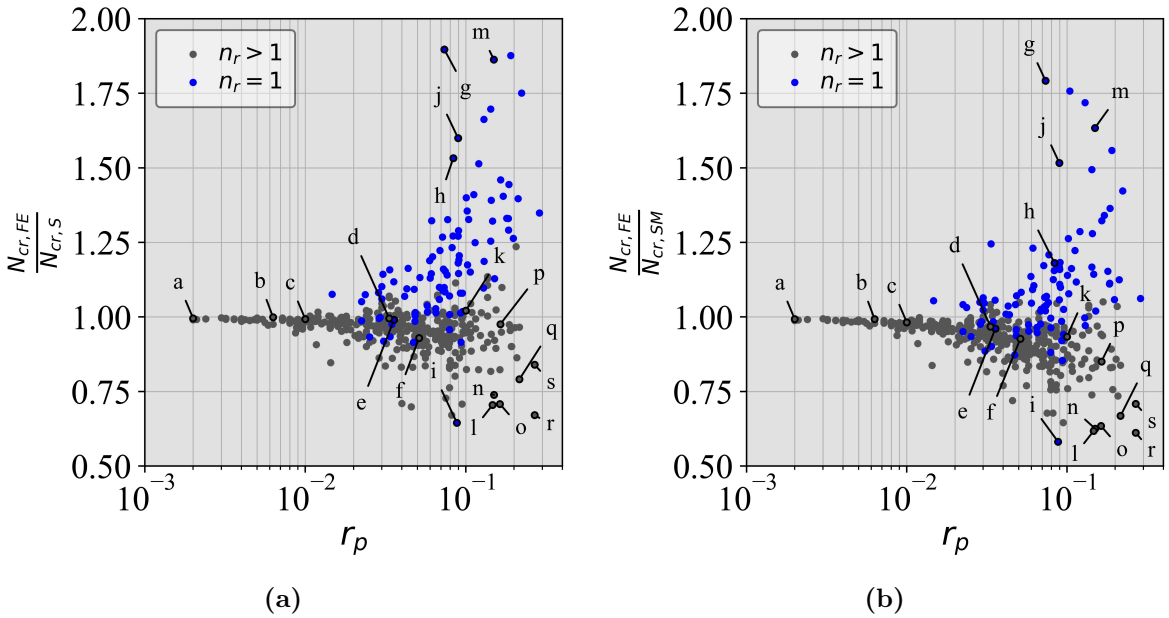


Figure 3.4: Accuracy of the prediction by (a) $N_{cr,S}$ [4] and (b) $N_{cr,SM}$ [3] in comparison to FE results for a regular array of perforations

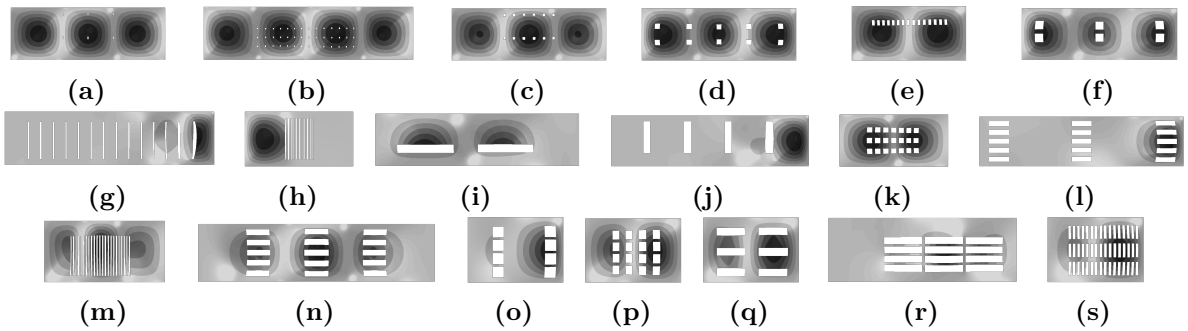


Figure 3.5: Examples of examined plates with a regular arrangement of perforations

For one row of perforations ($n_r = 1$), the prediction by both authors is conservative on average; with a mean ($N_{cr,FE}/N_{cr,SM}$) of 1.11 and standard deviation of 0.195 compared to a mean ($N_{cr,FE}/N_{cr,S}$) of 1.20 and standard deviation of 0.222, the prediction given by Smith & Moen (2014) [3] should be used in this case. This standard deviation is a magnitude greater than the standard deviation for models with more than one row of perforations ($n_r > 1$). It can be seen in Figure 3.5 that a large proportion of plates with one row of perforations do not exhibit whole plate buckling. The capacity of models (g), (h) and (j) is underpredicted; these models exhibit lateral perforation buckling and an increased elastic buckling capacity. Model (m) (Figure 3.5) gains a halfwave in comparison to the

unperforated plate buckling modeshape, and exhibits an increased elastic buckling capacity (i.e., 'wavelength stiffening' [2]). The capacity of Model (i) (Figure 3.5) is overpredicted, and less than the elastic buckling capacity of an equivalent unperforated plate, it clearly exhibits unstiffened strip buckling, not whole plate buckling.

Models with more than one row of perforations exhibit a smaller regression from the prediction derived by [4]. The mean for the 393 models with more than one row of regular perforations is unconservative for both predictions and reduces in accuracy as the number of rows increases. The accuracy reduces at a greater rate for the prediction found by [3] than for [4] and this is particularly obvious for models with a large perforation ratio (Figure 3.4(b)). For all 486 models the mean (μ) and standard deviations (σ) are 1.01 and 0.148 for [4], and 0.962 and 0.129 for [3] respectively. The prediction given by [3] cannot be applied to plates with an irregular array of perforations and only the prediction by [4] is examined. For such an arrangement, there is no regular spacing or distinguishable rows or columns. Staggered holes and a geometric pattern of holes are included in this definition, in addition to perforations occupying random location within the plate (Figure 3.7). The domain of variables is summarised in Table 3.2.

The accuracy of the prediction determined by [4] for all 320 models with irregular perforations is less than that found for a regular arrangement of perforations (μ=0.944, σ=0.0647). A relationship between elastic bucking capacity and perforation ratio is even less obvious (Figure 3.6). For example, (c), (d) and (e), and separately (k) and (l) (Figure 3.7), have the same perforation ratio (and so the same predicted capacity according to [4]) but exhibit a range of capacities. Models (a) and (e) (Figure 3.7) are most well-predicted, despite the prediction by [4] being established in the domain of square plates.

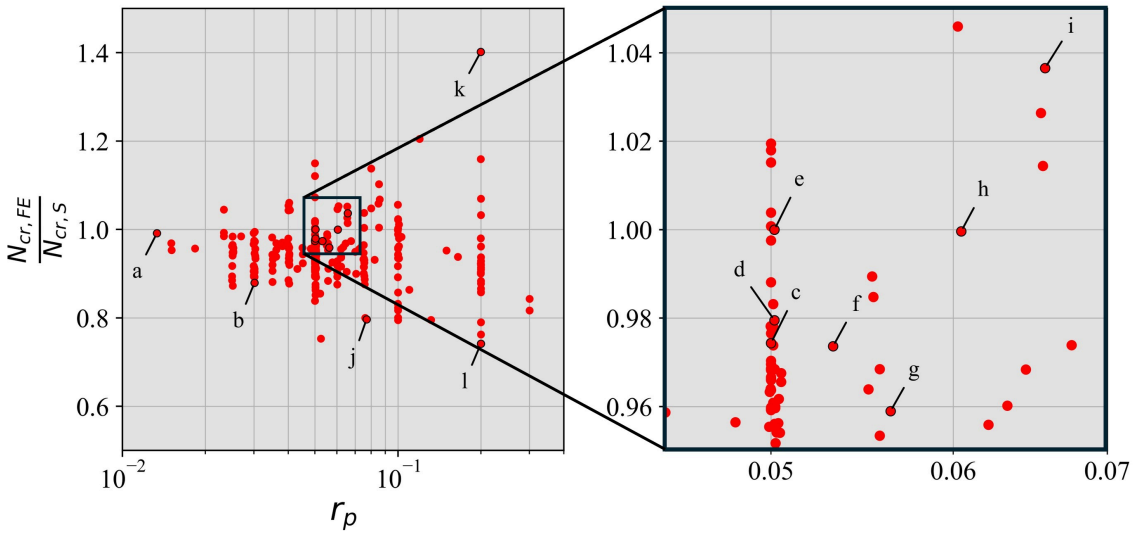


Figure 3.6: Accuracy of the prediction $N_{cr,S}$ by [4] in comparison to FE results for an irregular array of perforations. Closer detail is provided on the right.

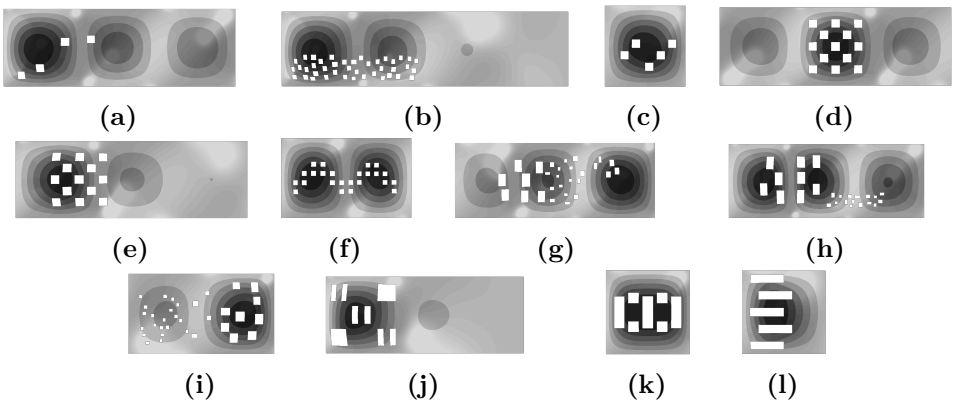


Figure 3.7: Examples of examined plates with an irregular arrangement of perforations

3.5 Modified Energy Prediction

Consider a plate subject to uniform compression N_x per unit width along the short edge. When a hole is present, its length along A'A' can no longer provide resistance and the force at this location must be redistributed into the section adjacent to the hole (Figure 3.8). This force F_x is an unknown function of y (i.e., $f(y)$) which may be different above and below the hole ($f_1(y)$, $f_2(y)$, Figure 3.8).

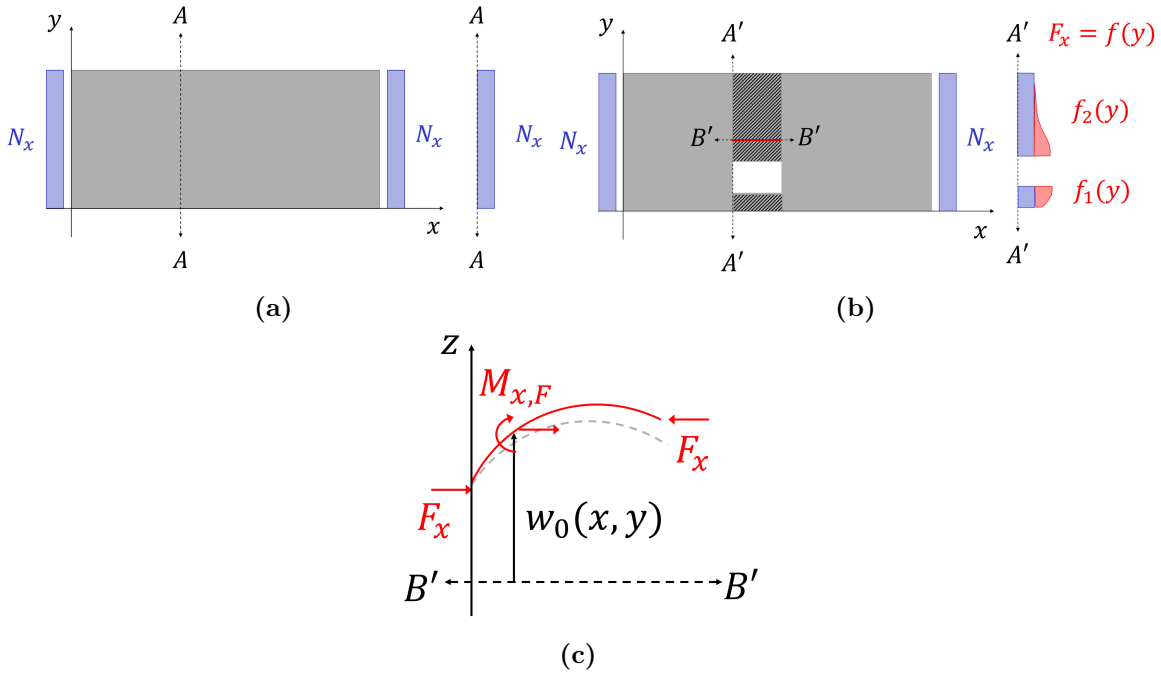


Figure 3.8: Assumptions for the Modified Energy Prediction (a) The force distribution within an unperforated plate (b) The force distribution within a perforated plate (c) A length B'B' along the adjacent surface above the hole will displace by a function $w(x, y)$ (grey). The unperforated plate function $w_0(x, y)$ is assumed (red)

Let the sections above and below the hole be termed the 'adjacent surfaces'. It is assumed that the force F_x per unit width of plate is constant along the length of the hole B'B' (Figure 3.8). The length B'B' displaces by some unknown function $w(x, y)$. It is hypothesised that the modeshape $w_0(x, y)$ of the unperforated plate is an approximate representation of this unknown function (Equation 3.2, Figure 3.8). The additional moment in the longitudinal direction $M_{x,F}$ is given in terms of this additional force and unperforated plate displacement function $w_0(x, y)$:

$$M_{x,F} = F_x w_0(x, y) \tag{3.8}$$

When perforations are arranged in an array, the adjacent surface can be narrow and unsupported along its length. It is assumed that the curvature is uncoupled (beam-like) hence the additional force F_x per unit width is related only to the curvature in the x-direction (χ_x). It follows that the additional energy along the length B'B' (Figure 3.8) is given by:

$$U_F = \frac{1}{2}\int_{x_1}^{x_2} M_{x,F}\chi_{x.0}dx, \tag{3.9a}$$

$$U_F = -\frac{1}{2}\int_{x_1}^{x_2} (F_x w_0(x,y)) \cdot \left(\frac{\partial^2 w_0(x,y)}{\partial x^2}\right) dx \tag{3.9b}$$

Therefore, for the case of a single perforation, the total additional internal strain energy can be expressed as (with x_1, x_2, y_1 and y_2 shown in Figure 3.9):

$$U_F = -\frac{1}{2}\int_{x_1}^{x_2} \left(\int_0^{y_1} f_1(y) \cdot w_0(x,y) \cdot \frac{\partial^2 w_0(x,y)}{\partial x^2}\, dy + \int_{y_2}^{b} f_2(y) \cdot w_0(x,y) \cdot \frac{\partial^2 w_0(x,y)}{\partial x^2}\, dy \right) dx \tag{3.10}$$

When an array of n perforations is considered, the superposition of the individual i^{th} hole is considered, such that:

$$U_F = \sum_{i=1}^{n} U_{F,i}, \tag{3.11a}$$

$$U_{F,i} = -\frac{1}{2}\int_{x_{i,1}}^{x_{i,2}} \left(\int_0^{y_{i,1}} f_{i,1}(y) \cdot w_0(x,y) \cdot \frac{\partial^2 w_0(x,y)}{\partial x^2}\, dy + \int_{y_{i,2}}^{b} f_{i,2}(y) \cdot w_0(x,y) \cdot \frac{\partial^2 w_0(x,y)}{\partial x^2}\, dy \right) dx \tag{3.11b}$$

Considering horizontal equilibrium for each of the i holes, the force function $f_{i,j}(y)$ can be expressed in terms of the applied load function $g(y)$ which is uniform (N_x for the width b). The redistribution of forces adjacent to a perforation is highly nonuniform [53]. However, for this research, the critical elastic buckling capacity is evaluated considering a uniform redistribution function and we may assert that a function more representative of the true force-field will be of greater accuracy. Therefore, we can express the redistributed force as $F_{x,i}$ for the remaining width above and below the i^{th} hole:

$$N_x = \int_0^b g(y)\,dy = \left(\int_0^{y_{i,1}} g(y)\,dy + \int_{y_{i,2}}^b g(y)\,dy\right) + \left(\int_0^{y_{i,1}} f_{i,1}(y)\,dy + \int_{y_{i,2}}^b f_{i,2}(y)\,dy\right), \tag{3.12a}$$

$$N_x(y_{i,2} - y_{i,1}) = \int_0^{y_{i,1}} F_{x,i}\,dy + \int_{y_{i,2}}^b F_{x,i}\,dy, \tag{3.12b}$$

$$F_{x,i} = \frac{N_x(y_{i,2} - y_{y,1})}{b - (y_{i,2} - y_{i,1})} \tag{3.12c}$$

The additional strain energy is a function of the applied force. When considering superposition, it is noted that the adjacent surface may not exist (i.e., the adjacent surface of the i^{th} hole may contain another hole, see Figure 3.9). However, in comparison to the total strain energy, this difference may be considered negligible. There is strain energy lost at the perforations:

$$U_P = \frac{D}{2}\int_{x_{i,1}}^{x_{i,2}}\int_{y_{i,1}}^{y_{i,2}} \left(\frac{\partial^2 w_0(x,y)}{\partial x^2} + \frac{\partial^2 w_0(x,y)}{\partial y^2}\right)^2 - 2(1-\nu)\cdot\left(\frac{\partial^2 w_0(x,y)}{\partial x^2}\cdot\frac{\partial^2 w_0(x,y)}{\partial y^2} - \left(\frac{\partial^2 w_0(x,y)}{\partial x \partial y}\right)^2\right) \tag{3.13}$$

With U_0 considered for the unperforated plate surface Equation 3.1, the total internal strain energy of the perforated plate is therefore:

$$U = U_0 + U_F - U_P \tag{3.14}$$

Rayleigh's quotient is solved, where the external work is conservatively given as established by Smith & Moen [3], and the critical elastic buckling force of the perforated plate $\widehat{N}_{cr,h}$ is determined:

$$\widehat{N}_{cr,h} = \frac{U_0 - U_P}{\frac{1}{2}\int_S \left(\frac{\partial w_0(x,y)}{\partial x}\right)^2 dS - \frac{U_F}{N_x}} \tag{3.15}$$

By considering the gross external work, a displacement function that is unchanged to that of the unperforated plate, and by simplifying the shape of the force redistribution function $f(y)$, the elastic buckling capacity of a perforated plate is determined as a 'modified' energy prediction (MEP) as mathematically it is not a 'true' minimum energy potential.

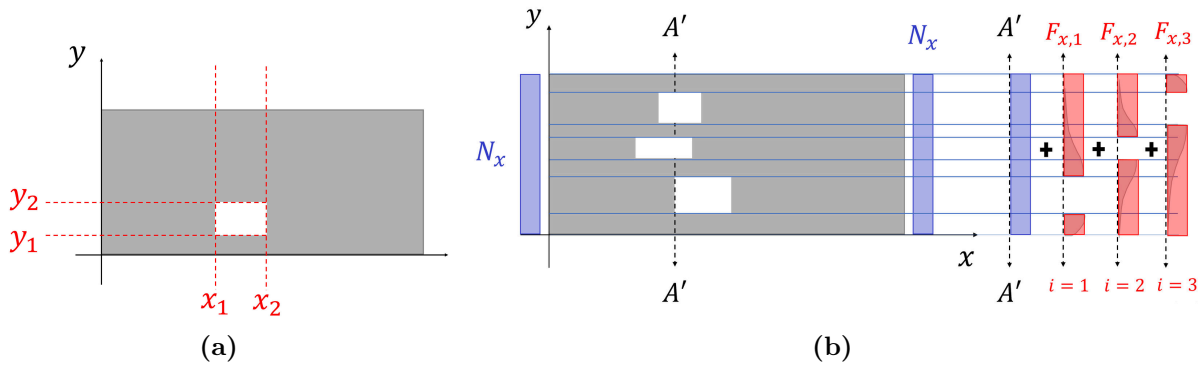


Figure 3.9: (a) Nomenclature of perforation co-ordinates, and (b) approximate uniform force function $F_{x,i}$ shown in red for each of the i holes

3.6 Discussion and Conclusion

The elastic buckling capacity determined in ABAQUS ($N_{cr,FE}$) for all 486 regularly and 320 irregularly perforated parametric models is compared to the predictions given by [4] and the MEP ($N_{cr,S}$ and $\widehat{N}_{cr,h}$ respectively, Figure 3.10). The ratio of the FE result to that predicted for the perforated plate elastic buckling capacity is given on the y-axis.

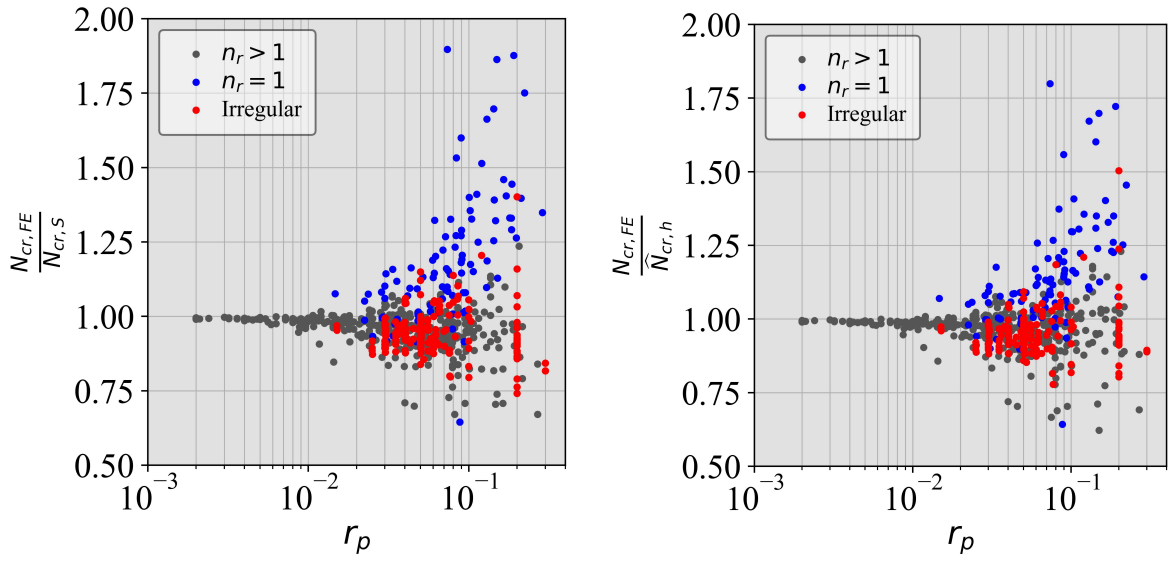


Figure 3.10: Accuracy of (a) the prediction $N_{cr,S}$ by [4], and (b) the Modified Energy Prediction $\widehat{N}_{cr,h}$ in comparison to ABAQUS results for all 806 parametric models

The predictions are of similar accuracies for all models with means of 0.981 and 0.985 for the prediction given by [4] and the MEP respectively; there is no relationship between the accuracy of either prediction and the aspect ratio of the plate, or the perforation ratio when this exceeds approximately 0.01. Both predictions are highly accurate for very small perforation ratios, and where there is very little difference in the perforated plate buckling modeshape to that of the unperforated plate (for example, models (a)-(f), Figure 3.5). This is expected for the MEP, where the modeshape is considered unchanged (Equation 3.2). The capacity of models (c), (d) and (f) in Figure 3.7 are very accurately predicted with the MEP, with accuracy ratios of 1.001, 1.002 and 0.999 respectively, compared to 0.974, 0.978 and 0.974 found with the prediction by [4]. When the average centroid of all perforations is approximately equal to the location of the antinode, and whole plate buckling manifests, the MEP may enable a more accurate prediction than that determined by [4]. Both predictions are particularly unconservative for plates with holes of a large aspect ratio, where the elastic buckling capacity of the perforated plate is almost half of the unperforated plate (for example, models (i), (l), (n) and (r), Figure 3.5), and (l) (Figure 3.7). Both predictions are overconservative for plates with one row of particularly tall perforations (for example, models (g), (h) and (j), Figure 3.5), where the elastic buckling capacity of the perforated plate is almost 50% greater than the unperforated plate. At both these extremes, the modeshape greatly deviates from whole plate buckling. However, the underprediction of capacity in the latter examples is less extreme for MEP than it is with [4]. Overall, the MEP exhibits less variation in accuracy with a smaller standard deviation of 0.109 compared to 0.126 for the prediction given by [4]. In response to the first objective of this paper, it is demonstrated that the straightforward formulation determined by [4] is of good accuracy for a wider range of perforations arrangement than indicated in their paper and is less computationally demanding than FE procedure. The accuracy and standard deviation of the MEP is very similar in most of the domain explored. Yet, despite some improvement in the predictions for plates for the cases identified, there is insufficient benefit compared to [4]. However, it is encouraging that a basic uniform approximation to the force redistribution function can enable such accurate predictions to the elastic buckling capacity. It is expected that as this function approaches the true

nature of the inhomogeneous force flow, the accuracy of the MEP will be improved, albeit at the compromise of increased computational demand. Furthermore, the combination of parameters that determine which buckling manifestation will dominate remain to be determined (Figure 3.1). Given this information, the MEP may be further improved when a more accurate approximation to the deflection modeshape is chosen.

4

Local Buckling Load of a Perforated Plate, a Computational Study

4.1 Commentary

The first issue presented in Section 1.2 surmises that design methods should be semi-empirical and therefore simple to employ and verify. Although most authors undertake parametric assessments to find these empirical relations (Section 2.1.1), none of the research from the last half-century has been successfully implemented in Eurocode for the design of perforated plates [18]. The simple empirical expression for elastic buckling load proposed by [4] was shown to perform with a similar accuracy to the MEP developed in the previous chapter, despite it being applied outside of the domain for which it was developed. Yet, Chapter 3 demonstrated that both the MEP and $N_{cr,S}$ [4] are unconservative for plates with holes of a large aspect ratio, and overconservative for plates with one row of particularly tall perforations.

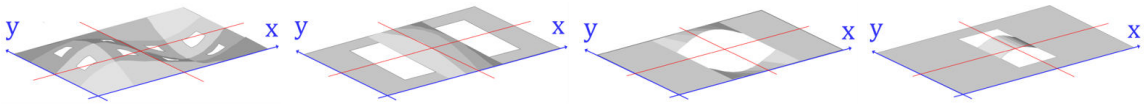
However, when the buckling modeshape is similar to the unperforated case (i.e., whole plate buckling, Figure 3.1), the MEP predicts the elastic buckling load with great accuracy. To use the MEP reliably is to know when WPB will occur, which must depend on various geometric parameters. Measuring these, this chapter categorises the four modes of local buckling (introduced in Section 2.2), and a fifth interaction mode. For each mode, a predicting equation for the critical elastic buckling load is proposed and compared

to the critical elastic buckling load determined with the finite element (FE) method. The mean ratio of the elastic buckling load for a perforated plate found with FE to that predicted with the proposed formulae is 1.00 for 775 models, with an overall R^2 coefficient of 0.840. Finally, a reliability assessment determines a standard-appealing statistical model factor γ for a 95% confidence interval. This factor is found for each local buckling mode, and proposed as 1.17 for all plates with any number of perforations, not exhibiting lateral perforation buckling (LPB).

4.2 Introduction

In structural engineering, buckling is the sudden deformation (bifurcation) of a component (e.g. plate buckling) under compressive load. The deflection modeshape is characterised by buckling waves; the wavelength is dependent on the type of buckling, which in turn depends on the component's geometry and boundary conditions. For a column with a given cross-section, Euler buckling depends solely on the column buckling length. Conversely, plates are susceptible to local buckling which depends on the length to width (aspect) ratio. Often thin-walled section columns fail through an interaction of several buckling modes [2, 76, 104] such as local buckling, distortional buckling, global (flexural or torsional) buckling or any combination of those. Additionally, perforations cause changes to the stress distribution within a thin plate. Subsequently, local buckling of a plate with a series of perforations can arise in one of four modes, and a fifth interaction mode. These modes are defined as Whole Plate Buckling (WPB), Lateral Perforation Buckling (LPB), Unstiffened Strip Buckling (USB), and Euler Strip Buckling (ESB); they depend on the number, size, and location of perforations (Figure 4.1). The nomenclature for these buckling modes is available in various publications [2, 3, 42] although diagrams and dimensional criteria specifying the causes of each mode are often vague or absent. The relationship between the perforation dimensions and the elastic buckling load of a perforated plate depends on how buckling manifests; this influences whether the elastic buckling load may be greater or less than the elastic buckling load of an unperforated plate. As the elastic buckling load and modeshape are inter-related, this chapter establishes an algorithm to

determine the perforated plate local buckling mode, and a predicting equation for the elastic buckling load is proposed for each corresponding mode.



(a) Whole Plate Buckling (WPB) **(b)** Lateral Perforation Buckling (LPB) **(c)** Unstiffened Strip Buckling (USB) **(d)** Euler Strip Buckling (ESB)

Figure 4.1: Local buckling modes of perforated plates

4.3 Research Objective and Hypotheses

This research evaluates the buckling of plates with an array of rectangular and square perforations. The differential equation of the deflection surface is derived on the assumption of small deflections, and that the stress condition is constant throughout the plate, which is undergoing uniform bending [8]. According to this theory, the bending stresses within a buckling plate of uniform thickness and material properties are a function of the deflection function $w = f(x, y)$:

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{p(x,y)}{D} \tag{4.1}$$

When holes are introduced into the plate, the bending stress now depends on a superposition of curvature and the equilibrating moment at the hole periphery [8]. Therefore, the Föppl-von Kármán equations for a variable stress field should be considered (written in terms of the Airy stress function ϕ, and where q represents any out of plane loading [45]):

$$\nabla^4 \phi = Et\left(\frac{\partial^2 w}{\partial x \partial y} - \frac{\partial^2 w}{\partial x^2} \cdot \frac{\partial^2 w}{\partial y^2}\right), \tag{4.2a}$$

$$D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = q + \left(\frac{\partial^2 \phi}{\partial y^2} \cdot \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 \phi}{\partial x^2} \cdot \frac{\partial^2 w}{\partial y^2} - 2\frac{\partial^2 \phi}{\partial x \partial y} \cdot \frac{\partial^2 w}{\partial x \partial y}\right) \tag{4.2b}$$

However, as the number and size of perforations increase, the stress state becomes incredibly complex. This causes localised variations in the deflection function w (as the curvature and bending stresses are coupled), such that the linearised expressions

for the curvature are invalidated. Thus, an analytical solution for the elastic buckling load may be impossible to find.

This necessitates the development of empirical expressions for the elastic buckling load ($N_{cr,h}$), which is the research objective of this chapter. This superposition of stress states is used to frame a hypothesis for the relationship between the elastic buckling load and the height of the perforations. For particularly small perforations, the influence on the bending stress state is negligible, (Equation 4.1) is reasonably valid, and the elastic buckling load will tend towards that of the unperforated counterpart. However, for a small-medium sized hole located near the plate centre, the superposition of stresses occurs at the most onerous location, as the hole border is located at an area of greater curvature. This may be why the elastic buckling load reduces when a small hole is located at the buckling maximum [46, 54]. Conversely, the effect is stabilising for a tall concentric hole, as the superposition occurs in a region of smaller curvature, and stresses are diverted to the supports. It is hypothesised that tall perforations will enable the greatest increase to the elastic buckling load.

4.4 Finite Element Model

To determine the nature of the relationship between the critical elastic buckling load of a perforated simply supported thin plate and the parameters discussed, performance is investigated with the finite element (FE) method. This is determined with the ABAQUS linear perturbation procedure [122], which solves the eigenvalue problem of the constitutive and geometric stiffness matrices.

4.4.1 Modelling Conditions

The material is steel with a Young's Modulus of 203.4 GPa, and a Poisson's ratio of 0.3. It is assumed to have isotropic properties and obey the stress-strain relationships for isotropic materials. The perforated plate buckles in the elastic regime, with slenderness determined in accordance with [4] and [34]. The following limit ensures the plate is slender for all variables investigated in terms of the plate width (b) and thickness (t):

$$\min\left(\frac{b}{t}\right) = 100 \tag{4.3}$$

As the plate thickness is very small in comparison to the plate dimensions, the problem is adequately described by Kirchhoff's shell theory, and thin shell quadratic elements with reduced integration (S8R5) are chosen. A structured mesh is used to negate transverse shear locking and allow for easier data management and thus faster convergence. All perforations are oriented with their major and minor dimensions parallel to the axes of the global co-ordinate plane, which allows for rectangular FE elements everywhere. A limit is placed on the aspect ratio of elements in the mesh to further ensure accurate convergence, according to an investigation by [77].

$$\min\left(\frac{a_e}{b_e}\right) = 1, \tag{4.4a}$$

$$\max\left(\frac{a_e}{b_e}\right) = 1.5 \tag{4.4b}$$

where a_e and b_e represent the length and width of the finite element respectively. Simply supported boundary conditions are represented as shown in (Figure 4.2). This is similar to the conditions applied by [35] who examine a symmetrical distribution of circular perforations arranged along the length. Compressive uniform edge loading is simulated by consistent nodal loads, as considered by [2]. ABAQUS input files are generated with a custom-built python script where these meshing, loading and boundary conditions are prescribed to the model (Appendix B). The FE model has been used to reproduce the results produced by [2] who examined 32 FE models of plates with rectangular holes with a corner radius. Despite the holes in this study being sharp-cornered, the results accurately follow the same trend. The limit on the maximum element size is chosen as a consequence of this comparison, defined as:

$$\max\left(\frac{b_e}{b}\right) = 0.02 \tag{4.5}$$

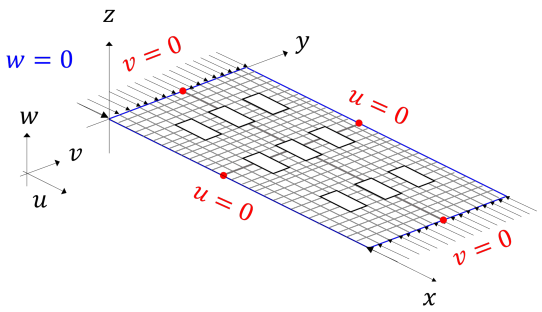


Figure 4.2: Simply supported boundary conditions and nodal loading simulated in ABAQUS

4.4.2 Parametric Assessment

The parametric analysis follows [26].

Table 4.1: Domain of variables examined in parametric study (perforation variables defined in Figure 1.4)

	a/b	b/t	n_r	n_c	S_{xo}/a	S_{yo}/b	L_{xo}/a	L_{yo}/b	s_x/a	s_y/b	l_x/a	l_y/b	e_x	e_y
Mean	2.62	100	2.00	9.00	0.163	0.183	0.495	0.426	0.060	0.252	0.051	0.210	0.089	0.104
Minimum	1.00	100	1.00	2.00	0.045	0.40	0.032	0.047	0.005	0.015	0.004	0.020	0	0
Maximum	5.32	100	5.00	30.0	0.410	0.461	0.907	0.890	0.710	0.890	0.347	0.890	0.360	0.380

A total of 870 parametric models with regularly spaced perforations has been generated such that the variables are randomly selected from the domain using the python 'randrange' function [123], and the elastic buckling load is determined by FE analysis with ABAQUS. For each of these models, the ratio of the perforated plate buckling load ($N_{cr,h}$) to the unperforated plate buckling load (N_0) (termed the 'load ratio') is plotted with respect to the perforation ratio (r_p, Section 1.3.2) (Figure 4.3). It ranges from 0.204 to 1.62 with considerable scatter. A Gaussian distribution of these results is shown for a histogram of the results with 50 bins, in Figure 4.4; the standard deviation (S.D) of ± 0.164 is shown by the dotted lines. Of these 870 models, 14.8% have a load ratio greater than one, and thus exhibit an increased elastic buckling load when perforated to when unperforated.

Although the points highlighted in Figure 4.3 correspond to a similar perforation ratio r_p, the buckling mode and buckling load varies considerably. Example A exhibits LPB, and the greatest increase to the buckling load, whereas G exhibits USB, and the greatest reduction to the buckling load (Figure 4.3). The buckling modeshape of example E and F exhibit a strong resemblance to the unperforated plate, demonstrating WPB. As the

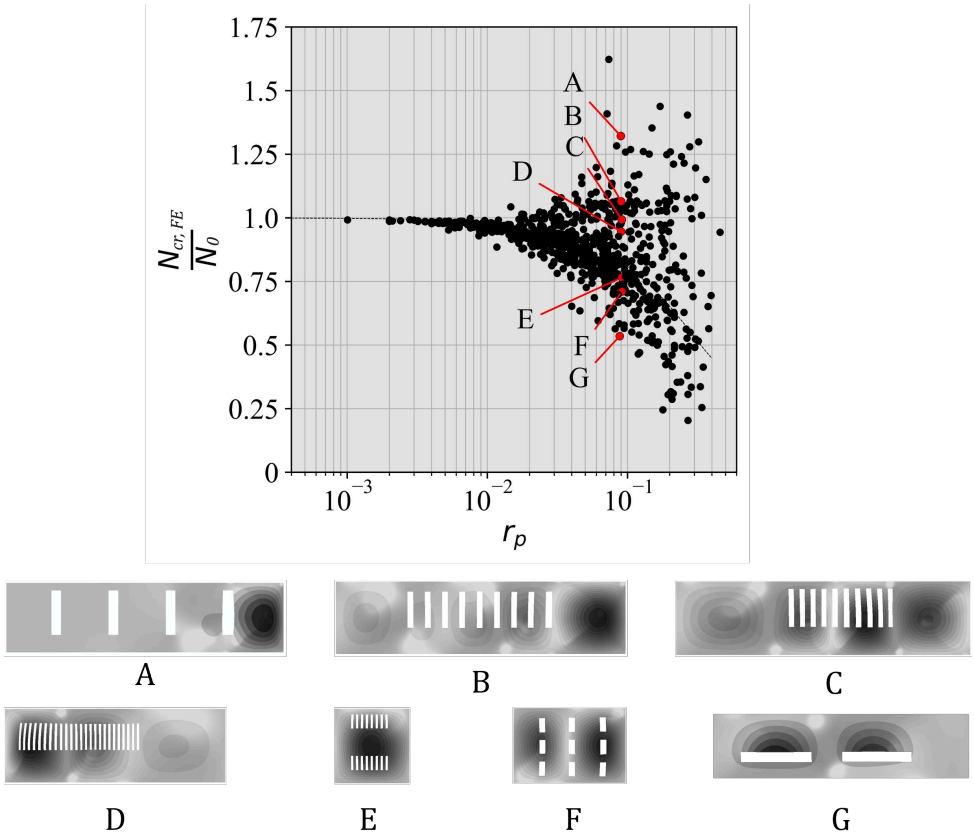


Figure 4.3: The load ratio of the perforated plate to the unperforated plate with respect to perforation ratio for all 870 parametric models

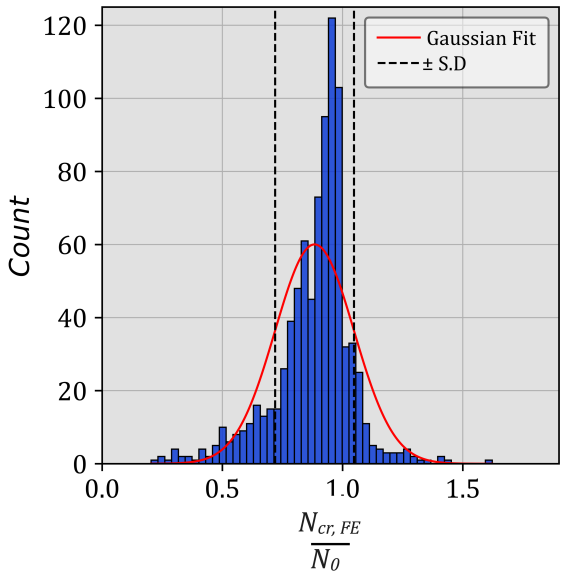


Figure 4.4: Distribution of load ratio for all 870 models in the dataset

wavelength and amplitude of the buckles in B and C exhibit small variation throughout the plate, examples B and C are considered as 'weakly WPB'. Such behaviour is on the border of what is termed an 'interaction' (INT) of buckling modes (in this case a WPB-LPB interaction); example D does not portray a strong resemblance to any of the four buckling types and is classified as such. For the r_p selected, none of the models exhibits ESB, and examples are given in the following section. As a plate with a given perforation ratio can exhibit a range of buckling modes, it follows that empirical equations for the elastic buckling loads must be established with alternative variables.

4.5 Local Buckling Mode Algorithm

A perforated plate exhibiting Whole Plate Buckling (WPB) shows very little difference in modeshape to its unperforated counterpart (Figure 4.5(a)). This is typical where the influence of perforations on the stress field is quite small. In comparison to an unperforated plate, a plate with a random distribution of perforations exhibiting WPB typically exhibits a reduced elastic buckling; primarily because the perforations are quite small, so the reduction to the plate bending stiffness dominates over the beneficial effect of stress redirection to the supported edges (Section 4.3). A perforated plate exhibits little difference in behaviour to an equivalent unperforated plate when the total perforated area is particularly small, this limit is approximately $r_p = 0.014$ (Figure 4.3) and therefore must exhibit WPB by definition. If $a_r > 1$, the WPB mode will interact with USB or ESB, as the increase in the aspect ratio of areas above and below the perforation causes a localised unstiffened plate behaviour. If, however, the aspect ratio of the perforation is less than one, with a particularly large height ratio, the WPB mode may interact with LPB, as the beneficial effect of stress redirection to the supported edges becomes dominant. This effect also dominates if the average eccentricity of the perforations is particularly large. The WPB buckling load of a plate is particularly sensitive to e_x, as the plate behaviour greatly depends on how far perforations exceed into the 'end panel' (Figure 2.4). If $e_x < 0.15$, a plate will buckle by WPB without interacting with any other modes (Figure 4.5(a)).

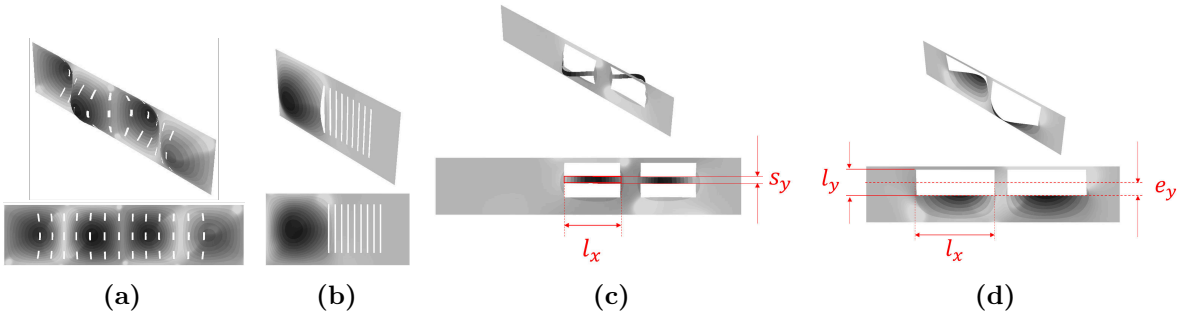


Figure 4.5: (a) WPB involves the net plate area (b) LPB at the lateral portion between the perforation and plate edge (c) ESB at the transverse portions between perforations (d) USB at the transverse portion between the perforation and plate edge.

Of the four buckling modes, a plate undergoing LPB is more likely to exhibit an increase in elastic buckling load to the unperforated state. LPB is typical of plates with one row of particularly tall perforations ($l_y > 0.6b$), with $4e_y < 0.1$ to negate interaction with other modes (Figure 4.5(b)). USB may occur instead of LPB, depending on the aspect ratio of the portion above and below the perforation which can be summarised by the ratio $\frac{l_x e_y}{b}$. Interaction with the WPB mode dominates for a row of holes with $0.5b < l_y < 0.6b$, and $e_y \geq 0.1$. There is a redistribution of forces to the plate edges, and the portions between the hole and the plate edge undergo a great increase in stiffness due to the greater force concentration, resulting in a pinning behaviour; buckling is 'pushed' into the sections laterally adjacent to the perforations.

ESB is best described as column-like plate behaviour (Figure 4.5(c)). For it to happen, there must be $n_r > 1$ and $\frac{l_x}{s_y} > 1$. However, there will be an ESB-USB interaction if $\frac{l_x}{S_{yo}} > 1$. USB is akin to localised unstiffened plate behaviour, typical of plates with a single row of large perforations (Figure 4.5(d)). It manifests when $S_{yo,i}$ is particularly large; however, this variable cannot adequately describe USB behaviour which is highly sensitive to e_y. For example, a concentric line of perforations may possess a large $S_{yo,i}$ but is more likely to exhibit WPB or a WPB-USB interaction. The condition $\frac{l_y e_y}{b} \geq 0.027$ encapsulates these behaviours. Although USB is typical when $\frac{l_x}{l_y} \geq 1$, it can occur instead of LPB for perforations with $\frac{l_x}{l_y} < 1$ as described. In comparison to an unperforated plate, the former case causes a reduction in elastic buckling load, whereas the latter case may enable it to increase. Therefore, the limits for the USB domain are much more complex than other local perforated plate buckling modes.

The numerical limits given for the various conditions discussed were determined iteratively. For example, the limit $\frac{l_y e_y}{b}$ was varied between 0.025 and 0.035 for a collection of models, which were sorted into one of the five mode subsets. The final limit of 0.027 optimised the number of correctly identified deformation modes, while minimising the number of incorrectly identified models. This process determined numerical limits on a plate's dimensions, establishing an algorithm to identify local plate buckling modes.

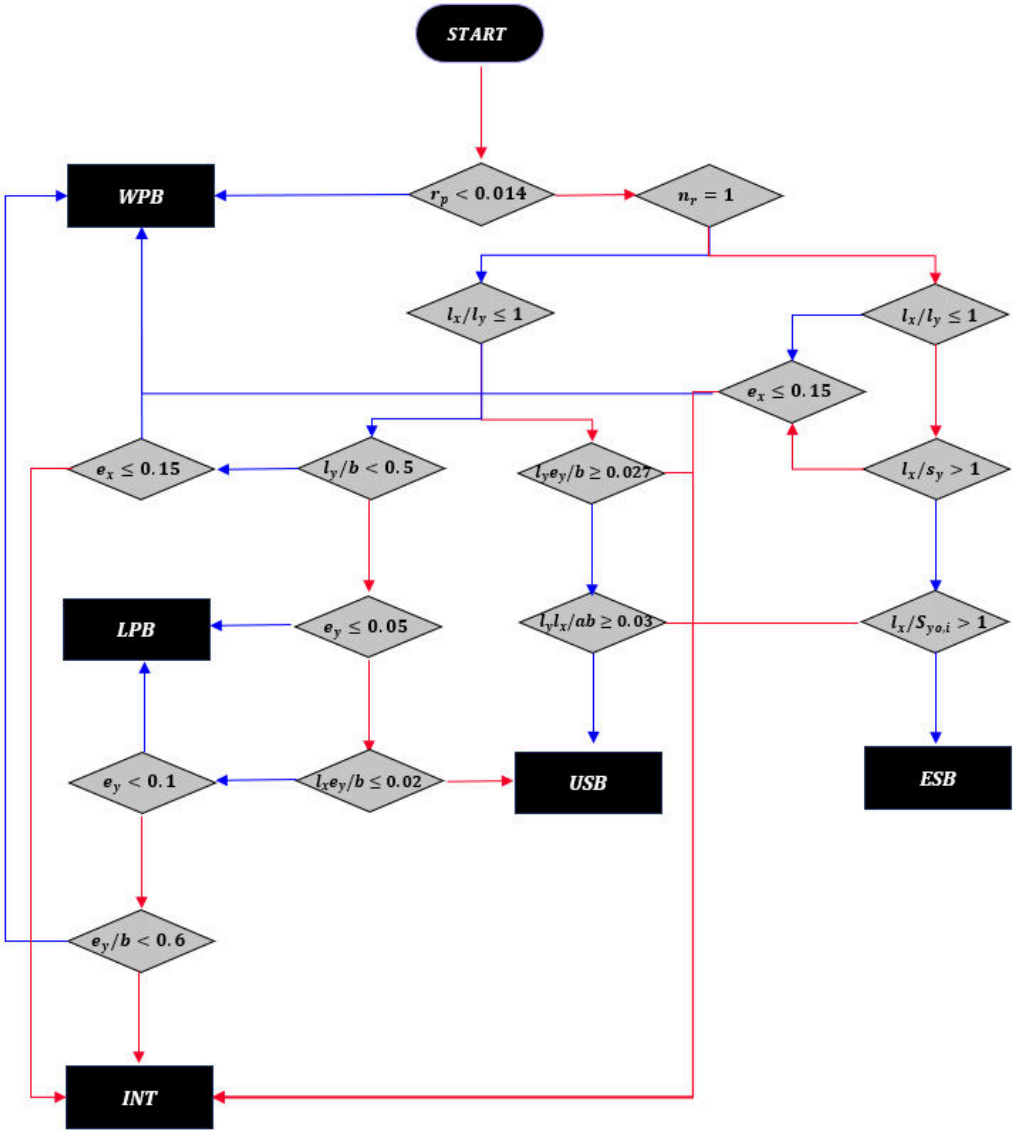


Figure 4.6: The algorithm used to determine the buckling mode corresponding to the geometry of a plate with an array of rectangular perforations. 'Yes' and 'No' are blue and red respectively. The perforation variables are defined in Figure 1.4

The performance of the resulting algorithm shown in Figure 4.6 was compared to a logistic regression model developed with machine learning using python's scikit-learn module [121]. To mitigate training on a large proportion of incorrectly classified plates, this model was trained on half of all 870 results determined with the algorithm. The random seed (i.e., collection of results used to train the model) was changed 50 times and the predictions for the test set were output each of the 50 times. Then the predicted result was compared to the 'true' result determined with the algorithm (Figure 4.6). A given result will therefore have x number of predictions based on the collection of test sets it appears in. If the elastic buckling mode predicted with the algorithm disagrees to the mode predicted with logistic regression for more than 25% of these x results, then it is assumed that the model was misclassified with the algorithm. By this assessment, only 2.76% of all parametric results were misclassified (i.e., the algorithm is very good at identifying the correct mode).

4.6 New Prediction Method

When the algorithm is applied to the dataset of 870 models, the number of models identified for the WPB, ESB, USB, and LPB subsets are 451, 48, 46 and 95 respectively; the remaining 230 models exhibit an interaction (INT) of at least two modes. The prediction developed by [4] depends solely on the unperforated plate buckling load N_0 and the perforation ratio r_p. However, we can observe that the relationship between the elastic buckling load ratio found with finite element analysis ($\frac{N_{cr,FE}}{N_0}$) and r_p is starkly different for plates with $n_r > 1$ (Figure 4.7) compared to that for plates with $n_r = 1$ (Figure 4.8). In particular, plates with $n_r > 1$ with ESB deformation, and plates with $n_r = 1$ with LPB deformation, possess a greater deviation from the general trend with respect to r_p.

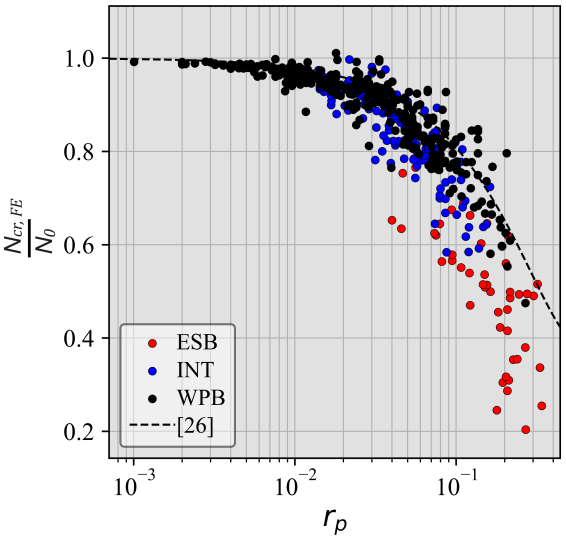


Figure 4.7: Relationship between elastic buckling load ratio and perforation ratio (r_p) for plates with $n_r > 1$.
Please note that these figures are taken from [23]; the reference to [26] in the legend is [4] in this thesis

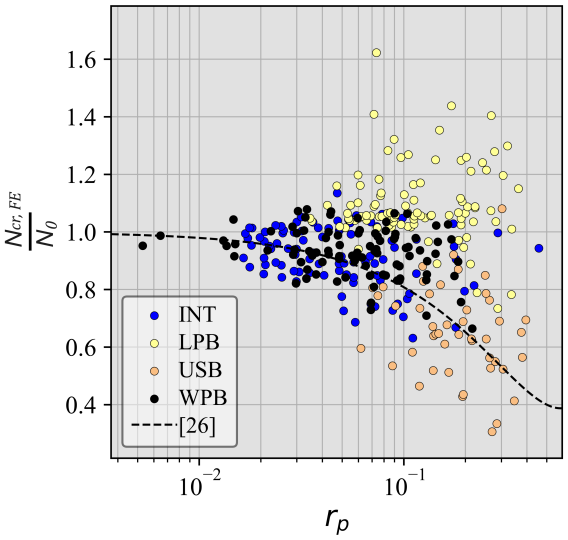


Figure 4.8: Relationship between elastic buckling load ratio and perforation ratio (r_p) for plates with $n_r = 1$
Please note that these figures are taken from [23]; the reference to [26] in the legend is [4] in this thesis

These observations directed a study to establish empirical predictions for the elastic buckling load ratio $Y = N_{FE}/N_0$ as a function $Y = f(X)$, where $X = g(x_i)$, expressed in terms of dimensionless parameters x_i, These parameters are related to the geometry of the perforations within the plate (summarised in Figure 1.4). The method to establish

the dominant parameters and numerical coefficients followed a hybrid approach termed 'machine-learning-directed-regression-analysis'. This approach was adopted to identify a compact set of influential parameters x_i. First, candidate geometric and material variables were screened using multiple linear regression, with statistical significance assessed using the associated p-values. Variables significant at the 5% level were retained as dominant predictors.

Subsequently, an expression found using direct regression on all raw dimensional variables was avoided, since the resulting equations would be unit-dependent, lengthy, and difficult to interpret. Although nondimensional groups generally provide a more transferable basis, a full dimensional-analysis on the many geometric variables would generate a large number of possible combinations. Similarly, black-box supervised learning methods, such as support vector regression, were not adopted because they provide limited physical transparency. Instead, the statistically significant variables were used to guide the construction of a physically motivated nondimensional parameter $X = g(x_i)$, which was then calibrated by regression against the numerical results.

As the elastic buckling load ratio depends on whether the number of rows of perforations is equal to or greater than one, a prediction is established for each of these subsets. The functions $g(x_i)$ are τ and ω for plates with $n_r > 1$ and $n_r = 1$ respectively. The chosen expression must meet three objectives. Firstly, it must be accurate for the 'target' subset of plates, defined in the next sections. Secondly, it must be accurate for a distinct mode, identified with the algorithm. This 'accuracy' is determined by the R^2 test result, which is used to evaluate how well the proposed regression model fits the data. The third objective is that it must be simple; it is desired to minimise the number n of constants and variables x_i where i varies from 1 to n. The scipy.optimize.minimize function [124] was used to determine the coefficients a_i with least squares regression.

4.6.1 An Expression for plates with $n_r > 1$

In this subset, 360 and 48 models are identified to exhibit WPB and ESB respectively; the remaining 135 models exhibit an interaction (INT) of at least two modes. It was discussed that ESB demonstrates a greater deviation from the general trend with respect to r_p than

WPB and INTeraction modes for plates with $n_r > 1$. Regardless, it will be shown that the proposed variable τ can be used to establish the elastic buckling load ratio for all plates with $n_r > 1$, and thus the target subset. Machine learning was not necessary for these plates, as r_p already provides a very good starting point for the choice of dominant variables (Figure 4.7). A series of potential equations were used to predict the elastic buckling load of plates with more than one row of perforations. The proposed options for $f(X)$ and $g(x_i)$ are shown in Table 4.2, and determined by iteratively modifying $g(x_i)$ according to the observed performance. This performance is described by the R^2 test result, given for each option and mode in Table 4.3. Note that the results are determined for non-rounded coefficients a_i, which are given in Table 4.2 to three significant figures. The expression proposed by [4] is given in the first row of Table 4.2, and the performance is given in Table 4.3, in addition to that of the predictions by [2] and [3].

Table 4.2: Functions $f(X)$ and $g(x_i)$ where i takes values from 1 to n for plates with $n_r > 1$

Option	$f(X)$	$X = g(x_i)$	n
0[4]	$1.8X^2 - 2.1X + 1$	r_p	1
1	$e^{-3.16X}$	r_p	1
2	$e^{-1.15X}$	$r_p \cdot \frac{a}{b}$	3
3	$e^{-4.96X}$	$r_p \left(0.287\frac{S_{yo}}{b} + 0.594\frac{l_y}{b} + 1.52\frac{l_x}{a} + 0.0802\frac{a}{b} + 1.73e_x + 0.718e_y\right)$	8
4	$e^{-1.96X}$	$r_p \left((\frac{S_{yo}}{b})^{5.04} + (\frac{l_y}{b})^{3.43} + (\frac{l_x}{a})^{0.546} + (\frac{a}{b})^{0.140} + e_x^{0.440} + e_y^{1.71}\right)$	8
5	$e^{-0.714X}$	$r_p \left((\frac{S_{yo}}{b})^{-0.407} + (\frac{l_y}{b})^{-0.274} + (\frac{l_x}{a})^{0.452}\right)$	6
6	$e^{-2.04X}$	$r_p \left((\frac{a}{b})^{0.230} + e_x^{0.360} + e_y^{1.33}\right)$	5

All of the proposed options exceed the performance of the prediction proposed by [4], [2] and [3] for the entire target subset. This is unanticipated given that the parametric domain examined by [3] is similar to that examined here (Table 4.1). Option 5 in Table 4.3 is assessed to best meet each of the desired objectives. Therefore, the elastic buckling load of plates with $n_r > 1$ is found with the following equation, where variables are

Table 4.3: R^2 test performance for plates with $n_r > 1$, assessed for # models exhibiting each mode

Option	R^2	R^2_{WPB}	R^2_{INT}	R^2_{ESB}
[2]	-1.69	- 3.96	-2.95	-8.30
[3]	0.416	0.607	0.023	-3.32
0 [4]	0.711	0.848	0.523	-1.31
1	0.822	0.734	0.724	0.071
2	0.719	0.452	0.210	0.313
3	0.877	0.767	0.768	0.539
4	0.880	0.815	0.781	0.429
5	0.871	0.843	0.852	0.193
6	0.861	0.785	0.730	0.356
# Models	543	360	135	48

defined in (Figure 1.4):

$$\tau = r_p \cdot \left(\left(\frac{S_{yo}}{b} \right)^{-0.407} + \left(\frac{l_y}{b} \right)^{-0.274} + \left(\frac{l_x}{a} \right)^{0.452} \right), \tag{4.6a}$$

$$\widehat{N}_{cr,h,n_r>1} = N_0 e^{-0.714\tau} \tag{4.6b}$$

The elastic buckling load determined with Equation 4.6 for plates with $n_r > 1$ is plotted with respect to the variable τ in Figure 4.9.

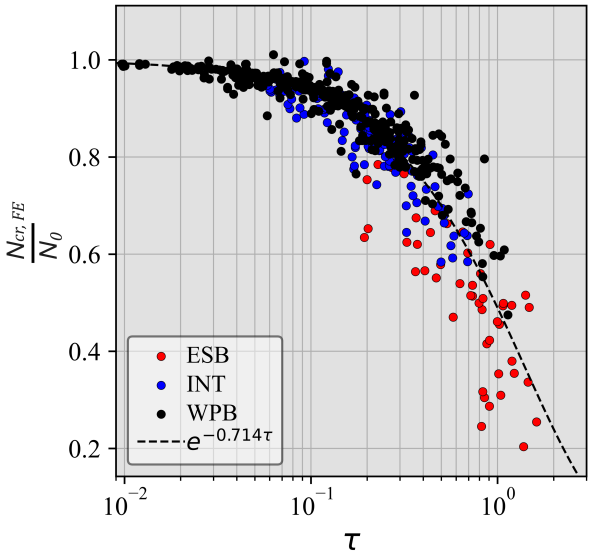


Figure 4.9: Relationship between the load ratio and τ for plates with $n_r > 1$

Table 4.3 also provides the R^2 test result for plates exhibiting ESB, which is found to be particularly poor. ESB behaviour is characterised by local buckling at the portion of

the plate between perforations. Thus, assuming that $l_x \gg s_y$ and that the full length l_x is effective, one can posit that $N_{cr,ESB}$ per unit width could be approximated as:

$$N_{cr,ESB} = \frac{E\pi^2 t^3 (b - l_y n_r)}{12 b l_x^2} \tag{4.7}$$

However, when used to determine the elastic buckling load ratio, an R^2 test score of -40.8 is obtained for Equation 4.7 (i.e., inferior to the score determined with the new expression proposed in Equation 4.6). It follows that Equation 4.6 should be used to determine $N_{cr,ESB}$, despite the fact that the R^2 test result for plates exhibiting ESB is much poorer than WPB and INTeraction modes.

4.6.2 An Expression for plates with $n_r = 1$

In this subset, the number of models identified for the WPB, USB, LPB, subsets are 91, 46 and 95 respectively; the remaining 95 models exhibit an interaction (INT) of at least two modes. On observation and mathematical analysis, plates with $n_r = 1$ deforming by LPB exhibit a separate behaviour to that of other buckling modes with $n_r = 1$. Therefore, only plates with $n_r = 1$ exhibiting WPB, USB or an INTeraction of modes define the target subset and are used to develop an expression for the elastic buckling load. The subset of plates with $n_r = 1$ exhibits a very poor relationship with the perforation ratio r_p (Figure 4.8 and τ Figure 4.10). When [4] and Equation 4.6 are used to determine the elastic buckling load ratio for this subset, the R^2 test score is 0.223 and -1.22 respectively. To establish a new initial function $f(X)$, machine learning was used again to determine the dominant variables. Backwards elimination was performed to build a multivariable linear regression model. The following procedure to determine functions $f(X)$ and $g(x_i)$ was performed iteratively by regression analysis, and by addressing the performance of the corresponding equation with the R^2 test. The proposed options are summarised in Table 4.4, their performance evaluations are given in Table 4.5, and the relationship with the chosen variable ω is shown in Figure 4.11.

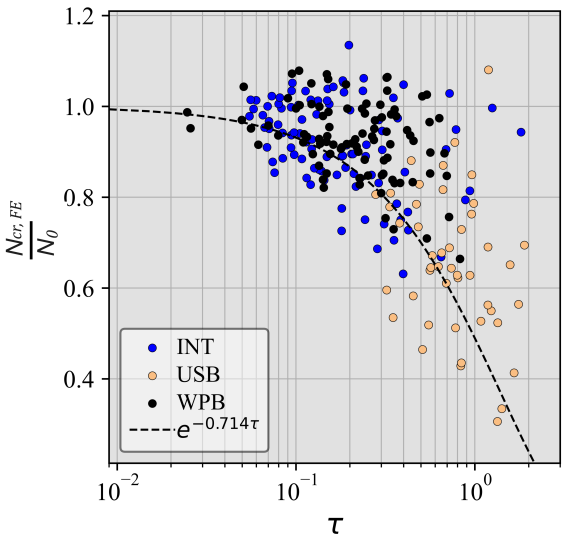


Figure 4.10: Relationship between load ratio and τ for plates exhibiting WPB, INT or USB with $n_r = 1$

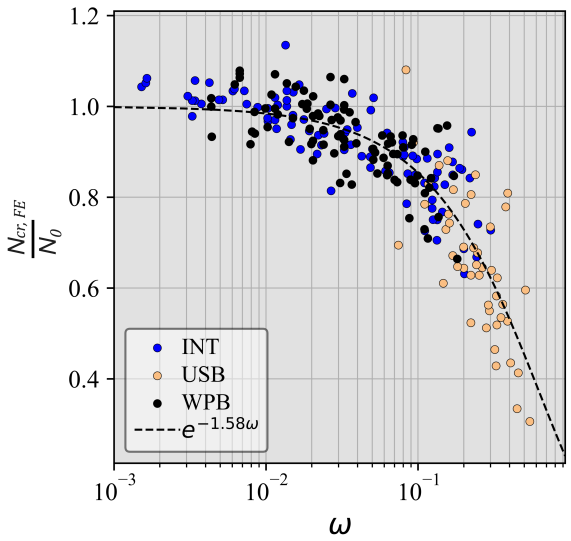


Figure 4.11: Relationship between the load ratio and ω for plates exhibiting WPB, INT or USB with $n_r = 1$

Table 4.4: Functions $f(X)$ and $g(x_i)$ where i takes values from 1 to n for plates with $n_r = 1$

Option	$f(X)$	$X = g(x_i)$	n
0[4]	$1.8X^2 - 2.1X + 1$	r_p	1
1	$e^{-1.48X}$	r_p	1
2	$0.740 + X$	$0.0199m - 0.123e_x - 0.567e_y + 0.129(\frac{S_{xo,2}}{a}) + 0.317(\frac{S_{yo,1}}{b}) + 0.298(\frac{S_{yo,2}}{b}) + 0.333(\frac{s_x}{a}) + 0.125(\frac{s_y}{b}) + 0.125(\frac{L_{yo}}{b}) - 1.50(\frac{l_x}{a}) + 0.125(\frac{l_y}{b})$	13
3	$e^{-0.0391X}$	$0.407m - 5.283e_x + 17.5e_y - 10.3(\frac{S_{xo}}{a}) + 1.22(\frac{S_{yo}}{b}) - 12.0(\frac{s_x}{a}) - 0.549(\frac{s_y}{b}) - 0.549(\frac{L_{y,0}}{b}) + 46.4(\frac{l_x}{a}) - 0.549(\frac{l_y}{b})$	12
4	e^{X}	$(0.463e_x + 1.31e_y) \cdot (-9.12\frac{l_x}{a})$	4
5	$e^{-20.4X}$	$(0.146e_x + 0.489e_y) \cdot \left((\frac{l_x}{a})^{0.891} + (\frac{l_y}{b})^{22.0}\right)$	6
6	$e^{-1.58X}$	$\left(\left(e_x + \frac{l_y}{b}\right)^{35.3} + \left(e_y^{1.08} + \frac{l_x}{a}^{0.561}\right)\right)^{2.28}$	6

Table 4.5: R^2 test performance for plates with $n_r = 1$, assessed for # models exhibiting each mode

Option	R^2	R^2_{WPB}	R^2_{INT}	R^2_{USB}
[2]	-1.79	-9.57	-5.82	-0.158
[3]	0.185	-0.327	-0.355	-0.945
0 [4]	0.223	-0.797	-0.681	-0.207
1	0.311	-0.260	-0.241	-0.489
2	0.790	0.474	0.609	0.649
3	0.791	0.569	0.626	0.576
4	0.708	0.198	0.527	0.498
5	0.711	0.248	0.513	0.494
6	0.766	0.515	0.592	0.514
# Models	232	91	95	46

Option 6 requires half the number of the variables n considered in the expressions determined with machine learning, with only a 6.58% lower performance score. It predicts the elastic buckling load ratio for plates with $n_r = 1$ as follows:

$$\omega = \left(\left(e_x + \frac{l_y}{b}\right)^{35.3} + \left(e_y^{1.08} + \left(\frac{l_x}{a}\right)^{0.561}\right)^{2.28}\right), \tag{4.8a}$$

$$\widehat{N}_{cr,h,n_r=1} = N_0 e^{-1.58\omega} \tag{4.8b}$$

4.7 Prediction Formulae for $\widehat{N}_{cr,h}$

In summary, two new equations are proposed: Equation 4.6 for plates with $n_r > 1$ and Equation 4.8 for plates with $n_r = 1$ with the exclusion of LPB. Finally, the evaluated elastic buckling loads ($\widehat{N}_{cr,h}$) are plotted against the numerically computed buckling load with FE analysis ($N_{cr,FE}$) in Figure 4.12 for all 775 models (i.e., excluding LPB models). We find a mean accuracy and COV of 1.00 and 0.093 respectively, and an overall R^2 coefficient of 0.840.

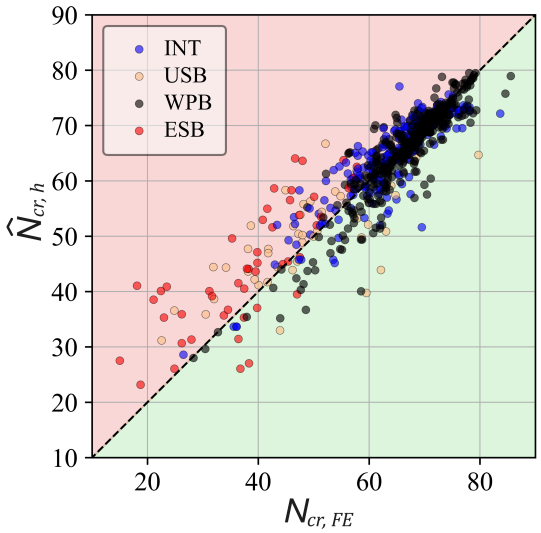


Figure 4.12: Relationship between predicted ($\widehat{N}_{cr,h}$) and FE ($N_{cr,FE}$) elastic buckling loads for all 775 considered geometries, including WPB, ESB, USB, INT, excluding LPB (points in the green region are conservative).

The proposed Equation 4.6 and Equation 4.8 exhibit the best performance for each of the subsets examined in Table 4.6, where performance is defined by the best R^2 test result. However, the performance of the ESB subset is very poor.

The models in the ESB and USB subsets have a mean ratio less than unity, and are thus unconservatively approximated. In line with Eurocode, the design elastic buckling load is defined as:

$$N_{Ed} = \frac{\widehat{N}_{cr,h}}{\gamma} \tag{4.9}$$

Table 4.6: Mean (μ), COV and R^2 test performance for the elastic buckling load of a perforated plate found with FEM to that found with various methods, for different subsets of n parametric models, and different local buckling modes (Figure 4.1)

		$n_r > 1$			$n_r = 1$		
Modes		WPB, INT, ESB	WPB,INT	ESB	WPB, INT, USB	WPB, INT	USB
[2]	μ	1.06	1.107	0.568	1.38	1.40	1.28
	COV	0.406	0.377	0.306	0.303	0.318	0.196
	R^2	-1.69	-3.27	-8.30	-1.79	-7.39	-0.158
[3]	μ	0.913	0.938	0.654	0.984	1.02	0.854
	COV	0.122	0.065	0.254	0.180	0.139	0.280
	R^2	0.416	0.470	-3.22	0.185	-0.336	-0.945
[4]	μ	0.945	0.967	0.716	1.06	1.07	1.00
	COV	0.104	0.054	0.208	0.186	0.154	0.283
	R^2	0.711	0.766	-1.31	0.223	-0.721	-0.207
Eq. 4.6	μ	1.00	1.01	0.870	N/A		
	COV	0.088	0.056	0.219			
	R^2	0.871	0.858	0.193			
Eq. 4.8	μ	N/A			1.00	1.01	0.976
	COV				0.102	0.072	0.179
	R^2				0.765	0.562	0.514
# Models		543	495	48	232	186	46

This is a lower bound approach, where the statistical model factor γ is chosen to ensure that the probability of unconservative design is limited to only α% of cases, such that:

$$P(N_{cr,FE} \geq N_{Ed}) = 1 - \alpha \tag{4.10}$$

To ensure that the actual buckling resistance determined with FE analysis ($N_{cr,FE}$) exceeds the design value (N_{Ed}) with at least a 95% probability, the model factor is found for $\alpha = 0.05$ ($\gamma_{5\%}$). It is also found for $\alpha = 0.01$ ($\gamma_{1\%}$), for a probability that 99% of predictions will be less than the actual buckling resistance. The value for γ is given in Table 7 for each local plate buckling mode. The results are displayed in Figure 4.13 for all 775 models, such that $\gamma_{WPB,5\%} = 1.06$ is applied for all 451 WPB models, $\gamma_{INT,5\%} = 1.12$

is applied for all 230 INT models, $\gamma_{USB,5\%} = 1.38$ is applied for all 46 models, and $\gamma_{ESB,5\%} = 1.83$ is applied for all 48 ESB models.

Table 4.7: The factor γ determined for each mode with n models at the 5% and 1% significance level

Mode	# Models	$\gamma_{5\%}$	$\gamma_{1\%}$
WPB	451	1.06	1.15
INT	230	1.12	1.18
USB	46	1.38	1.47
ESB	48	1.83	2.27
$n_r > 1$	543	1.15	1.54
$n_r = 1$ (not LPB)	232	1.17	1.38
All (not LPB)	775	1.17	1.41

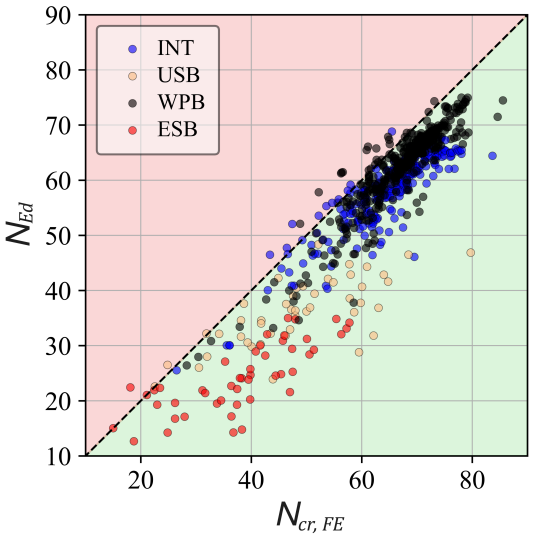


Figure 4.13: Relationship between predicted (N_{Ed}) and FE ($N_{cr,FE}$) elastic buckling loads for all 775 considered geometries, including WPB, ESB, USB, INT, excluding LPB (Figure 4.1). Points in the green region are conservative

4.8 Conclusion

This research investigates the elastic buckling behaviour of rectangular plates with a regular array of perforations. The local buckling of such plates can exhibit one of four modes: Whole Plate Buckling (WPB), Euler Strip Buckling (ESB), Unstiffened Strip Buckling (USB), Lateral Perforation Buckling (LPB) or an interaction (INT) of them. A parametric study of 870 models is first conducted to develop an algorithm for determining the buckling mode based on plate and perforation dimensions. We demonstrate that this algorithm can be reliably used to determine which of the four types of mode will dominate. At the extents of each of the four domains, interaction occurs (INT), and a particular mode is visually 'weaker'. Of the models examined in this study, a plate with one row of perforations exhibiting LPB possess the greatest average l_y/b, with a mean increase to the elastic buckling load of the unperforated plate counterpart of 12%. In comparison, plates with one row of perforations exhibiting WPB or USB exhibit a mean reduction to the elastic buckling load of the equivalent unperforated plate of 7% and 34% respectively. Plates with a regular array of square or rectangular perforations with more than one row exhibit an even greater reduction to the unperforated plate elastic buckling load, of 10% and 49% for WPB and ESB respectively. Therefore, only WPB (particularly for one row of tall perforations) and LPB may increase the buckling efficiency of a plate, where efficiency is the capacity-to-weight ratio. This is attributed to the redirection of stresses to the supported edges; similar behaviour may be seen for the perforated web of a steel section. In the absence of a procedure to determine by which mode a perforated plate would buckle, several authors have proposed methods to find the elastic buckling load of such plates. In [3], an expression for the elastic buckling load of plates with multiple rows of perforations is developed. Despite this, it performs poorly for plates with $n_r > 1$ in the parametric domain investigated here. Overall, the prediction by [4] is most accurate for plates with $n_r > 1$, in particular for plates exhibiting WPB with $n_r > 1$ ($R^2 = 0.848$). Equations to determine the local elastic buckling load for each buckling mode identified with the algorithm are then developed with the help of machine-learning-directed-regression-analysis. We establish first a new governing geometrical parameter τ for plates with $n_r > 1$ and a

corresponding equation (Equation 4.6), valid for WPB, INT and ESB; and second, a new governing geometrical parameter ω for plates with $n_r = 1$ and a corresponding equation (Equation 4.8), valid for WPB, INT and USB. Overall, the ratio of the local elastic buckling load appropriately predicted with either Equation 4.6 or Equation 4.8 to the value computed with FE has a mean of 1.00 with a COV of 0.093, with an overall R^2 coefficient of 0.840 for all 775 models We show that in addition to a high R^2 test result for the WPB mode, Equation 4.6 outperforms the equation proposed by [4] for the INT and ESB modes. Of the three modes, the R^2 accuracy of the latter is however somewhat poorer. Similarly, Equation 4.8 surpasses the R^2 test performance of current predictions (although this is less than that for plates with $n_r > 1$). Yet, the R^2 accuracy of plates with USB deformation is slightly worse than that found for the INT or WPB. Therefore, when the USB or ESB mode is identified with the algorithm, a greater model factor γ should be applied to the predicted elastic buckling load. A model factor $\gamma = 1.17$ is proposed for all buckling modes, with the exception of LPB. This was determined for $\alpha = 0.05$ such that 95% of buckling load design predictions (N_{Ed}) possess a buckling load less than the computed value ($N_{cr,FE}$), and are thus conservative. Furthermore, we conclude that the LPB buckling mode follows a distinct behaviour, and Equation 4.8 should not be used for this buckling mode; the elastic buckling load prediction for this mode should be a focus for future research. Lastly, both new geometrical parameters τ and ω are independent of the spacing between perforations. This suggests that they may be applicable to other arrangements such as staggered or irregular arrays, which will also be explored in future research.

5

An Experimental, Numerical and Analytical Study of Local Buckling in Perforated Plates

5.1 Commentary

The previous two chapters have presented comprehensive parametric studies, verified against published work. To verify both this work and studies into post-critical behaviour presented later in Chapter 6 and 7, this chapter consolidates a series of uniform buckling compression tests on simply supported perforated aluminium plates.

This investigation is the first to measure buckling deformation of a perforated plate using digital image correlation (DIC). Open-source software 'DuoDIC' [125] was used for post-processing the results, but did not involve a tool to measure the initial geometric imperfection. The images shown in Table 5.3 were generated from a script I wrote in python, which measures this imperfection, by measuring the minute difference in displacement between the first two images. Additionally, I wrote code that could translate the field of points captured by the DuoDIC triangulation into a structured FE mesh for GMNIA assessment. These python scripts are provided in Appendix C and Appendix D.

Also in Appendix E are drawings of the loading frame I designed to be manufactured. This frame was designed to be mounted in an INSTRON 5582 loading rig; the top boundary attaches to the actuator and can accommodate rotational degrees of freedom; the connection can be dismounted, and the length of the top boundary can be changed to

test plates of different widths. Four large bolts in the bottom platen secure the assembly to the loading rig, thus ensuring the whole set-up is centered about the load axis. The vertical boundary conditions can be jacked in and out to accommodate the dimensional tolerance of the plate. The assembly is designed to maximise the visible area for DIC.

After a qualitative discussion on how perforations influence performance and deformation, a linear perturbation procedure is undertaken to assess the buckling behaviour quantitively, by comparing the method proposed by [2, 115] (used by North American design standard [7]) to the method proposed by [23] (Chapter 4).

After submission to Thin-Walled Structures, this paper received three rounds of revisions, with reviewers questioning the discrepancy between the experimental and computational results near $N_{u,h}$ for some of the specimen.

The reasons for these differences are discussed in Section 5.5. In summary, the experiments have many sources of error such as poor speckle pattern and contrast, skew in the set-up, and imperfect boundary conditions. Conversely, the FE model provides an idealised representation of the physical problem, with perfectly pinned boundaries, and without modelling local effects such as the heat-affected zone surrounding the perforations. While this idealisation contributes to discrepancies near $N_{u,h}$, it also provides a consistent framework for investigating the influence of perforation geometry and plate slenderness. Thus, the FE prediction is not bespoke to these experiments, and is instead intended to capture the underlying structural response sufficiently, to generalise across the broader parametric studies presented in this thesis.

It should also be highlighted that these experimental tests were conducted on aluminium plates, whereas the majority of this thesis concerns perforated steel plates. The reason for this was a practical one; subject to the dimensional constraints of the load frame, perforated plates measuring 130 × 300 × 1 mm were manufactured for testing. For simply supported unperforated specimen with these dimensions, the elastic buckling load is 5.77kN for steel (E = 203.4 GPa, v = 0.3) and 1.74kN for aluminium (E = 60 GPa, v = 0.33). The load cell should be chosen so that expected loads fall within 10-90% of the load cell's full-scale capacity, preferably toward the upper end of this range to maximise measurement quality. The available load cells had capacities of 1 kN, 5 kN, and 100 kN, thus aluminium was

adopted to allow testing with the 5kN load cell, thereby permitting improved measurement sensitivity, resolution and accuracy for loads at and beyond elastic buckling.

Additionally, the use of aluminium does not undermine the validity of the experimental campaign, as the observed response was governed by elastic behaviour rather than material yielding or plasticity; the proportionality limit is approximately 70MPa (Figure 5.5), significantly greater than the average elastic buckling stress of 9.83MPa, and average ultimate compression stress of 18.1MPa. Furthermore, it is shown in Chapter 6 that aluminium and steel possess a similar stress-strain response, which is well predicted with the Ramberg-Osgood two-stage model. Hence, the use of aluminium is justified on practical grounds, without compromising the elastic buckling mechanics relevant to steel plates.

Finally, in Section 5.5, the point of elastic buckling was identified from the force-displacement response using a tangent stiffness criterion. The initial stiffness was determined from the approximately linear region at low load, and buckling was taken as the point where a sustained reduction from this stiffness occurred, provided it coincided with measurable growth in the out-of-plane displacement field. Due to the measured geometric imperfections, this transition was often gradual rather than a distinct bifurcation point. Therefore, the identified buckling point should be interpreted as an approximate onset of elastic buckling. The term 'point of inflection' refers to this change in tangent stiffness, rather than a strict mathematical inflection point.

5.2 Introduction

Plated structures are widely used in the construction industry as webs or flanges of members such as beams or columns, for which their edge conditions are conservatively approximated as simply-supported (internal elements) or free (outstand element). Perforations are often introduced in slender steel plates for various purposes (e.g. for services access, ease of fixations, or material savings) with minimal impact on their elastic buckling load [35], therefore increasing their structural efficiency. For instance, plates with a staggered arrangement of perforations are commonly used in structural rack sections to enable greater mounting flexibility (Figure 5.1). Members such as those depicted in Figure 5.1 are often subject to bending or a combination of compression and bending, and their elastic

buckling load is usually conservatively assessed under pure uniform compression. This approach, commonly applied to unperforated plates, is assumed to be valid for perforated plates herein. Methods for the prediction of elastic buckling load of perforated plates have been developed in recent years [2–4, 23, 26]. Most recently in [23], an algorithm to identify the local plate buckling mode for a plate with a regular array of rectangular perforations was proposed. This buckling mode may be either Whole Plate Buckling (WPB), Lateral Perforation Buckling (LPB), Unstiffened Strip Buckling (USB), Euler Strip Buckling (ESB), or an interaction of these (INT), see Figure 5.2. According to the mode identified, design formulae determining the buckling capacity were proposed [23]. However, little research examines the behaviour of such plates experimentally.

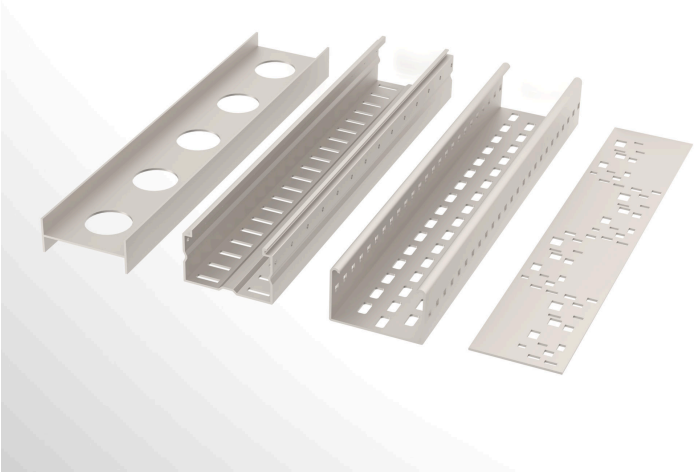


Figure 5.1: Examples of perforated plates

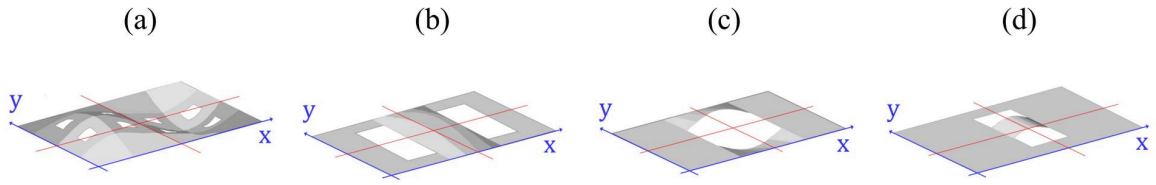
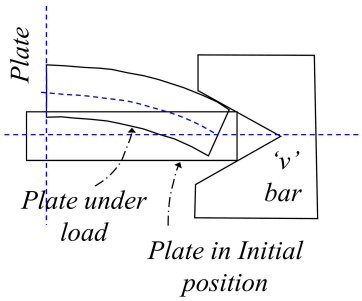


Figure 5.2: Local buckling modes of perforated plates (a) Whole Plate Buckling, WPB (b) Lateral Perforation Buckling, LPB (c) Unstiffened Strip Buckling, USB (d) Euler Strip Buckling, ESB

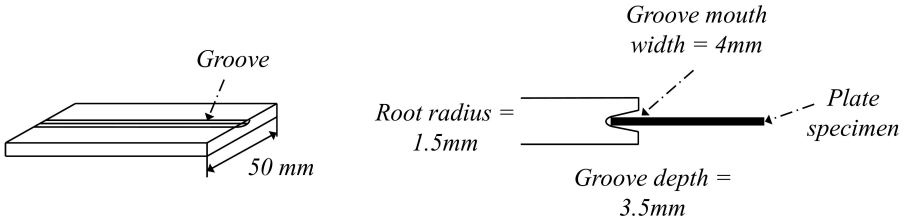
5.3 Experiment Design

5.3.1 Load Frame

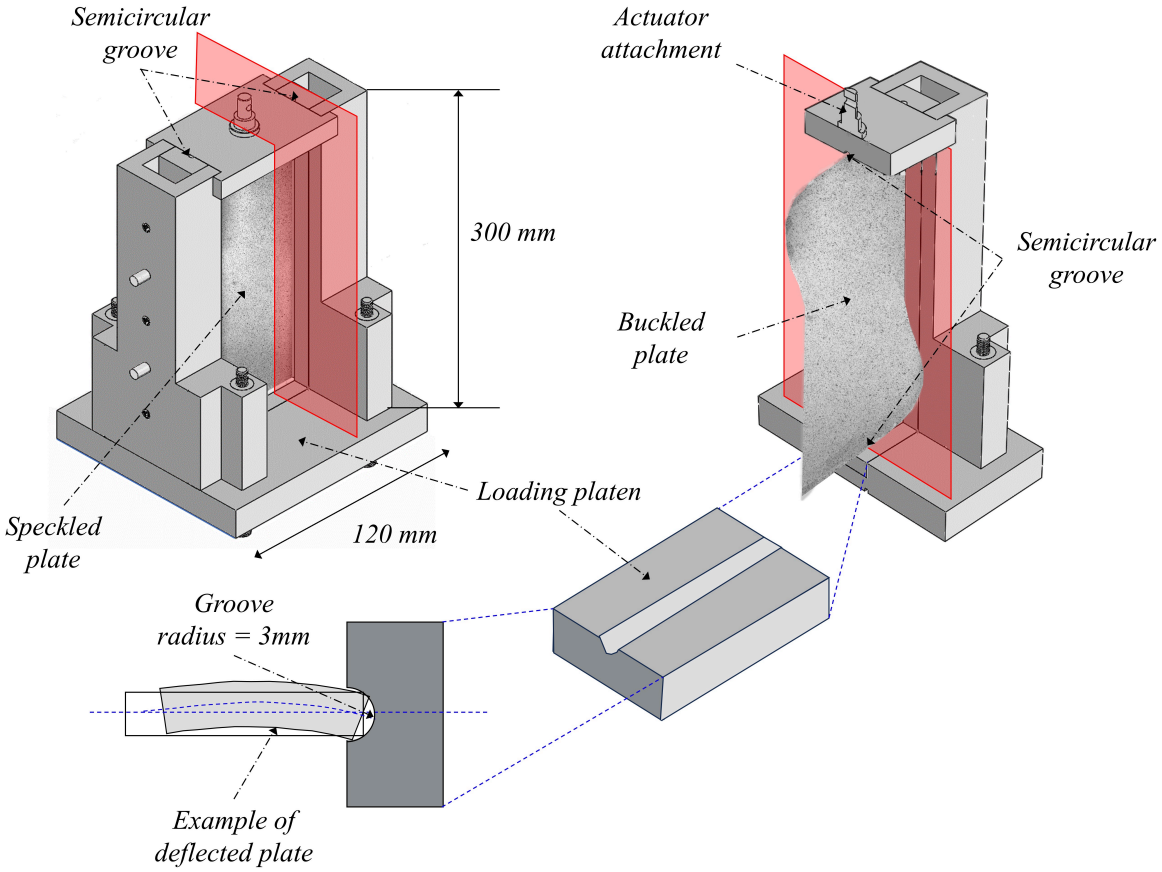
Figure 5.3 illustrate the boundary notch designs considered in this study: a V-notch (Figure 5.3(a)), an elliptical groove (Figure 5.3(b)), and the final adopted configuration, including the custom loading rig and groove geometry (Figure 5.3(c)). Finger and roller supports are prone to translational slip and may introduce local bending moments due to misalignment or uneven load transfer. Moreover, such moving parts are susceptible to friction and wear over time. Elliptical grooves and V-shaped notches, as proposed by [108] and [9], offer broader contact zones that distribute bearing stresses more uniformly, reducing local plasticity, and enhancing repeatability by minimising sensitivity to surface roughness (see Section 2.4). In the present study, 'near-pinned' boundary conditions were implemented using edge notches enabling free rotation with minimal moment restraint and ensuring consistent support along all sides of the specimen. The selected notch profile, shown in Figure 5.3(c), consists of a V-section terminating in a semi-circular relief at its base, which accommodates limited edge translation without compromising the intended rotational freedom. Importantly, the groove was designed to permit unrestricted deflection at and near to ultimate capacity.



(a) V-notch design adapted from [9]



(b) Elliptical-notch design adapted from [108]



(c) Loading rig design with groove design

Figure 5.3: Design of loading frame and grooves permitting simply supported boundary conditions

In comparison to a typical V notch support which localizes contact to a sharp apex, the semi circular profile offers a broader bearing surface while retaining hinge-like kinematics. A semi-circular groove is also easier to manufacture than an elliptical or sharp groove. It maintains self alignment under load and preserves the intended simply supported condition even if minor specimen slip or out of plane deformation occur. Friction at the interface with the groove is minimised by lubrication prior to testing. The load frame was custom-designed to be compatible with an INSTRON 5582 Universal testing Machine [126]. It consists of modular parts that can be interchanged to accommodate different boundary conditions and small variations in plate dimensions. The width of the specimen is governed by the actuator attachment, which is 120 mm in width. The actuator attachment is locked to the machine collar via a washer, while the collar itself remains free to rotate, thus supporting the implementation of a simply supported boundary condition. All components were aligned centrally with respect to the actuator collar. The geometry of the overhanging notch was optimised to minimise optical occlusion, with less than 10% of the specimen area obstructed from view. This maximised the observable area for Digital Image Correlation (DIC), which was employed to capture deformation fields. The tests were performed with a crosshead speed of 0.2mm/min, slower than the speed examined by [105], and terminated after excessive plastic deformation [127].

5.3.2 Digital Image Correlation

DIC is a non-contact optical measurement technique capable of capturing full-field displacement data across the entire surface of the specimen, allowing for spatially continuous data acquisition without the need for intrusive setup [128]. This is advantageous to conventional instruments such as strain gauges or linear variable differential transformers (LVDTs), which provide localised measurements, and require prior knowledge of the region of deformation. The plates were painted with a high-contrast, non-repetitive, homogeneous 'speckle pattern', and DIC involved the tracking of this pattern across a sequence of images acquired via cameras during loading. A subset of pixels within the image possesses a distinct grayscale signature, which is assumed to remain locally undistorted throughout the deformation. These subsets are tracked to estimate surface point coordinates, and

displacement fields are obtained by solving an optimisation problem that determined the best-fit transformation of the pixels within these subsets [129, 130].

5.3.2.1 Speckle Pattern and Subset Size

The accuracy of the DIC measurements greatly depend on the uniqueness and quality of the pixel signature, and so is strongly influenced by the speckle pattern's contrast, randomness, and spatial resolution. A black-on-white speckle pattern was applied thinly and uniformly with matt spray paint. Based on the sampling theorem, at least two pixels per period are required for a speckle to be unambiguously represented digitally [131]. Speckles smaller than this size cause aliased results, which adds noise to the measurement and negatively affects the accuracy. This often results from a fine 'mist' produced alongside larger spray paint droplets. To minimise this effect and maintain spatial resolution, a speckle size of three to five pixels in diameter was targeted. A speckle size of three pixels requires a subset size approximately four speckles in diameter [129], therefore, a subset size of three to five speckles in diameter was targeted.

5.3.2.2 Hardware Setup

To enable both in-plane and out-of-plane deformation measurements, a stereo digital image correlation (stereo-DIC) configuration was employed (Figure 5.4). The imaging system comprised two Manta G-895B ASG PoE cameras equipped with 35mm focal length lenses, mounted in a vertically aligned, rigid stereo arrangement on a single tripod to eliminate relative camera motion during acquisition [132]. These were positioned at 1.0m from the specimen surface in accordance with the speckle size and focal length for an appropriate field of view and adequate resolution. The minimum stereo angle between the optical axes of the two cameras was set to approximately 15°, consistent with that recommended for high in-plane accuracy, while maintaining focus across a wide range of out-of-plane deformations [130]. Nevertheless, it is acknowledged that reduced stereo angles may increase the uncertainty associated with out-of-plane displacement measurements [130]. The tests were performed over three days, and both the setup and calibration were

re-established at the start of each test. It is noted that the stereo angle on days two and three (19/02/2024 and 20/02/2024) was slightly greater than that on day one (16/02/2024).

In conjunction with speckling quality and subset size (Section 5.3.2.1), the lighting, exposure and contrast is paramount to accuracy. National Instruments Measurement & Automation Explorer (NI Max, [133]) was used to prescribe an identical exposure time for each camera. To reduce glare, matt paint was used to speckle the metal plates, and the reflective parts of the loading frame were covered with paper tape. Two 85W 5500°K lamps were positioned symmetrically 50mm either side of the cameras, to minimise the shadow from the load frame boundaries while avoiding overheating the cameras, see Figure 5.4.

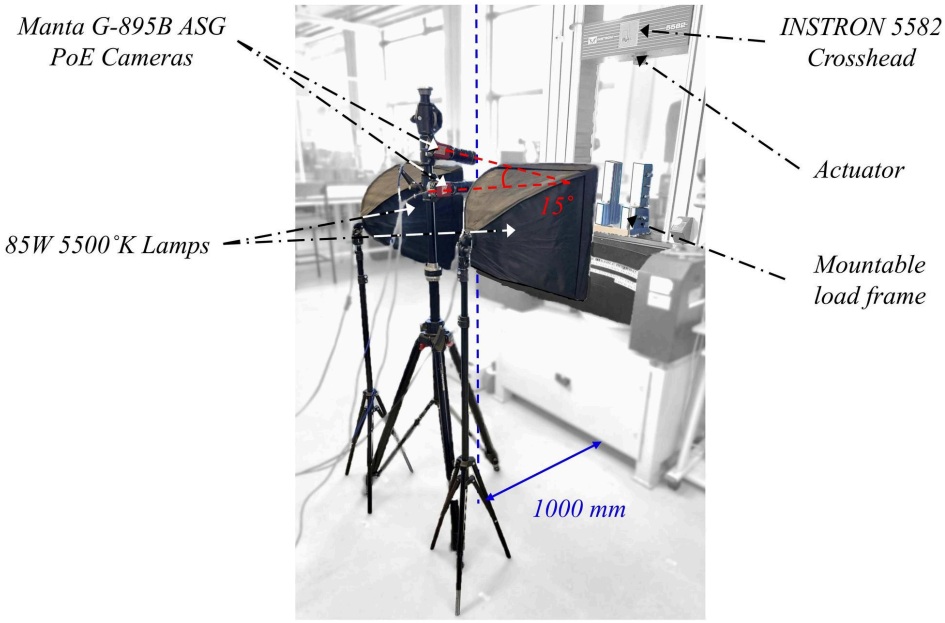


Figure 5.4: Hardware setup, with loading rig partially installed (actuator attachment and specimen absent), see Appendix E

5.3.2.3 Software Setup

All DIC measurements were recorded and analysed using the open-source MATLAB toolbox 'DuoDIC', with calibration and file management procedures carried out according to the provided documentation [125]. At least 225 images were collected for calibration, necessary for an accurate Bundle Adjustment [125]. Calibration involved moving a checkerboard pattern on a Perspex plate within the field of view. Images where the checkerboard was blurred or part of the checkerboard was out of view were removed during processing. The selection of calibration image candidates was refined until the average mean error of the selection was 0.33 pixels or less. Next, a static image was recorded at negligible load and was used to provide the geometric shape and amplitude of the geometric imperfection in the plate (Section 5.3.3.4). After the commencement of loading, subsequent images were captured and recorded every 2 seconds with the aid of a LabView routine [134]. Approximately 300 images were recorded for each test, from which 30-40 were selected at appropriate intervals between the initiation of loading and the attainment of ultimate load capacity. These images were used to extract full-field deformation information.

5.3.3 Specimen

5.3.3.1 Material Properties

The plates were made of aluminium alloy 1050 in an annealed temper [135], with nominal dimensions 300mm by 130mm. In accordance with the guidelines in [127], dog-bone specimens were designed for tensile test to determine the stress-strain curve. These tests were completed and the engineering stress-strain curve obtained was imported into ABAQUS and converted into the true stress-plastic strain curve (shown until rupture in Figure 5.5). The yield stress was taken as the proof stress (i.e., at the point at which the plastic strain attains 0.2%), and the ultimate tensile stress was taken as the maximum stress reached (88.0 MPa and 219 MPa respectively). The measured Young's Modulus was 60.1 GPa, and the Poisson's ratio was entered manually as 0.33. The proof (x%) strain (ϵ_x) and stress (σ_x) is summarised in Table 5.1.

Table 5.1: True stress-strain values

	0.05%	0.2%	1.0%	2.0%	Ultimate
ϵ	0.00174	0.00345	0.01177	0.02215	0.141
σ	78.8	88.0	112	138	219

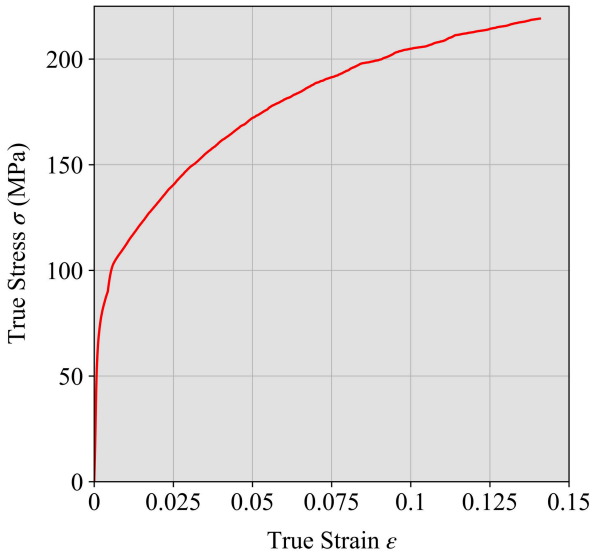


Figure 5.5: True stress-strain curve

5.3.3.2 Dimensions

The thickness of each specimen was measured with a digital calliper in six locations approximately 50mm from the edge of the plate and summarised in Table 1. Although each plate measures 300mm by 130mm, approximately 10% of the plate area was obscured by the loading rig, and a portion of each plate sat within the grooved boundary. Furthermore, small shadows excluded part of the edges from the region of interest (ROI). These dimensions will be of interest when post-processed DIC images will be used to create the finite element (FE) model described in Section 5.4. The length and width (a and b respectively) of the resulting modelled region are summarised in Table 5.2.

5.3.3.3 Perforation Patterns

Seventeen perforation patterns were examined, alongside one unperforated plate. The dimensions of both the perforations and the plate for each test are summarised in Table 5.2. This study aimed to investigate patterns frequently encountered in both structural and architectural applications, as illustrated in Figure 5.1. In addition, it sought to assess how plates with an irregular or patterned perforation arrangement compare to a standard regular design. Often encountered in rack sections or cable trays, Tests 2, 3, 4, 7 and 8 have a regular arrangement of perforations in non-overlapping rows and columns (Figure 5.6(a)) and Tests 5, 6, 9, 10, 11 and 13 have a staggered arrangement of perforations (Figure 5.6(b)). More representative of architectural cladding or bridge parapets (Figure 5.1), the perforations in Test 12 and 16 are in a pattern (Figure 5.6(e), Figure 5.6(f)), and those in Tests 14, 15, 17 and 18 are arranged irregularly (Figure 5.6(g), Figure 5.6(h)).

Design manuals and standards typically characterise perforated plates by their hole length and width (l_x and l_y), and the perforation-to-perforation spacing (s_x and s_y) [1, 136]. For plates with a staggered arrangement of perforations, s_x and s_y are taken as the minimum distance between any two perforations in the x and y directions, whereas, for plates with an irregular arrangement of perforations, l_x, l_y, s_x and s_y are the average hole dimensions and spacings. As many of the designs considered herein extend beyond conventional ones, the minimum perforation-to-plate edge spacing (S_{xo} and S_{yo}) and the number of perforations (n_x and n_y) over the 'critical length' are also required for a complete characterisation of the perforations layout (Table 5.2). The critical lengths are the narrowest lengths in the longitudinal (x) and transverse (y) directions respectively (Figure 5.6(b), Figure 5.6(c), Figure 5.6(d)). In this case, n_x and n_y denote the number of holes along a given row and column, assuming the perforations are arranged into the closest equivalent regular grid (Figure 5.6(g)).

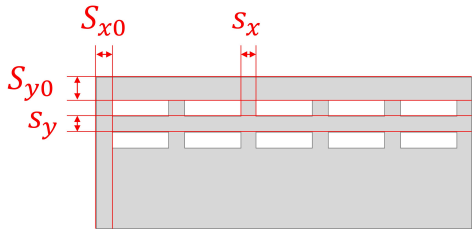
Wire Electrical Discharge Machining (EDM), water jet cutting, laser cutting and CNC milling were techniques considered for perforating the specimens. Unlike the former methods, CNC milling involves contact with the specimen, causing greater deformation ('lipping') at the perforations, and was dismissed. As it is a 'cold cutting' method, water jet perforating was desired, however commercialised perforated sections are unlikely to

use this manufacturing technique which is expensive and time consuming. As it is a superior technique to wire EDM for thinner and lighter specimen, laser cutting was the chosen technique for creating perforations within the plate. This exposed the perforation border to high temperatures which could induce recrystallisation (annealing), leading to reduced tensile strength and increased ductility at the hole border. This may cause reduced experimental post-buckling capacity. Given the thinness of the plates (Table 5.2), localised heating may also lead to warping.

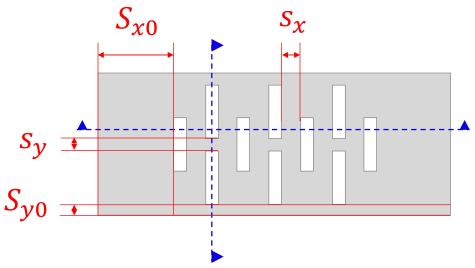
Table 5.2: Perforation parameters for each test (relevant variables defined in Figure 5.6)

Test	a (mm)	b (mm)	t (mm)	l_x	l_y	n_x	n_y	s_x	s_y	S_{xo}	S_{yo}
1	284	116	0.890	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	289	116	0.895	45	48	1	1	0	0	123	34
3	289	117	0.903	45	24	2	1	70	0	65	46
4	288	113	0.898	45	24	5	1	12.5	0	8	46
5♢	287	117	0.908	22.5	12	7	3	15	12	17	28
6♢	290	117	0.885	12	45	7	2	30	10	13	9
7	290	117	0.878	45	12	5	2	12.5	15	8	38
8	290	115	0.885	45	12	5	2	12.5	12	8	11
9♢	288	116	0.902	45	12	3	4	12.5	12	7	16
10♢	285	117	0.898	12	45	7	2	15	10	58	9
11♢	279	116	0.887	8	24	7	3	11	6	17	16
12♣	285	117	0.917	8	24	14	3	11	6	17	16
13♢	291	117	0.897	8	8	7	3	11	22	18	24
14★	285	118	0.897	8	24	7	3	10	9.5	44	9
15★	285	118	0.890	8	24	7	3	15	11.5	10	10
16♣	290	116	0.885	22.5	12	1	1	7.5	1	29	9
17★	286	119	0.910	8	8	13	5	8	9	31	9
18★	285	118	0.898	15	19	6	3	15	21.5	17	11

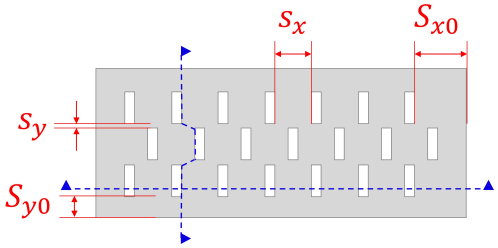
♢ The perforations are staggered. n_x and n_y are the number of perforations over the 'critical length' and 'critical width', taken as the narrowest length in the longitudinal and transverse direction respectively (Figure 5.6(a)). ★ The perforations are irregular; l_x and l_y are the average hole height and length respectively. n_x and n_y is the number of holes in a given row and column, for perforations arranged into the most similar regular arrangement (Figure 5.6(g)) ♣ The perforations are arranged in a pattern (see Figure 5.6(e) and Figure 5.6(f))



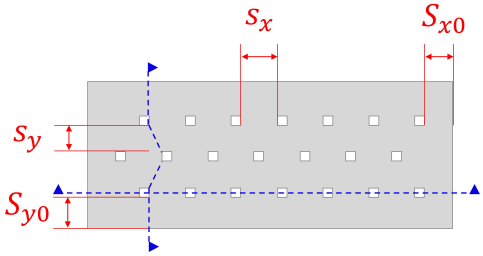
(a) Regular perforations, Test 8



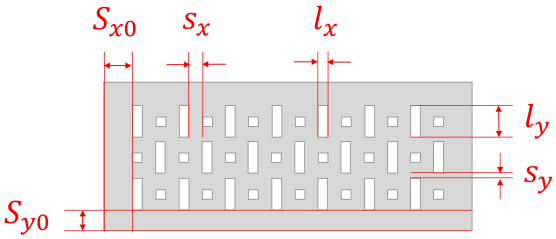
(b) Staggered perforations, with the critical lengths shown in blue, Test 10



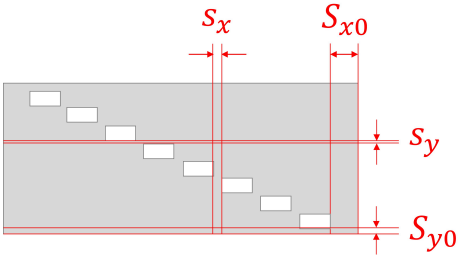
(c) Staggered perforations, with critical lengths shown in blue, Test 11



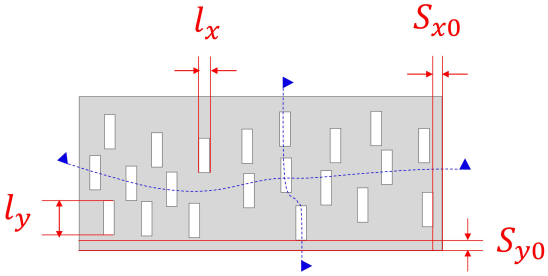
(d) Staggered perforations, with critical lengths shown in blue, Test 13



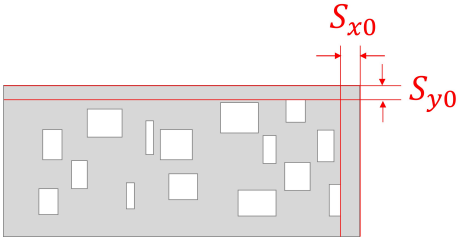
(e) Patterned perforations, Test 12



(f) Patterned perforations, Test 16



(g) Irregular perforations, with the critical length shown in blue, Test 15



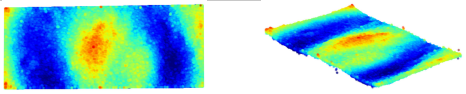
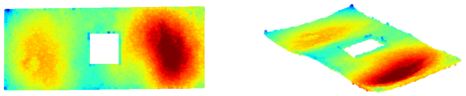
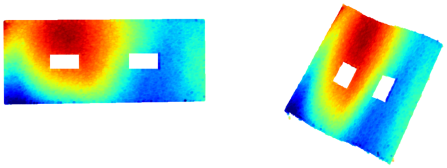
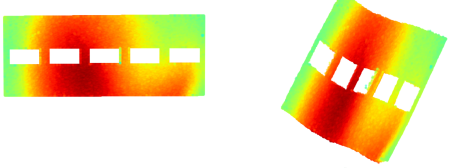
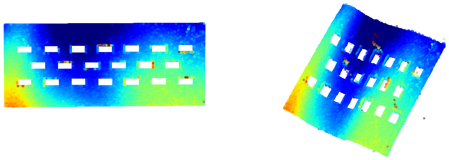
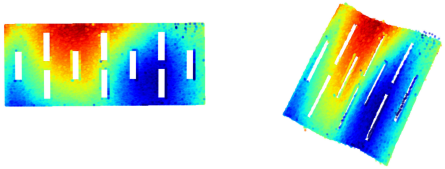
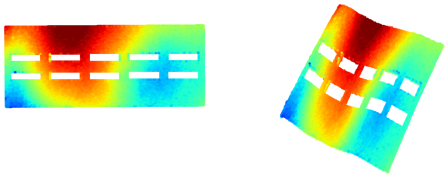
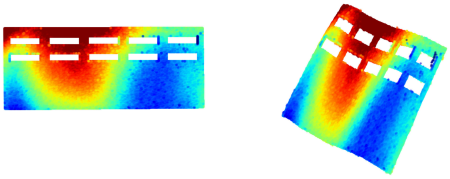
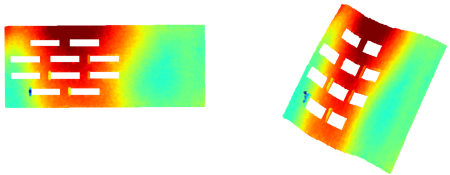
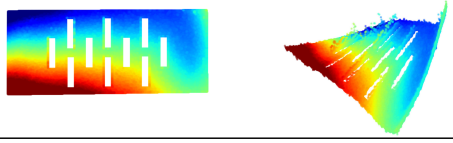
(h) Irregular perforations, Test 18

Figure 5.6: A selection of specimen with regular, staggered, patterned and irregular perforations

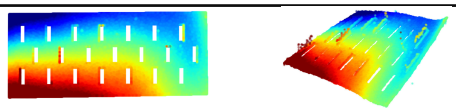
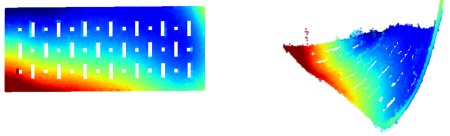
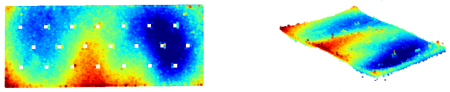
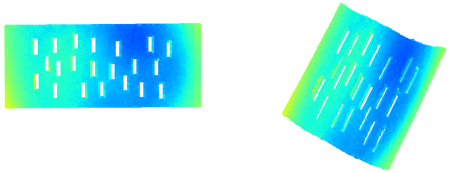
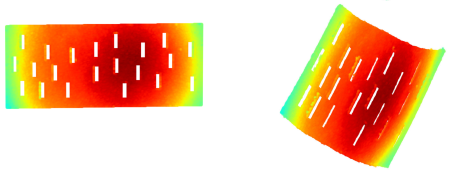
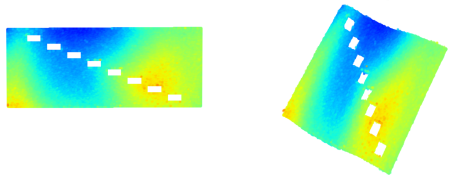
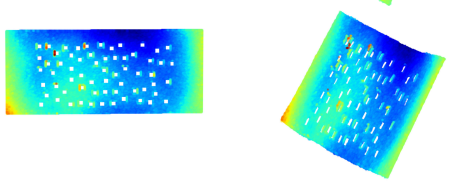
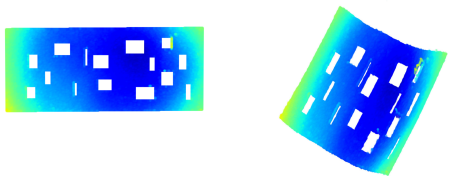
5.3.3.4 Initial Imperfection

As buckling is highly sensitive to imperfections, a visual inspection for scratches, dents or other unconformities was made prior to testing. Table 5.3 includes a description of these defects according to whether it resembled an eigenmode ('Eigenmode'), warping ('Twist'), was particularly dented in one or two areas ('Dent') or was deformed in only one direction, like a pin-ended column ('Bow'). When modelling plates with the FE method, it is recommended in [18] to consider a geometric imperfection shape equivalent to the first buckling eigenmode, with a magnitude δ_{nom} equal to the minimum of $a/200$ or $b/200$ where a and b are the longitudinal and transverse dimensions respectively. Noteworthy, many measured maximum imperfection amplitudes (δ_{exp}) exceed this limit ($\delta_{nom} \approx 0.58$mm), and are large relative to the nominal panel thickness (0.9mm). The comparatively large imperfections observed are characteristic of perforated plates, where local stiffness reduction around openings leads to increased out-of-plane deformation during fabrication. Furthermore, the suggested imperfection level in [18] is based largely on engineering judgement rather than comprehensive statistical evidence, as systematic measurements of initial deformations are rarely published [137]. Therefore the observation that $\delta_{exp} > \delta_{nom}$ does not invalidate this study, but instead contributes to this limited body of evidence with high-fidelity measurements. Importantly, these measured imperfections were directly incorporated into the FE models, ensuring that both experimental and numerical results accurately reflect the as-built geometry, and that the results remain representative of real manufacturing tolerances (Section 5.4.3). Although revisiting the current code-based imperfection limits is beyond the scope of this paper, these measurements suggest that existing prescriptions for unperforated plates may be unconservative when applied to perforated structural elements.

Table 5.3: Description of observed buckling deformation

Test	Shape	δ_{EXP} (mm)	Image
1	Eigenmode	0.372	
2	Eigenmode	0.543	
3	Dent	0.461	
4	Bow	0.828	
5	Dent	0.790	
6	Dent	0.598	
7	Dent	0.681	
8	Dent	0.623	
9	Dent	0.706	
10	Twist	1.58	

Continued on next page

Test	Shape	δ_{EXP} (mm)	Image
11	Twist	1.83	
12	Twist	1.54	
13	Eigenmode	0.573	
14	Bow	0.680	
15	Bow	1.06	
16	Dent	0.587	
17	Bow	0.597	
18	Bow	0.979	

An initial imperfection will influence both the critical elastic buckling load, and the post-buckling resistance of the plate. A greater imperfection size increases the ambiguity of the bifurcation point, causing a smoother transition between the initial and buckled states. The consequence is that a plate with a greater imperfection will buckle at a reduced capacity. Therefore, this imperfection must be accounted for in the subsequent 'geometrically and materially nonlinear analysis with imperfections' (GMNIA) finite element (FE) model. The elastic buckling load of a perfect plate can be determined in a linear perturbation assessment of a FE model, in which an initial imperfection would include distortions due

to residual stresses, lack of verticality, lack of straightness, or lack of flatness [18]. These influences can be captured in one equivalent geometric imperfection introduced into the numerical model and defined by both its shape and magnitude. A linear perturbation analysis produces the buckling modeshapes with eigenvector unity normalisation on the displacement, and it is typical to assume an equivalent geometric imperfection equal to the magnitude multiple of the first fundamental buckling mode, as this grows more rapidly [138]. However, for accurate comparison between the FE model and the experimental data, it is more appropriate to use a model considering both the imperfection size and shape measured with DIC. It was demonstrated by [105] that when the initial imperfection does not closely resemble a particular buckling eigenmode, its shape can be as influential as its amplitude. This may be especially relevant in the present study, where uneven cooling following laser-cutting resulted in warped or twisted imperfection geometries (Table 5.3). Furthermore, the imperfection size and shape determined from the DIC measurements revealed that the tested specimens were often slightly skewed. Consequently, imperfections were evaluated relative to a reference plane representing the undeformed mid-surface of the plate. Three points on the plate width were selected (two on the edge loaded by the actuator, and one on the opposite edge) to define this mid-surface, and the size and shape of the imperfection was described by the out-of-plane deviation of nodes relative to the reference surface (Figure 5.7). The imperfection images exhibited varying levels of noise, particularly in Tests 10-13 around the hole perimeter (Table 5.3). This may be due to suboptimal ROI selection during post-processing or inadequate speckle quality. As expected, Tests 1, 2 and 13 exhibited very little warping as their perforated area was low or none. In contrast, plates with many closely spaced or large perforations showed the greatest warping (Tests 10, 11 and 12, Table 5.3).

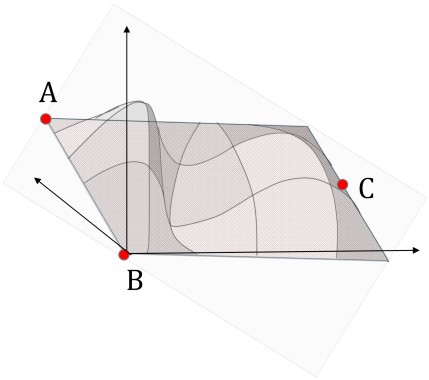


Figure 5.7: Example of skewed imperfect plate with reference mid-surface and three points in red

5.3.4 Issues with Experimental Testing

As outlined in [130, 139], many sources of error in DIC (such as speckle pattern, lighting, lens distortion, subset size, out-of-plane displacement, sub-pixel registration algorithm, interpolation scheme, etc) make it is nearly impossible to isolate specific error contributions. Although some general issues with DIC have been discussed in Section 5.3.2.1, one particular issue is the subset size. Larger subsets are desired to distinguish a unique pattern, whereas smaller subsets are required for accurate approximation of underlying deformation [139]. Additionally, the displacement of the deforming specimen is calculated as the average displacement of pixels within each subset, significantly influencing the mean percentage pixel error. This percentage error is given in Table 5.4, with a summary of the mean pixel error, alongside approximate mean speckle size. Accurate DIC measurements require the specimen to be viewed in high contrast, which significantly reduced during the day as the tests were conducted during winter. In Table 5.4 it can be seen that the mean pixel error generally increased throughout the second day, where the first and last test (Test 3 and Test 16 respectively) were commenced five hours apart, reaching a maximum of 15% error for Test 12.

Table 5.4: Percentage pixel error

Test	Test Date	Mean Pixel Error (pix)	Approx. Speckle Size (pix)	Error (%)
1	16/02/2024	0.15	7	2.14
2	16/02/2024	0.215	5	4.30
3	19/02/2024	0.2	5	4.00
4	19/02/2024	0.2	7	2.86
5	19/02/2024	0.2	9	2.22
6	19/02/2024	0.45	8	5.63
7	19/02/2024	0.2	8	2.50
8	19/02/2024	0.35	14	2.50
9	19/02/2024	0.45	15	3.00
10	19/02/2024	0.55	7	7.86
11	19/02/2024	0.7	5	14.0
12	19/02/2024	0.75	5	15.0
13	16/02/2024	0.1	6	1.67
14	20/02/2024	0.09	5	1.80
15	20/02/2024	0.09	8	1.13
16	19/02/2024	0.55	5	11.0
17	20/02/2024	0.05	7	0.71
18	20/02/2024	0.09	7	1.29

A relatively small pixel error can skew the apparent stiffness measurement by several percent, as the stiffness measured with DIC is given by:

$$k \approx \frac{\Delta F}{\Delta \delta_{meas} \pm \Delta \delta_{err}} \tag{5.1}$$

Other discrepancies in the homogeneity, contrast, sparseness, and size range of the speckles also contribute to pixel error. A relatively large proportion of the speckles in Tests 4, 7, 8, 11, 12, and 16 were smaller than the minimum recommended speckle size, while the contrast was poor for Tests 6, 9, 11, and 16 conducted later in the winter day. These issues caused some aliasing, whereby the centre of a speckle cannot be unambiguously assigned to a single pixel. This difference is illustrated by comparing the subset images from a good test

and a poorer test (Figure 5.8). However, the impact of these issues on stiffness measurement only becomes critical near ultimate failure, where the nonlinear behaviour may be obscured by resolution limitations. Hence, the most error-prone tests (i.e., Tests 6, 10, 11, 12 and 16, Table 5.3) should be interpreted with greater caution as they approached ultimate load.

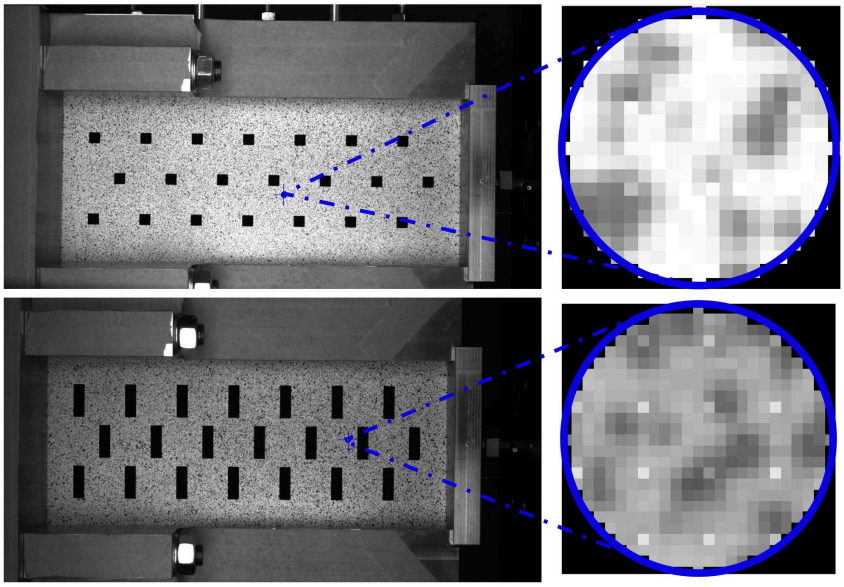


Figure 5.8: Example of good subset (Test 13, top); Example of poor subset (Test 11, bottom)

In addition to calibration limitations, experimental errors can arise from hardware such as the loading rig or the specimens themselves. Despite careful design, the experimental boundary conditions may allow slippage and warping at high loads (which is not possible in the perfectly pinned conditions of the computational model). Also, slight skewing of the plate during testing could cause eccentric loading. Thirdly, the presence of significant non-eigenmodal imperfections in most specimens could contribute to discrepancies between observed and computational mode shapes. As the effective aspect ratio of each specimen lies near the transition between buckling modes in two or three halfwaves (Figure 5.9), even minor rotation or overly-tight supports can introduce mixed boundary conditions and alter the number of halfwaves. Similarly, eccentric loading may trigger asymmetric modes, causing deformations to concentrate in one region.

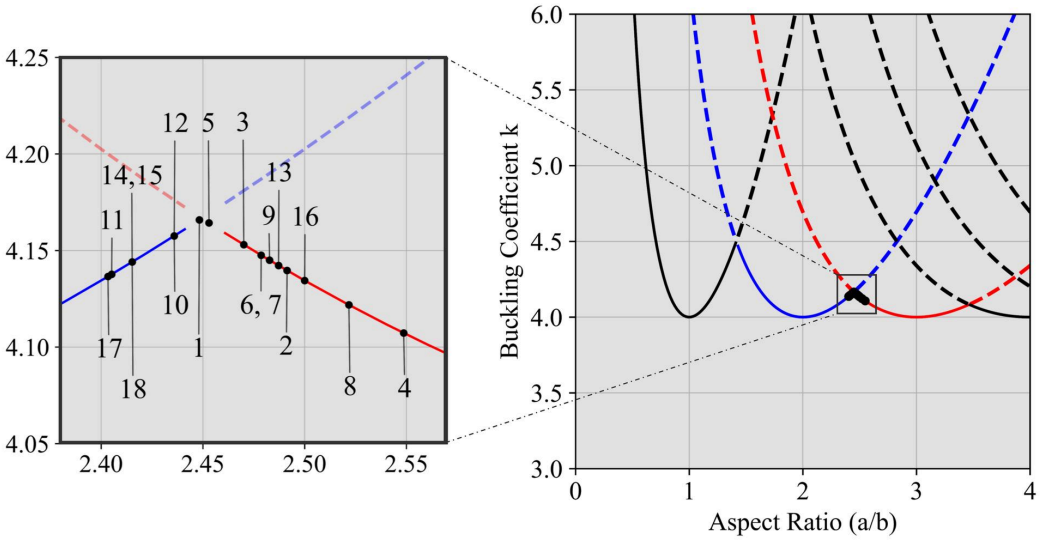


Figure 5.9: Aspect ratio of specimen lies near the halfwave-transition point

5.4 GMNIA ABAQUS Model

5.4.1 Convergence Study

As described in Section 5.3.3.4, the nodal data collected with DuoDIC is used to create the FE mesh. The configuration of these nodes prevents the use of S9R5 (Figure 5.10). Although a S4 element cannot adequately capture extreme nonlinear behaviour, [140] identify S4 elements to be the best alternative to S9R5 elements, outperforming S4R and S4R5 elements in problems prone to membrane or bending mode hour-glassing, areas where greater solution accuracy is required, and problems where in-plane bending is expected [140]. Therefore, a mesh convergence study was conducted with S4 elements for a plate with a slenderness (b/t) of 155. This study performed a linear perturbation buckling assessment on an unperforated plate with boundary conditions as described in Section 5.4.2, allowing comparison to the theoretical elastic buckling load of a simply-supported plate.

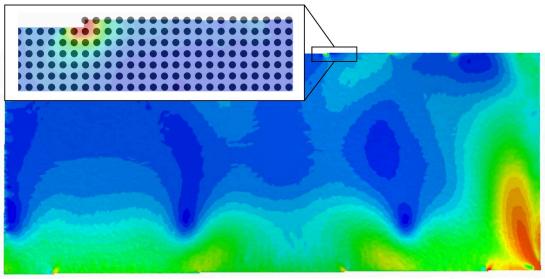


Figure 5.10: DuoDIC 3DPoint data used to create the ABAQUS mesh

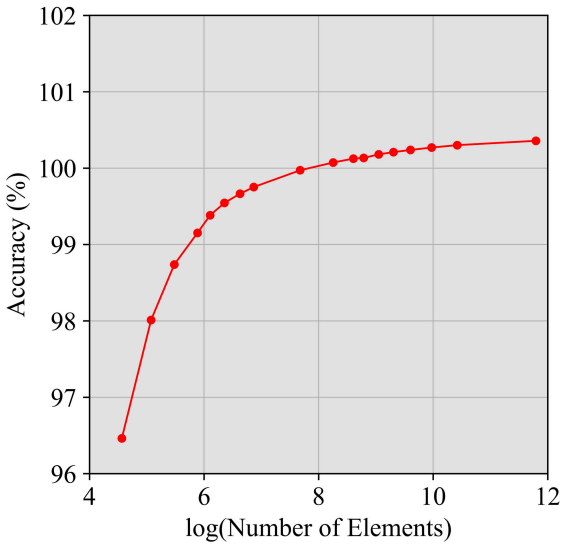


Figure 5.11: Convergence study for S4R elements

The 'accuracy' is defined as the ratio of the ABAQUS-found capacity to that given by Timoshenko's plate buckling equation [9]. At least 15000 S4R elements were required for an accuracy within 0.2% of the true value for the critical elastic buckling load. It is noted that the mesh is composed from elements that cannot account for shear; equally the theoretical result is invalid for particularly thick plates. This explains the convergence to a value slightly greater than 100% accurate. Thus, each model in this study was meshed with at least 15000 S4R elements, of a thickness equal to the measured value (Table 5.2), with five integration points through their depth.

5.4.2 Boundary Conditions

A GMNIA finite element model was developed for the purpose of this study. The boundaries and loading conditions were applied as nodal constraints, as shown in Figure 5.12. The displacements in the x-, y-, and z- directions are u, v and w respectively. All edges were restrained in out-of-plane motion (w), and the left-hand edge was fully restrained in the direction of loading (u). The in-plane transversal displacement (v) of the central node of the left-hand edge was also restrained.

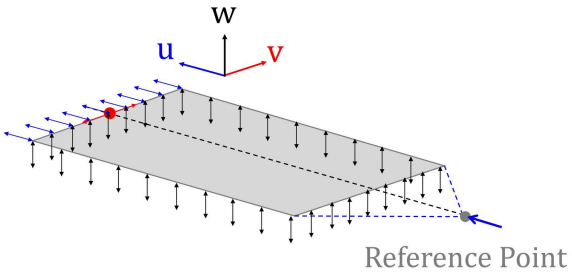


Figure 5.12: Boundary conditions shown by double-headed arrows. Load is applied in the u-direction (blue) at a reference point (grey) coupled to the edge

5.4.3 Loading and Imperfection Modelling

Both the Newton-Raphson and modified Riks arc-length methods were considered for the geometric nonlinear step. In [141],[142], and [143], authors find that arc-length methods are more robust than simple load-control when one measures behaviour past limit points, whereas the Newton-Raphson method is fine for monotonic and well-conditioned problems. Specifically, [143] shows that although the standard Newton-Raphson load-control method generally fails at a limit point, the displacement control method can pass turning points such as ultimate failure, provided there is no snap-through behaviour. Furthermore, the arc-length method implemented in ABAQUS is unable to include a stabilisation algorithm in the solution, yet this is accommodated by the Newton-Raphson method and is typically required when localised instabilities occur [140]. This study considered both methods with displacement control for all tests; the Riks step considered a minimum increment size $\Delta l_{min} = 0.0001$ and a maximum of $\Delta l_{max} = 2$, starting with an initial time increment in arc length of 0.01; the Newton-Raphson step considered a minimum increment size $\Delta l_{min} = 0.000001$ and a maximum of $\Delta l_{max} = 0.005$, starting with an initial step size of 0.001 (see Appendix F). Except for Test 7 (see Section 5.5.1), the study found no difference between the measured force-displacement relationships. Unless otherwise stated we will consider the Newton-Raphson method as the nonlinear step (although the Riks arc-length method could equally have been chosen). The applied load was approximately 50% greater than the elastic buckling load determined in a linear perturbation FE assessment and was

applied at the reference point, which was coupled with continuum distributing coupling to the loaded edge (Figure 5.12). This constrained the average displacement and rotation of the reference point to the displacement of the loaded nodes [144]. As the full edge is loaded, the computational representation of force differs from the experimental condition, where application of force to the edges of the plate within the semicircular groove is restricted by design (Figure 5.3(c)). However, this unloaded length is very small and considered negligible. Weight factors were selected to distribute the load uniformly over the loaded region. Displacement and force data were collected at the loaded reference point. The force at the reference point can be applied with either 'load' or 'displacement' control; the former being typical where nodal displacements are nonuniform along the loaded edge. However, displacement control may better replicate stress redistribution within the plate particularly when the strain is plastic, hence the ultimate capacity of a load-controlled FE model may be conservative as flexible parts of the buckled cross section are overstressed [140]. The imperfection was completely informed by the experimental static DIC image as the model was directly constructed with the 3DPoint z co-ordinate data from DuoDIC in [145]. The x, y co-ordinate data of the ABAQUS mesh was generated for each specimen from the 3DPoint data in the DuoDIC 3D Reconstruction Stereo file, and the z co-ordinate data was obtained via the plane equation derived as discussed in Section 5.3.3.4 [125]. Due to the nature of this mesh, it exhibited a stepped appearance at the top and bottom edges (Figure 5.10) where it displayed unrealistic stress concentration. However, this difference was not particularly significant for our study which focuses on the elastic and immediate post-buckling behaviour. In Figure 5.13, the experimental performance of the unperforated plate is compared to a GMNIA with the first eigenmode as the initial imperfection (with a size of 0.372mm as per Table 5.3, and GMNIA models 'completely informed' by the measured imperfection. Both of the 'completely informed' GMNIA models is much more representative of the initial stiffness measured with DIC than the eigenmode model. The ultimate capacity obtained by the displacement-controlled 'completely informed' GMNIA model is more similar to that observed experimentally and underestimated with load control. The post-ultimate behaviour depends on the numerical control method used [140], and modelling parameters, such as the maximum

load increment in the step and the choice of element, also affect the smoothness of peak load reduction [77]. However, this study is concerned with the behaviour of a uniformly compressed thin plate up until ultimate capacity, which is better modelled with the displacement-controlled 'completely informed' GMNIA model. To emphasise attention on the ante-ultimate behaviour, post-ultimate behaviour will henceforth be displayed with either hollow points or a dashed line in the next sections.

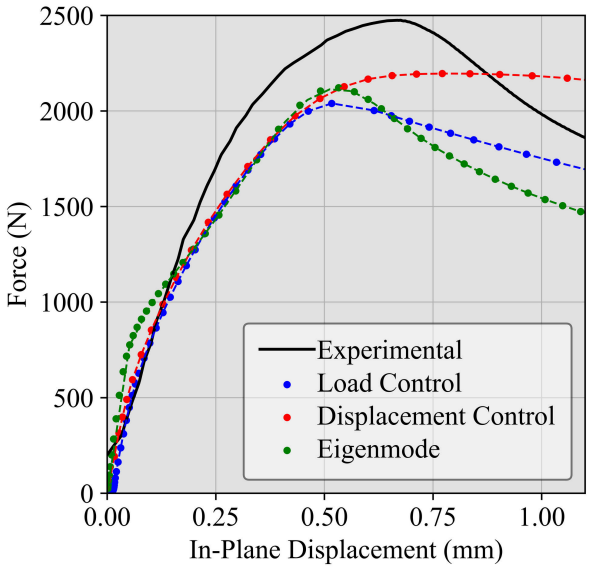


Figure 5.13: A comparison of different GMNIA models

5.4.4 Validation and Accuracy

The 'completely informed' displacement-controlled GMNIA model is taken forward and assessed in more detail. The force-displacement relationship for the unperforated plate shown in Figure 5.14 displays good agreement between the experimental and computational results for the initial period of loading. The modeshapes from the computational and experimental results are presented in Figure 5.14, labelled (i) and (ii) respectively, with loading from the right-hand edge. While three halfwaves were expected, and observed in the GMNIA model, only two halfwaves were seen experimentally, attributed to the factors discussed in Section 5.3.4. The elastic buckling load determined in the linear perturbation buckling step is denoted by the dotted horizontal line and has only a 2.2% error from the theoretical elastic buckling load of an unperforated plate with dimensions a,b given in

Table 5.2. The point at which elastic buckling occurs is indistinct for both the GMNIA model and experimental results, attributed to the geometric imperfection, a small dent visible on the loaded edge (Table 5.3). This imperfection promotes more pronounced deformation growth in the model than occurs experimentally, explaining why the GMNIA model underestimates the ultimate buckling capacity (Figure 5.14). In [146], a numerical expression for the nonlinear post-buckling behaviour of an imperfect unperforated plate is presented. This expression (shown as the dashed line in Figure 5.14(b)) is compared to the out-of-plane displacement at the buckling maximum in both the experimental data and the GMNIA results. The out-of-plane displacement data for the experimental results is of low quality initially, which is attributed to the narrow stereo angle used in the setup (Section 5.3.2.2). At an out-of-plane displacement of approximately 0.5 mm, the experimental data exhibits stiffness comparable to the numerical model. From this stage until buckling, the GMNIA model accurately represents the out-of-plane behaviour; with the difference in ultimate capacity discussed previously.

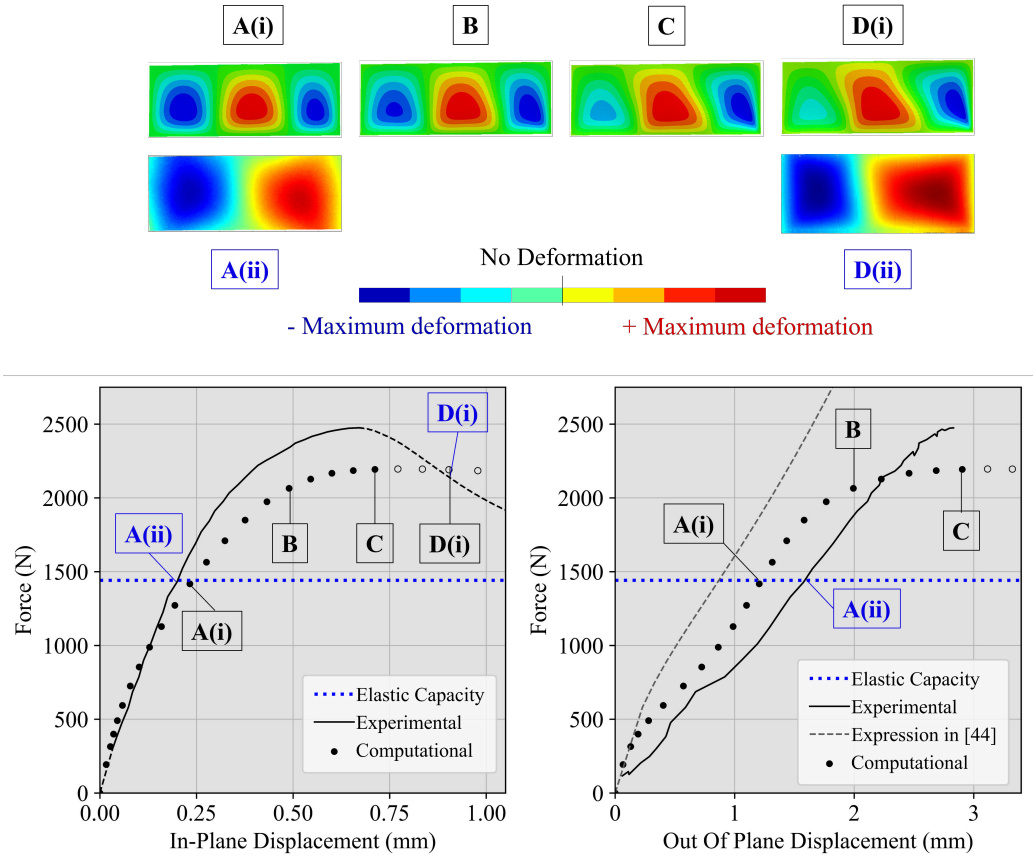


Figure 5.14: Force-displacement relationship for the unperforated plate. *Please note that this figure is taken from [25], where Ref. [44] is Ref. [146] in this thesis*

5.5 Results

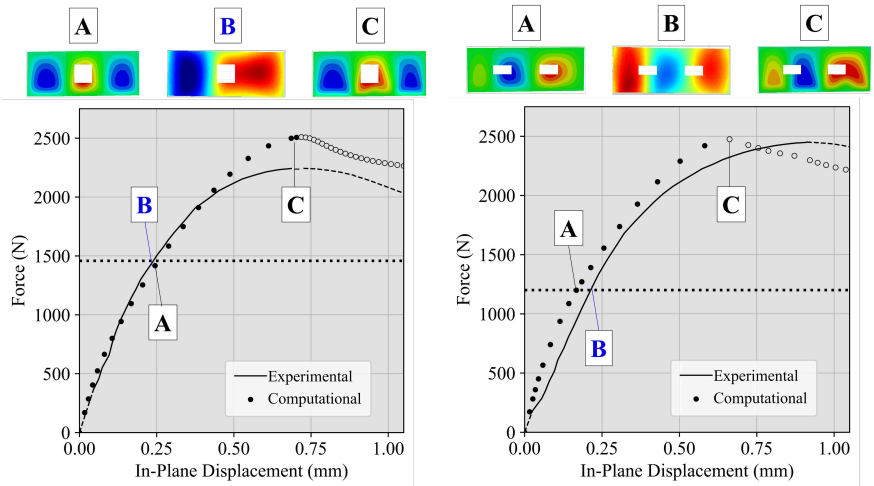
5.5.1 Load Displacement Curves

The relationship between the total load and the displacement of the loaded edge is plotted for all tests in Figure 5.15(a) to Figure 5.15(q). There is good agreement between the GMNIA and experimental force-displacement curves during the initial loading phase. For loads less than approximately 25% of the elastic buckling load, the force-displacement relationship is linear for all tests. There is no distinct transition to a nonlinear relationship at the elastic buckling load for most of the tested specimens due to the influence of geometric imperfections. However, the point of elastic buckling can be identified as a point of inflection for Tests 1, 5, 7, 8, 9, 13, and 17. This point typically occurs at a lower load than predicted by the linear perturbation procedure. The experimentally obtained and computed out-of-plane displacement modeshapes are shown for each test at the point

of elastic buckling, and the latter is also shown at the ultimate capacity. As discussed in Section 5.3.4, eccentric loading can trigger asymmetric modes that focus deformation in a specific area. This phenomenon is evident in Tests 2, 4, 8, 9, and 17, where the experimental and computational local buckling modeshapes show notable differences. Several irregularities in the force-displacement curves prior to ultimate failure can be identified, specifically for the computational results of Test 7 and experimental results of tests 5, 10 and 11. In Test 7, the GMNIA model exhibits snap-through behaviour at 1750 N (Figure 5.15(f) - C), triggered by deformation at the left-most buckle which shrinks after snap-through (Figure 5.15(f) – D), causing antinodes to shift away from the loaded edge. For this reason, the force-displacement data shown in Figure 5.15(f) is found using the Riks arc-length method (see Section 5.4.3). This behaviour is attributed to a large dent, situated approximately at the location of the buckling node (see Table 5.3), demonstrating less freedom for movement in the computational model compared to the test (and absence of this movement must be accommodated by slippage in the boundary, as discussed in Section 5.3.4). The experimental results for Tests 5, 10 and 11 exhibit a sudden change in stiffness at approximately 700 N, 1050 N and 1000 N respectively (Figure 5.15(d), (i) and (j)) which may suggest a premature onset of buckling. In comparison to the experimental behaviour, the GMNIA force-displacement data for Tests 4, 5, 10, 16 and 18 is significantly different. For Test 4 and 10, this can be attributed to the different number of buckling halfwaves; a greater number has a stiffening influence. Test 10 experimentally exhibits deformation characteristic of LPB, with more deformation at the portions of plate between the outermost perforation and the loaded edge. However, the GMNIA model exhibits greater deformation at the plate centre, and buckles elasto-plastically, Figure 5.15(i). This suggests that either the plate buckles at the discontinuity discussed, or more likely that there is a greater force redistribution to the longitudinal supports in the experimental setup, which reduces the stress in the narrow regions above and below the perforations. Tests 5, 16 and 18 exhibit a much more ductile response than the experimental results, yet the out-of-plane displacement at ultimate failure strongly correlates with the imperfection profile identified in Table 5.3. The more ductile GMNIA can only be attributed to the interacting effects from the stress-affected zone, local plastic strain, stress concentration,

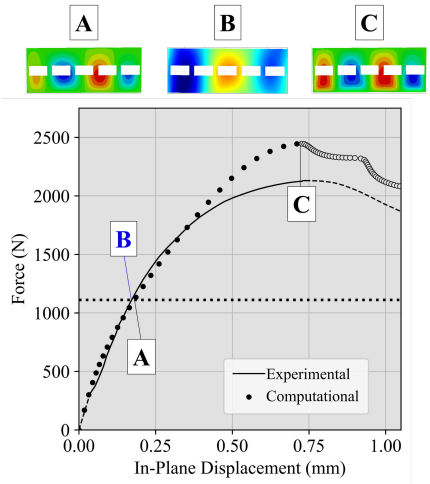
burred perforation edges and possible local thinning of these tested plates. The GMNIA model is much more sensitive to the dent observed in the imperfection profile for Test 16 (Table 5.3), which is attributed to the small premature buckle (Figure 5.15(o) - C). This difference in sensitivity is likely due to over-tightened boundaries.

The mean absolute percentage error between the computational and experimental results at ultimate capacity is 8.20% (see Table 5.5, column 7). This error is consistent with compression tests on simply supported perforated steel plates conducted by [110] and [147], who report an average error of 8-10% depending on the perforation shape. Additionally, [111] conduct an extensive experimental campaign on stiffened steel panels with a large concentric circular or elliptical cut-out and report an average error of 4-7% depending on the plate thickness. In addition to these similarities, when the results in Figure 5.15 are compared qualitatively to those in [148], the GMNIA model can be considered as a valid representation of the ante-ultimate force-displacement data obtained experimentally.

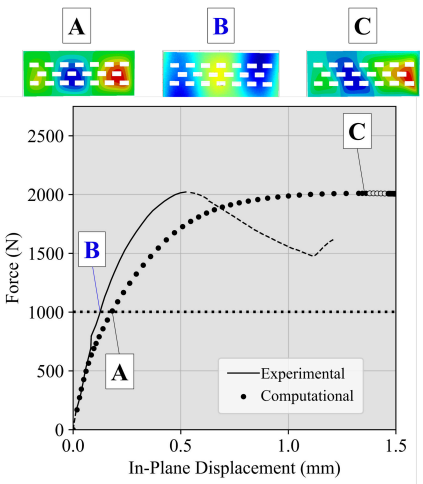


(a) Test 2 (4.30% pixel error)

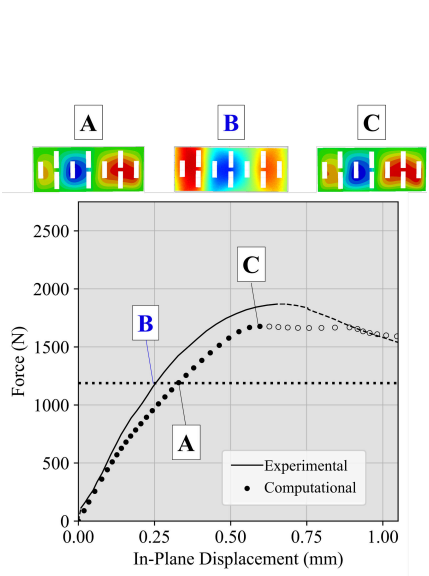
(b) Test 3 (4.00% pixel error)



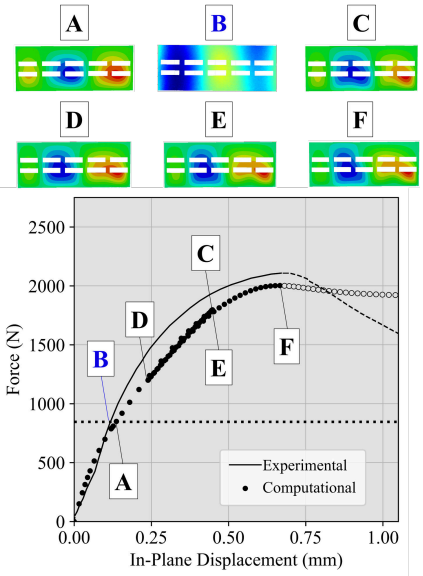
(c) Test 4 (2.86% pixel error)



(d) Test 5 (2.22% pixel error)

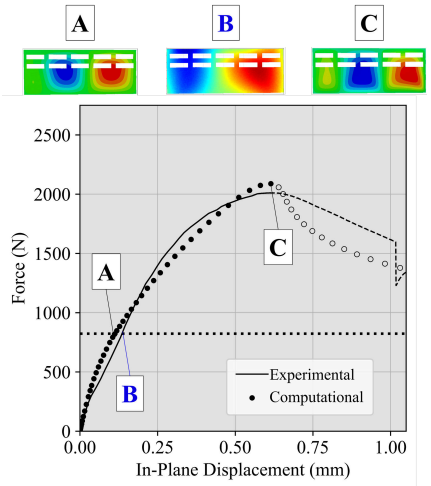


(e) Test 6 (5.63% pixel error)

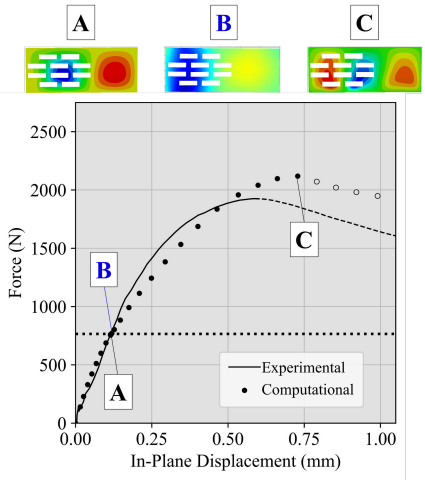


(f) Test 7 (2.50% pixel error)

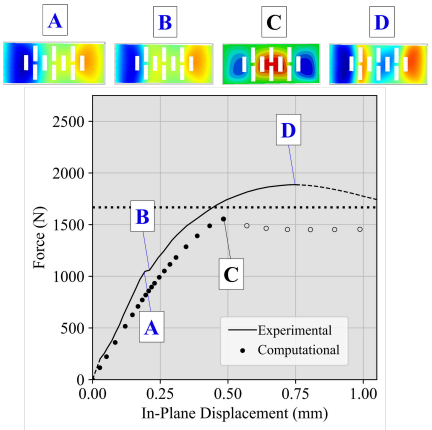
Figure 5.15: Force-Displacement Plots (Tests 2-7)



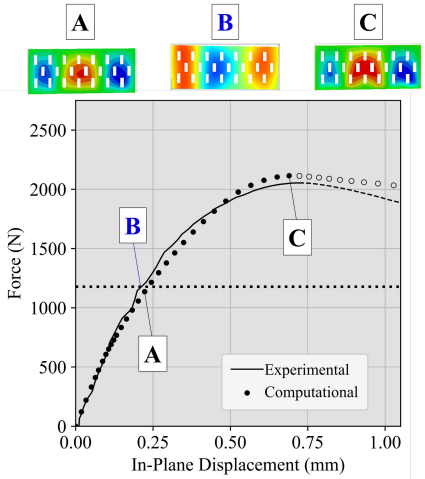
(g) Test 8 (2.50% pixel error)



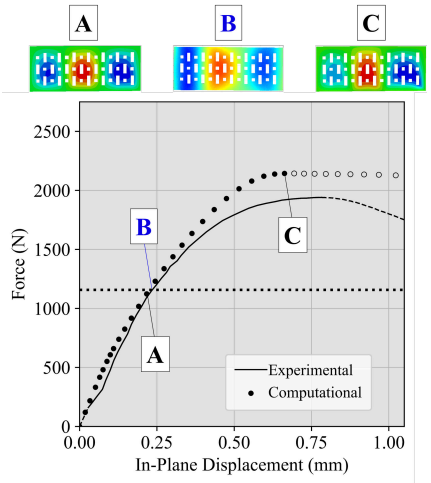
(h) Test 9 (3.00% pixel error)



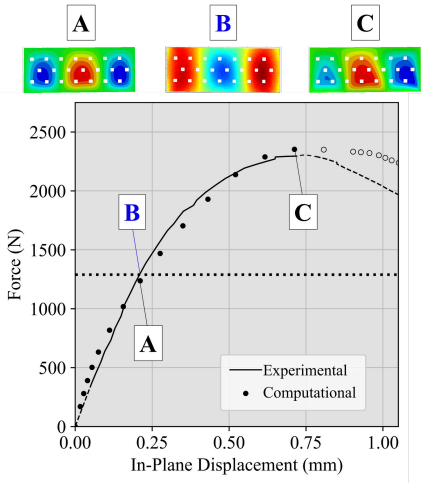
(i) Test 10 (7.86% pixel error)



(j) Test 11 (14.00% pixel error)

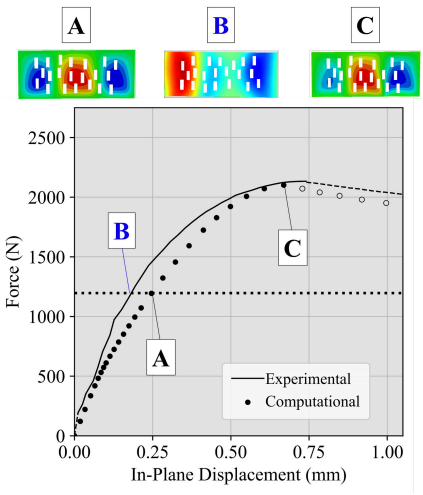


(k) Test 12 (15.0% pixel error)

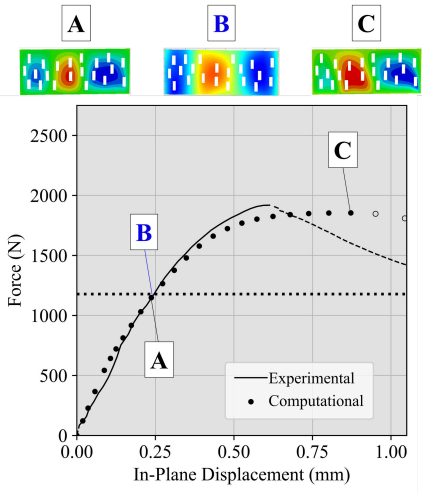


(l) Test 13 (1.67% pixel error)

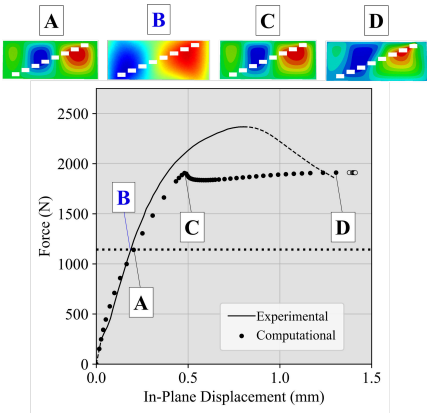
Figure 5.15: Force-Displacement Plots (Tests 8-13)



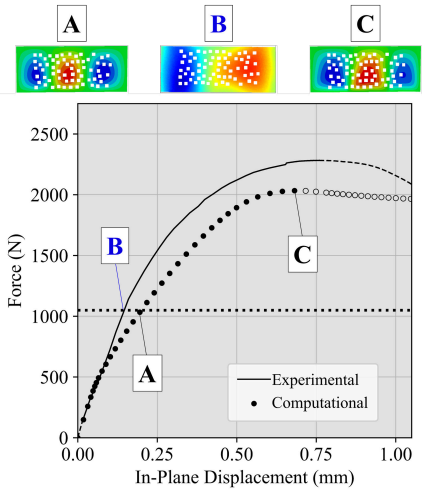
(m) Test 14 (1.80% pixel error)



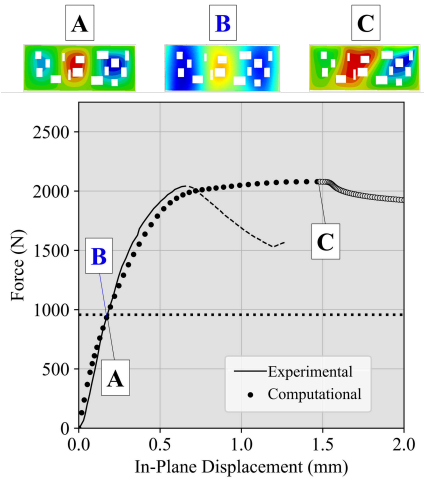
(n) Test 15 (1.13% pixel error)



(o) Test 16 (11.0% pixel error)



(p) Test 17 (0.71% pixel error)



(q) Test 18 (1.29% pixel error)

Figure 5.15: Force-Displacement Plots (Tests 14-18)

5.5.2 Deformation

5.5.2.1 Elastic Failure Mode

Table 5.5 summarises the experimental and computational results for the ultimate capacity ($N_{u,exp}$, $N_{u,FE}$ respectively), ordered by decreasing computationally-determined elastic buckling load ($N_{cr,FE}$). The number of buckling halfwaves expected for an unperforated plate with dimensions a,b (m_0) is compared to the observed number of buckling halfwaves (m_{exp}). The theoretical capacity of an unperforated plate with these dimensions N_0 is also given. Note that Test 10 and Test 6 exhibit much greater deformation towards the loaded edges of the plate (characteristic of LPB), thus m_{exp} is given as two. Finally, [23] presents a method to determine the local buckling failure mode for plates with a regular arrangement of perforations. The dimensions summarised in Table 5.2 are used to determine the predicted mode for all perforation arrangements examined, also given in Table 5.5.

Table 5.5: Predicted deformation mode, critical and ultimate capacity of tests and FE models

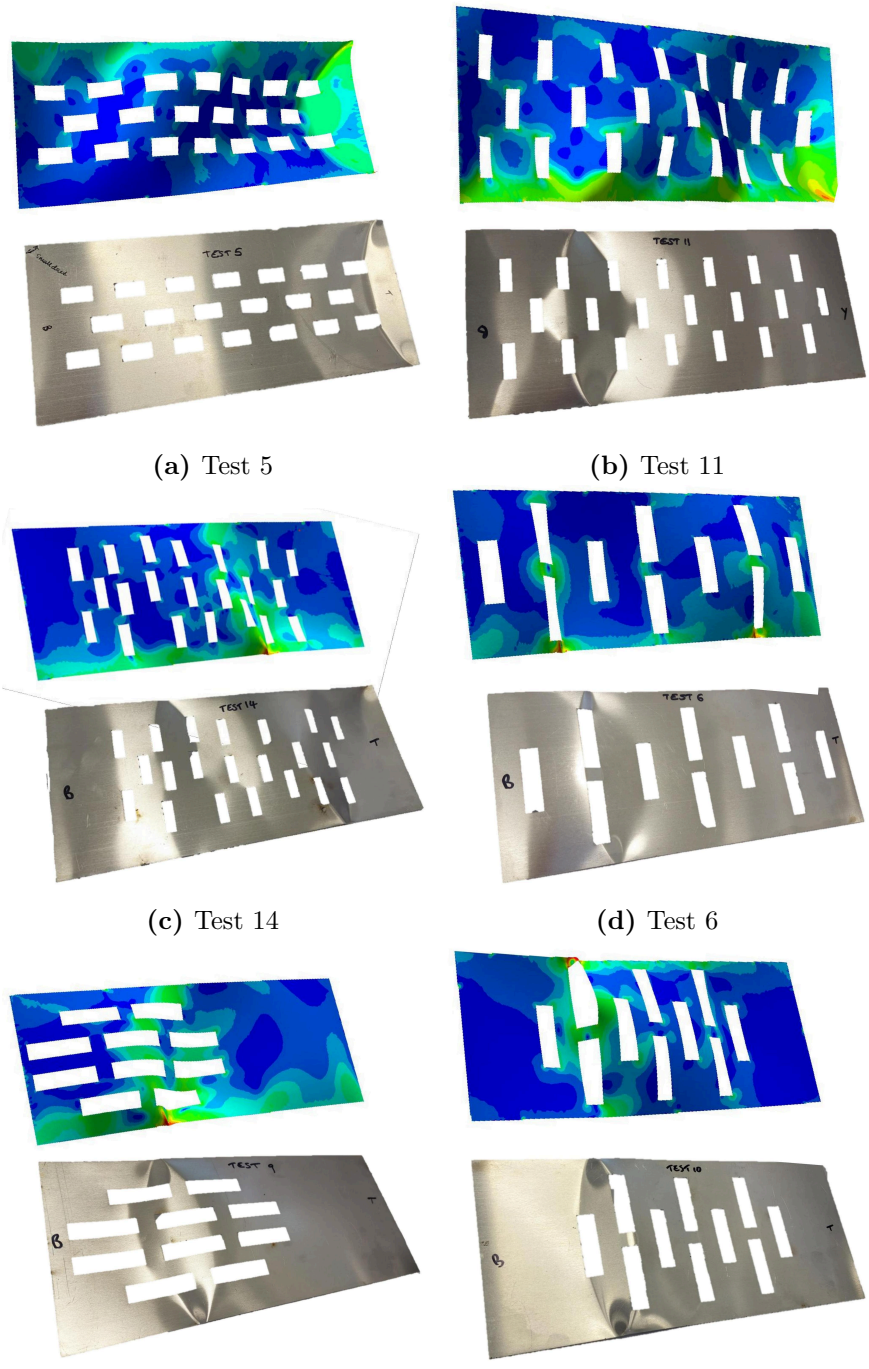
Test Name	Predicted Mode [23]	N_0 (N)	$N_{cr,FE}$ (N)	$N_{u,FE}$ (N)	$N_{u,exp}$ (N)	$\lvert\frac{N_{u,exp}-N_{u,FE}}{N_{u,exp}}\rvert$	$\frac{N_{u,exp}}{N_{u,FE}}$	$\frac{N_{u,FE}}{N_{FE}}$	m_0	m_{exp}	m_{FE}
Test 10	WPB	1429	1667	1554	1884	0.175	1.21	0.93	2	2	3
Test 2	WPB	1418	1458	2509	2244	0.118	0.89	1.72	3	2	3
Test 1	N/A	1441	1441	2196	2475	0.113	1.13	1.52	3	2	3
Test 13	WPB	1414	1289	2350	2295	0.024	0.98	1.82	3	3	3
Test 3	INT	1451	1201	2475	2448	0.011	0.99	2.06	3	3	3
Test 14	WPB	1398	1196	2072	2133	0.029	1.03	1.73	2	3	3
Test 6	WPB	1359	1188	1677	1869	0.103	1.11	1.41	3	3	3
Test 15	WPB	1379	1178	1854	1918	0.033	1.03	1.57	2	3	3
Test 11	WPB	1372	1177	2114	1987	0.064	0.94	1.80	2	3	3
Test 12	WPB	1519	1157	2144	1933	0.109	0.90	1.85	2	3	3
Test 16	INT	1368	1143	1911	2343	0.184	1.23	1.67	3	2	2
Test 4	INT	1463	1111	2446	2126	0.151	0.87	2.20	3	3	4
Test 17	WPB	1454	1049	2033	2261	0.101	1.11	1.94	2	2	3
Test 5	INT	1479	1002	2010	2022	0.006	1.01	2.01	3	3	3
Test 18	WPB	1408	957	2079	2042	0.018	0.98	2.17	2	3	3
Test 7	INT	1333	846	2003	2108	0.050	1.05	2.37	3	3	3
Test 8	INT	1376	823	2175	2002	0.086	0.96	2.54	3	2	2
Test 9	ESB	1455	765	2119	1924	0.101	0.91	2.77	3	2	2
Mean		1420	1150	2100	2110	0.082	1.02	1.90	2.61	2.56	2.89
Standard Dev.		48.2	229	255	192	0.056	0.107	0.440	0.502	0.511	0.471

The initial imperfection greatly influences the elastic buckling modeshape; its shape and relative size should be adequately captured with the computational model, which uses the true geometric imperfection data (Section 5.4.2). However, the number of buckling

halfwaves observed experimentally disagrees with the number observed computationally for tests 2, 4, 10, and 17. Although WPB is seen in the GMNIA results, LPB is observed in the experimental deformation of Test 2 (Figure 5.15(a)). This is attributed to the supports providing a greater rotational fixity than the pinned boundaries modelled. The disparity in behaviour is particularly surprising for Test 17. With the least percentage pixel error (Table 5.4) and the smallest plate aspect ratio investigated (Figure 5.9), its geometry is greatly indicative of two buckling halfwaves (observed experimentally). Thus, more onerous growth in the computational model must be attributed to an over-represented geometric imperfection (see Table 5.3), as observed for the unperforated plate (Section 5.4.4), and discussed for Test 16 in Section 5.5.1. This explains why the GMNIA model underpredicts both the ultimate and linear elastic buckling load (Figure 5.15(p)). Tests 3, 4, 7 and 8 possess a regular arrangement of perforations, and their local buckling mode is well-predicted with the algorithm by [23]; although Test 3 deformation is more characteristic of WPB, an interaction (INT) of WPB and USB modes is clearly observed for Tests 4 and 8 where failure is controlled by the unstiffened strip between the perforations and the plate edge, and Test 7 exhibits an interaction of WPB and ESB modes. ESB deformation is predicted for Test 9, which can be somewhat identified by greater deformation at the group of perforations (Figure 5.15(h)). Specimen exhibiting WPB are expected to buckle in the same number of halfwaves as the unperforated plate ($m_{FE} = m_0$), which is not observed for Tests 10, 14, 15, 11, 12, 17 and 18. Except for Test 10, this is attributed to the factors discussed in Section 5.3.4. The LPB behaviour exhibited by Test 10 (and to some extent Test 6), is not predicted with the algorithm by [23], which must be modified to accommodate plates with a staggered arrangement of perforations.

5.5.2.2 Plastic Failure Mode

As discussed in Section 5.5.1, differences between the experimental and computational behaviour around ultimate failure can be attributed to either the experiment resolution, or GMNIA modelling issues. However, this does not prevent a visual comparison between the experimental and GMNIA post-ultimate deformation, which is adequate for the majority of plates. As the computational model predicts that failure initiates in the same region as the experiments, this gives confidence to the underlying constitutive and geometric modelling assumptions and further supports the validity of the elastic analysis (Figure 5.16). Most plates fail by forming a yield line towards the edge loaded by the actuator (see a typical example for Test 5 in Figure 5.16(a)). However, the plastic deformation in the GMNIA model for Tests 6, 9, 10, 11, 14, 16 and 17 is more localised at the perforations, which is also identified in the experiment, except for Test 11 Figure 5.16(b). This more localised deformation is not indicative of a greater ultimate strength. For example, Tests 9 and 10 attribute the greatest and smallest $N_{u,FE}/N_{cr,FE}$ ratios respectively, and are both characterised by similar localised deformation (Figure 5.16(e-f)).



(a) Test 5 **(b)** Test 11

(c) Test 14 **(d)** Test 6

(e) Test 9 **(f)** Test 10

Figure 5.16: Comparison of plastic deformation in the GMNIA and experimental results

5.5.3 Influence of Perforations on Performance

For a constant total perforated area, orienting perforations with their major axis perpendicular to the loading direction markedly enhances elastic buckling load, provided the holes do not intrude excessively into the 'end panel', which is a region identified by [72]. Tests 6 and 10 achieve substantially higher elastic capacities than Tests 7 to 9, with Test 10 surpassing the unperforated plate capacity by 17% (Table 5.5). However, this increased elastic capacity is at the expense of ultimate strength for which both Test 6 and Test 10 rank in the bottom three (Table 5.5). Notably, the tall and closely spaced perforations in Test 10 cause force concentration in the narrow regions above and below the perforations, preventing the plate from attaining the theoretical elastic buckling load (Section 5.5.2.1). Although this stiffening effect causes LPB behaviour and improves the elastic buckling load, it causes elasto-plastic failure and weakens the ultimate capacity. Conversely, plates exhibiting low elastic buckling loads (Tests 7 to 9) display the highest ratios of ultimate-to-elastic capacity, reflecting a trade off between stiffness and strength (Table 5.5). Plates with a row of perforations along the longitudinal centreline possess the greatest ultimate capacities of all perforated plates examined (Tests 2, 3, 4, Table 5.5). Tests 5 and 7 also feature rows of perforations with no y-eccentricity, and the same total perforation area as Test 4. Comparing these plates show that placing holes on the mid line (Tests 4 to 5) stabilises the elastic response, whereas locating a perforation border along this line (Test 7) reduces elastic buckling load. This outcome arises from superposed stress fields concentrating at a structurally onerous location. To evaluate the influence of geometric randomness on the elastic buckling load, Test 18 is compared to Tests 4 to 10 (which have the same total perforation area, but with uniformly arranged perforations). The average aspect ratio of a perforation is given as:

$$\theta = \frac{\overline{l}}{\overline{h}} \tag{5.2}$$

Where $\overline{l}$ and $\overline{h}$ are the average hole length and hole height respectively. We observe that a lower θ generally correlates with a greater elastic buckling load. For example, Tests 6 and 10 (with $\theta = 0.267$) outperform Tests 4 and 5 (with $\theta = 1.88$), which in turn exceed Tests 7 to 9 (with $\theta = 3.75$). As Test 18 (with $\theta = 0.791$) does not outperform

all tests with a greater average aspect ratio, this suggests that a plate with a regular arrangement of perforations is more structurally efficient. Finally, the effect of pattern superposition is evaluated by comparing Tests 11 to 13. Although the perforated area in Test 13 is less than half that in Test 12, the critical elastic buckling load is not doubled, and all three tests exhibit a similar relative variation in ultimate strength. This demonstrates that the capacity-to-weight ratio is a function that can be optimised for both elastic buckling and ultimate strength.

5.6 Prediction of the Buckling Capacity

In addition to the algorithm for elastic buckling mode identification, [23] proposed a set of equations to determine the elastic buckling load of a plate with a regular arrangement of perforations, not exhibiting LPB. This is assessed in comparison to the elastic buckling load found in the linear perturbation assessment ($N_{cr,FE}$). Depending on whether n_r is greater than or equal to 1, the authors proposed two equations for $\widehat{N}_{cr,h}$ in terms of the unperforated plate elastic buckling load (N_0) and two parameters denoted τ or ω both functions of the perforation layout.

$$\tau = r_p \cdot \left(\left(\frac{S_{yo}}{b} \right)^{-0.407} + \left(\frac{l_y}{b} \right)^{-0.274} + \left(\frac{l_x}{a} \right)^{0.452} \right), \tag{5.3a}$$

$$\widehat{N}_{h,n_r>1} = N_0 e^{-0.714\tau}, \tag{5.3b}$$

$$\omega = \left(\left(e_x + \frac{l_y}{b} \right)^{35.3} + e_y^{1.08} + \left(\frac{l_x}{a} \right)^{0.561} \right)^{2.28}, \tag{5.3c}$$

$$\widehat{N}_{h,n_r=1} = N_0 e^{-1.58\omega} \tag{5.3d}$$

In Equation 5.3(a), the perforation ratio (r_p) is the area of the perforations divided by the area of the plate. In Equation 5.3(c), e_x and e_y are the absolute x- and y- eccentricity of the centroid of the group of perforations about the plate centre respectively. All other variables are summarised in Section 5.3.3.3. Additionally, in [2], Moen & Schafer propose the following set of equations to determine $N_{cr,B}$:

$$N_{cr,B} = \min \left(N_0, N_{cr,B,net} \left(1 - \frac{l_y}{b} \right) \right), \tag{5.4a}$$

$$N_{cr,B,net} = \min(N_{cr,1}, N_{cr,2}), \tag{5.4b}$$

$$N_{cr,i} = k_i \cdot \frac{E\pi^2}{12(1-\nu^2)} \cdot \left(\frac{t}{S_{yo,i}} \right)^2, \tag{5.4c}$$

$$k_i = \begin{cases} 0.425 + \dfrac{0.2}{\left(\frac{l_x}{S_{yo,i}} \right)^{0.95} - 0.6} & \text{if } \dfrac{l_x}{S_{yo,i}} \geq 1 \\ 0.925 & \text{otherwise} \end{cases} \tag{5.4d}$$

Where $i = 1$ or 2, Figure 5.17.

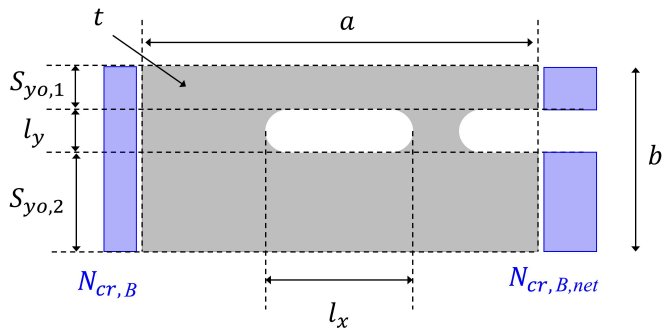


Figure 5.17: Variables considered by [2] reproduced from [23]

The elastic buckling load determined by Equation 5.3(b) and Equation 5.3(d) is termed $\widehat{N}_{cr,h}$, and the prediction determined with Equation 5.4 is termed $N_{cr,B}$. The ratio of the elastic buckling load to that determined in the linear perturbation step ($N_{cr,FE}$) is provided in Table 5.6, and the elastic buckling mode predicted with [23] is provided again for clarity.

Although the prediction given by [2] has a more accurate mean μ than the prediction proposed by [23] ($\mu = 0.969$ and $\mu = 1.15$ respectively), it is unconservative, and exhibits a much greater coefficient of variation (Table 5.6). This variation of Equation 5.4 is evident when the average accuracy is determined for all plates with one row of perforations ($\mu = 1.23$, COV = 0.512) compared to those with multiple rows of perforations ($\mu = 0.889$, COV = 0.403). In contrast, Equation 5.3(d) for all plates with one row of perforations ($\mu = 1.03$ and COV = 0.150) combined with Equation 5.3(b) for all plates with multiple rows of perforations ($\mu = 1.19$ and COV = 0.219) perform better. Equation 5.4 was developed by examining plates with one row of large perforations exhibiting USB, similar to the arrangement of perforations in Tests 2 to 4 and 8. USB deformation dominates interaction modeshape of the latter test (Section 5.5.2.1), however the accuracy of Equation 5.4 is quite poor. As the width of the unstiffened strip between the perforations and the plate edge varies for perforations in a random arrangement (Tests 14, 15, 17 and 18), it was particularly complex to determine the elastic buckling load with Equation 5.4 for these tests. Furthermore, it is significantly over-conservative for Test 16, where the aspect ratio of the unstiffened strip between the perforation and longitudinal edge is very small. Finally, both Equation 5.3 and Equation 5.4 largely underpredict the elastic buckling load of Tests 6 and 10. It is discussed in [23] that Equation 5.3(d) underpredicts the

elastic buckling load of plates exhibiting LPB, which is observed for Tests 10 and 6, where LPB deformation is exhibited although not predicted. An alternative prediction to both Equation 5.3(d) and Equation 5.4 should be used to determine the elastic buckling load of plates with a regular or staggered arrangement of perforations exhibiting LPB.

Table 5.6: Accuracy of elastic buckling load predictions, and efficiency

Test Name	Predicted Mode [23]	$\frac{N_{FE}}{N_0}$	$\frac{\widehat{N}_h}{N_{FE}}$	$\frac{N_{cr,B}}{N_{FE}}$
Test 2	WPB	1.03	1.25	1.20
Test 3	INT	0.827	1.00	0.827
Test 4	INT	0.759	0.939	0.759
Test 5	INT	0.677	1.07	0.677
Test 6	WPB	0.874	1.44	1.36
Test 7	INT	0.635	0.987	0.635
Test 8	INT	0.598	1.05	0.841
Test 9	ESB	0.526	0.8838	0.526
Test 10	WPB	1.17	1.93	1.88
Test 11	WPB	0.858	1.22	0.858
Test 12	WPB	0.762	1.21	0.762
Test 13	WPB	0.912	1.03	0.912
Test 14	WPB	0.855	1.20	0.855
Test 15	WPB	0.854	1.25	0.854
Test 16	INT	0.835	0.915	2.13
Test 17	WPB	0.721	1.12	0.721
Test 18	WPB	0.680	1.14	0.680
Mean		0.789	1.15	0.969
Standard Dev.		0.158	0.247	0.439

5.7 Conclusion

This paper presents the results of uniform compression tests on a range of simply supported perforated thin aluminium plates. Seventeen perforation arrangements are examined including regular, irregular, staggered, and patterned. For each test, the initial imperfection, deformation patterns during loading, and failure modes are carefully measured. A geometrically and materially nonlinear finite element model incorporating the measured imperfections is then compared with all experimental results. Both experimental and computational challenges are described in detail. Good agreement is found between the experimental and computational load-displacement results prior to elasto-plastic behaviour, and it is concluded that the behaviour near and beyond the ultimate capacity can be safely assessed qualitatively. The accuracy of analytical predictions of the elastic buckling load by [23] and [2] is then assessed by comparing it with the numerical linear perturbation results from the verified finite element (FE) model. Firstly, it is observed that the plates examined lie at the limiting aspect ratio for the transition from two to three buckling halfwaves, however they do not always buckle into the number expected. Therefore, the aspect ratio defining this transition point for a perforated plate is different to that of an unperforated plate; likely related to the type of local buckling mode. Secondly, this study demonstrates that the algorithm developed by [23] can determine the type of local buckling mode based on the perforation dimensions and pattern with good accuracy. However, some modification is required to accommodate tall overlapping staggered holes, which are seen here to exhibit Lateral Perforation Buckling (LPB). Last, the theoretical predictions for the elastic buckling load $\widehat{N}_{cr,h}$ of a plate with $n_r > 1$ (Equation 5.3(b)) and $n_r = 1$ (Equation 5.3(d)) proposed by [23], and $N_{cr,B}$ (Equation 5.4) proposed by [2] are compared to the computed (FE) values, $N_{cr,FE}$. The former is found to exhibit a greater accuracy with little variation for both subsets of perforation arrangements ($n_r > 1$ and $n_r = 1$). However, neither prediction should be used to determine the elastic buckling load of plates exhibiting LPB, which they significantly underpredict. In terms of their influence on performance, perforations oriented perpendicular to the load-axis and with a low aspect ratio lead to the greatest elastic plate buckling capacity, albeit at

the expense of the ultimate strength due to elasto-plastic failure. Additionally, a single row of perforations organised with no y-eccentricity can improve both the elastic and ultimate strength (in comparison to perforated plates with similar perforation ratios), provided that the hole border is not situated along the plate centreline. Furthermore, comparing the performance of plates with superposed patterns demonstrates that critical elastic buckling load is not proportional to the perforation area.

6

The Direct Strength Method for Plates with Perforations

6.1 Commentary

The Design Strength Method (DSM) is a standard method for estimating the ultimate capacity of cold-formed plates. However, the DSM does not account for perforations. Chapter 4 [23] developed two empirical expressions to predict the elastic buckling load of a perforated plate $N_{cr,h}$, which were benchmarked against experiments conducted in Chapter 5 [25]. These experiments showed that perforations also influence ultimate capacity $N_{u,h}$. The ultimate capacity is the maximum post-critical strength obtained by slender plates, due to stress redistribution (Section 1.3.3). However, [4, 34] showed that perforations affect the slenderness limits imposed by [18] (Section 2.2.3). Moen & Schafer (2011) [113] and Yao & Rasmussen (2017) [5] make different modifications to known DSM equations, to accommodate members with perforations (Section 2.5.1). However, both studies use methods that were shown to be unconservative or inaccurate for perforated plates in Chapter 4 [23]. Additionally, neither study considers the influence of local buckling mode shapes on $N_{cr,h}$. This may be significant, as [114] suggested that local buckling can alter the deformation pattern sufficiently to invalidate the applicability of DSM to columns governed by local buckling. Thus the classification of local buckling modes presented in

Chapter 4 [23] could aid the derivation of a DSM equation that is applicable to plates with a regular arrangement of perforations.

In this chapter, the experiments conducted in Chapter 5 [25] are first compared to the prediction found using DSM with Moen & Schafer (2011) [113] using Smith & Moen (2014) [3], and separately Yao & Rasmussen (2017) [5], then used to calibrate a GMNIA model. The number and size of perforations in both regular and staggered arrangements are varied randomly to produce 300 FE models. The failure mode is shown to be governed by a geometric parameter Ω, distinguishing between two distinct types of ultimate behaviour. A modified slenderness $\lambda_{l,m}$ is introduced, revealing a strong relationship with the ultimate-to-net section yield strength ratio for plates typically exhibiting WPB or an interaction of local buckling modes. Given in terms of a parameter κ, a DSM equation is derived using least-squares regression on the FE results. This expression can accommodate both failure types with a mean accuracy $\mu = 1.01$, $COV = 0.154$, and a good overall R^2 performance of 0.777, demonstrating a significant improvement to existing methods.

6.2 Introduction

Large, widely spaced perforations are commonly used in the webs of beams to enable weight reduction or the passage of services without compromising flexural stiffness. In contrast, rack sections typically feature multiple small perforations across the web and flange width to accommodate the attachment of other members. The relationship between elastic plate buckling capacity, perforation geometry and their arrangement is complex, but notable progress has been made in recent years toward developing predictive models [2–4, 23, 26, 42]. Despite these advances, the ultimate capacity of compressed plates with perforations remains underexplored. In practical design scenarios, engineers usually rely on the finite element method (FE) to estimate these capacities. One simple yet reliable design tool (and a widely adopted approach for cold-formed steel members) used to approximate the ultimate capacity with a 'theoretical' or 'nominal' capacity is the Direct Strength Method (DSM). The DSM gained popularity among structural engineers and is today incorporated by design standards internationally [7, 22, 112](Section 2.5.1). However, its direct application to cross-sections with webs containing perforations is still under

development [5, 113, 114]. In contrast to the single, well-defined local elastic buckling mode of an unperforated plate (N_0), perforated plates exhibit four distinct local buckling modes, as categorised in Chapter 4 [23]. These are Whole Plate Buckling (WPB), Lateral Perforation Buckling (LPB), Unstiffened Strip Buckling (USB) and Euler Strip Buckling (ESB) (Figure 5.2). It is also identified in [23] that plates can exhibit an interaction (INT) of these modes, akin to local-distortional buckling of open sections. Although these modes have been recognised in previous research, recent attempts for a set of perforated plate DSM equations have not accounted for these different failure modes, and so the relationship between perforated plate slenderness and local plate buckling mode has not been fully characterised. To address this shortfall, this chapter presents an integrated experimental and computational study of simply-supported perforated plates subject to uniform compression. The objective is to discover a more accurate DSM equation for predicting the ultimate buckling capacity of such plates. By systematically investigating the influence of perforation geometry and distribution, we aim to extend the DSM to better support the design of perforated structural elements.

6.3 Applicability of Existing DSM Equations to Experimental Results

The results of 17 buckling tests on simply supported perforated plates are reported in Chapter 5 [25]. The plates are made of aluminium with a Young Modulus of 60.0 GPa, and a Poisson's ratio of 0.33, and are perforated with a regular, staggered or irregular arrangement. this chapter focuses on a selection of 11 plates with a regular or staggered array of perforations. Each pattern is characterized by their hole length and width (l_x and l_y), the number of perforations along the 'critical length' (n_x and n_y), the perforation-to-perforation spacing (s_x and s_y), and the minimum spacing between a perforation and the plate's edge (S_{xo} and S_{yo}), see Figure 6.1 to Figure 6.3. The critical length refers to the narrowest section in the longitudinal (x) or transverse (y) direction, see Figure 6.2 and Figure 6.3.

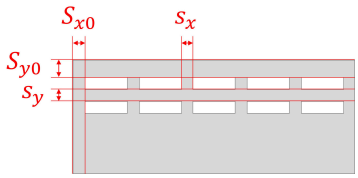


Figure 6.1: An example of a plate with a regular arrangement of perforations (Test E)

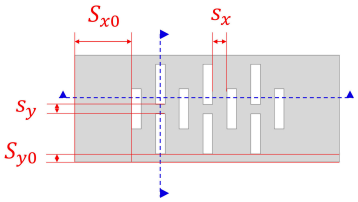


Figure 6.2: A critical length and a critical width in a staggered arrangement of perforations (Test I)

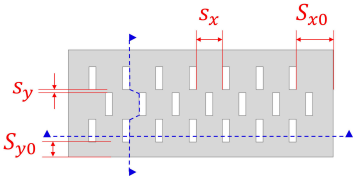


Figure 6.3: A critical length and a critical width in a staggered arrangement of perforations (Test J)

Table 6.1: Experimentally-found ultimate strengths (in Newtons) of tested aluminium perforated plates

Test	Mode	a (mm)	b (mm)	l_y (mm)	l_x (mm)	n_t	n_l	t (mm)	t_r (mm)	$N_{cr,l,k}$ (N)	$N_{cr,l,tr}$ (N)	$N_{u,SM}$ (N)	$N_{u,YR}$ (N)	$N_{u,exp}$ (N)
A	LPB	289	116	48	45	1	1	0.895	0.781	1260	932	2284	2261	2244
B	INT	289	117	24	45	1	2	0.903	0.778	1282	917	2778	2795	2448
C	INT	288	113	24	45	1	5	0.898	0.576	1115	382	1928	2716	2126
D	INT	290	117	12	45	2	5	0.878	0.568	1081	356	1913	2662	2108
E	USB	290	115	12	45	2	5	0.885	0.568	1119	360	1908	2674	2002
F$^\diamondsuit$	INT	287	117	12	22.5	3	7	0.908	0.522	1182	274	1623	2586	2022
G$^\diamondsuit$	LPB	290	117	45	12	2	7	0.885	0.427	1019	180	695	1223	1869
H$^\diamondsuit$	ESB	288	116	12	45	4	3	0.902	0.514	1132	266	1430	2283	1924
I$^\diamondsuit$	LPB	285	117	45	12	2	7	0.898	0.421	1063	147	644	1244	1885
J$^\diamondsuit$	WPB	279	116	24	8	3	7	0.887	0.605	1163	435	1309	1685	1987
K$^\diamondsuit$	WPB	291	117	8	8	3	7	0.897	0.804	1310	1017	2880	2763	2295

$\diamondsuit$ The perforations are staggered. n_l and n_t are the number of perforations over the 'critical length' and 'critical width', taken as the narrowest length in the longitudinal and transverse direction respectively, see Figure 6.2 and variables are defined in Figure 1.4

Table 6.1 summarises the perforation arrangement and their dimensions for each test. Tests A to E feature a regular perforation arrangement with non-overlapping rows and columns. Tests F to K have a staggered arrangement, where perforations may overlap in either direction. The mode observed in the tests conducted by [25] is also given in Table 6.1. Despite a geometrical slenderness (b/t) of 130, Test I (Figure 6.2) was the only specimen to possess an ultimate capacity less than the elastic buckling load – indicating that this arrangement of perforations led to failure by yielding. The authors also found that most plates failed with a yield line located towards the edge by the actuator. Several plates formed yield lines local to the perforations; however, this was not found to dictate

the ultimate capacity. The ultimate capacity obtained in each experiment, $N_{u,exp}$, is provided in Table 6.1. For comparison, we also include in Table 6.1 the nominal ultimate capacity determined with Equation 2.8 based on [113] with [3], namely $N_{u,SM}$ (subscript denoting 'Smith & Moen'). There is no global buckling and thus we consider $N_{u,e} = N_{y,net}$, giving λ_l in terms of the net section yield capacity:

$$\lambda_l = \sqrt{\frac{N_{y,net}}{\min(N_{cr,l,k}\,,\, N_{cr,l,tr})}} \tag{6.1}$$

Where $N_{cr,l,tr}$ is found with a FSM programme (e.g. CUFSM [149]) with a reduced thickness given by Equation 2.4 and $N_{cr,l,k}$ is found with Equation 2.5. Lastly, the nominal ultimate capacity $N_{u,YR}$ (subscript denoting 'Yao & Rasmussen') predicted with [5] and determined with Equation 2.8 is also provided in Table 6.1, using our consideration that $N_{u,e} = N_{y,net}$, and the author's assumption that $\widehat{N}_{cr,h} = N_0$, λ_l is given by:

$$\lambda_l = \sqrt{\frac{N_{y,net}}{N_0}} \tag{6.2}$$

As $N_{u,e} = N_{y,net}$ is considered, note that both Equation 6.1 and Equation 6.2 are 'modified' versions of the equations proposed by the authors. In Figure 6.4, these predictions are normalised by $N_{y,net}$ and plotted against the computational local plate buckling slenderness $\lambda_{l,h}$, determined as:

$$\lambda_{l,h} = \sqrt{\frac{N_{y,net}}{N_{cr,FE}}} \tag{6.3}$$

Where $N_{cr,FE}$ is the elastic buckling load determined in a linear perturbation step with ABAQUS [150].

Both approaches are compared to the 11 experimental results (i.e., with $N_{u,h} = N_{u,exp}$) in Figure 6.4, and the DSM local buckling curve is shown (Equation 2.8). As expected, there is a decreasing relationship between the capacity ratio and the slenderness of each perforated plate Figure 6.4. Furthermore, neither modified approach can adequately determine the ultimate capacity of the tested plates. Table 6.1 shows that the capacity determined with the reduced thickness obtained with CUFSM ($N_{cr,l,tr}$) is always limiting to $N_{u,SM}$, potentially demonstrating that the reduced thickness calculation is incorrect (as proposed by [114]). The top 5 most accurate predictions for $N_{u,h}$ with [3] are for each of

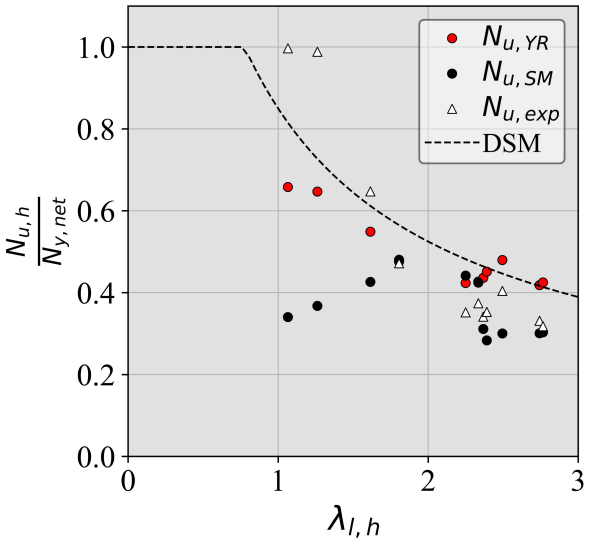


Figure 6.4: Nominal ultimate capacity determined with current DSM approaches compared to experimental result

the experimental tests with a regular arrangement of perforations. Considering only these 5 experiments, the mean (μ) and coefficient of variation (COV) for the $N_{u,exp} - to - N_{u,SM}$ is more accurate than $N_{u,exp} - to - N_{u,YR}$ ($\mu = 1.02$, COV= 0.090 and $\mu = 0.838$, COV= 0.117 respectively). However, the ultimate capacity for plates with tall, staggered perforations (Test G and Test I) is underestimated by over 50% with $N_{u,SM}$, and the capacity of Test K with small, staggered perforations is overestimated by over 25%. Additionally, for the complete set of 11 experiments, including staggered perforations, the accuracy of $N_{u,exp} - to - N_{u,SM}$ reduces, with $\mu = 1.42$, COV=0.504, compared to $N_{u,exp} - to - N_{u,YR}$ with $\mu = 0.988$, COV = 0.294. It follows that the modified prediction for the ultimate capacity given by [113] with [3] ($N_{u,SM}$) is not applicable to plates with a staggered arrangement of perforations. Therefore, it is proposed to investigate the ultimate capacity predicted with the modified approach based on [5] for the remainder of this chapter, whereby $N_{y,net}$ replaces $N_{u,e}$ in λ_l.

6.4 New DSM Equations for Regular and Staggered Perforations

6.4.1 GMNIA Model

The authors conducted a series of tests on perforated aluminium plates, for which a GMNIA FE model was validated using the measured material properties and geometric imperfections [25]. The full-range stress-strain mode originally proposed by Ramberg-Osgood was modified by [151] for stainless steels, and later extended to cold-formed steel [152]. This model is widely accepted in design standards [136, 153], given by the following equation for $f_y < \sigma \leq f_u$:

$$\epsilon = \frac{f_y}{E} + 0.002 + \frac{\sigma - f_y}{E_y} + \epsilon_u \cdot \left(\frac{\sigma - f_y}{f_u - f_y}\right)^m \tag{6.4}$$

More recently, [154] found that aluminium alloys can be effectively modelled with the following two-stage formulation also based on Ramberg-Osgood's original model:

$$\epsilon = \begin{cases} \dfrac{\sigma}{E} + 0.002\left(\dfrac{\sigma}{f_y}\right)^2 & \text{for } \ 0 < \sigma \leq f_y \\ \dfrac{\sigma - f_y}{E_{0.2}} + \left(\epsilon_u - \epsilon_{0.2} - \dfrac{f_u - f_y}{E_{0.2}}\right) \cdot \left(\dfrac{f - f_y}{f_u - f_y}\right)^m & \text{for } \ f_y < \sigma \leq f_u \end{cases} \tag{6.5}$$

$E_{0.2}$ is the tangent modulus given by [152, 154]:

$$E_{0.2} = \frac{E}{1 + 0.002\frac{En}{f_y}} \tag{6.6}$$

Where n is the strain hardening exponent [152, 154]:

$$n = \frac{\ln(4)}{\ln\left(\frac{f_y}{\sigma_{0.05}}\right)} \tag{6.7}$$

To capture the degree of nonlinearity of the second stage of the stress-strain curves for aluminium alloys, the exponent m is given by [152, 154]:

$$m = \frac{\ln\left(0.008 + \frac{\sigma_1 - f_y}{E} - \frac{\sigma_1 - f_y}{E_{0.2}}\right) - \ln\left(\epsilon_u - \epsilon_{0.2} - \frac{f_u - f_y}{E_{0.2}}\right)}{\ln(\sigma_1 - f_y) - \ln(f_u - f_y)} \tag{6.8}$$

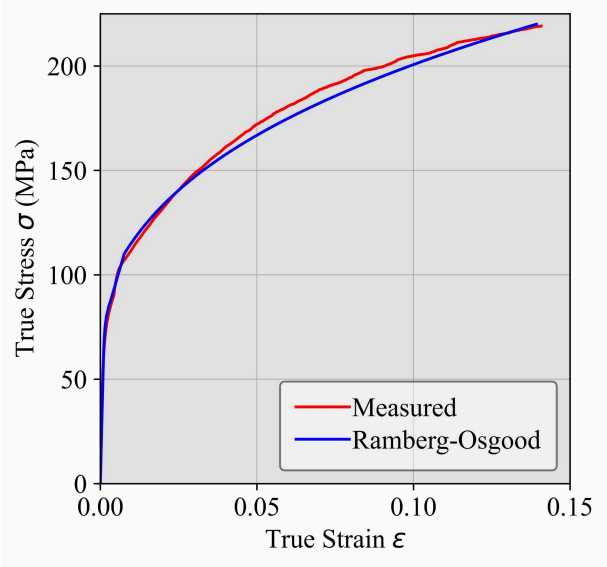


Figure 6.5: Ramberg-Osgood model fitted to measured data

Table 6.2: True stress-strain values

	0.05%	0.2%	1%	2%	Ultimate
ϵ	0.00174	0.00345	0.01177	0.02215	0.141
σ (MPa)	78.8	88.0	112	138	219

The values computed for our aluminium plates are summarised in Table 6.2 and shown in comparison to the true stress-stain data measured in a tensile test Figure 6.5, see [25]. The values $E_{0.2} = 3300$ MPa, $n = 12.6$ and $m = 2.59$ align with those reported by [154]. The results confirm that our aluminium adheres to the two-stage Ramberg-Osgood model proposed by [154]. The GMNIA FE model developed by [25] for the buckling of aluminium perforated plates can be used in this study for cold-formed steel, as it has been shown that their stress-strain response is very similar. In this study, the Young's Modulus (E) was taken in accordance with parametric investigations conducted on the elastic buckling load of perforated plates ([23]), and the 0.2% yield stress (f_y) and ultimate strength (f_u) agree with the values considered by [116]. The exponent n is recommended by design guidance [155], and the exponent m and ultimate strain ϵ_u are in accordance with [152]. These values are summarised in Table 6.3.

Table 6.3: Material parameters considered in FE models

E (MPa)	f_y (MPa)	f_u (MPa)	$E_{0.2}$ (MPa)	n	m	ϵ_u
203400	450	500	24700	8	3.97	0.060

Table 6.4: Control parameters considered in FE models

Method	Initial Step Size	Min Step Size	Max Step Size	Total Period	Max Disp. (mm)
Newton-Raphson	0.001	0.000001	0.015	1	1

The validated model was used to develop the database of 300 GMNIA models considered herein, for cold-formed steel with eigenmodal imperfection. Load is applied as a uniform compressive displacement to the plate width, in accordance with [140] to adequately represent the post-ultimate load drop [156]. This procedure is also more able to replicate the elasto-plastic redistribution of stresses. The Newton-Raphson solution scheme recommended by [140] and [116] is considered in this research for comparison. This method is selected because ABAQUS does not provide a stabilisation algorithm for the Riks Arc Length-Method, which is not representative of the true buckling behaviour when localised instabilities are present [140]. The control parameters given in Table 6.4 are recommended by [140]. The 'adaptive damping factor' was used in conjunction with the Newton-Raphson solver and could automatically vary over the duration of the incremental step [157]. Up to three models were considered: (1) an undamped model, (2) a model with adaptive damping, and (3) a model with adaptive damping and time incrementation parameter refinement. Successive models were only considered if the previous model(s) could not converge (see Appendix F). Time incrementation refinement adjusts how ABAQUS automatically selects, increases or decreases load increments during the nonlinear analysis; if both the previous two models did not converge, these parameters were modified (see Table 6.5) [122]. The geometric imperfection was taken as the first eigenvalue obtained in a linear perturbation assessment, with an amplitude equal to $\min(a/200, b/200)$, as per [18]. Default field parameters are considered, which govern the convergence tolerances and iteration limits

for solving nonlinear field equations such as displacements (Table 6.6) [122]. ABAQUS 2022 is used in this investigation, which implements solver stabilisation improvements and the migration of node-based surface couplings to a new constraint entity [158].

Table 6.5: Time incrementation parameters considered in FE models

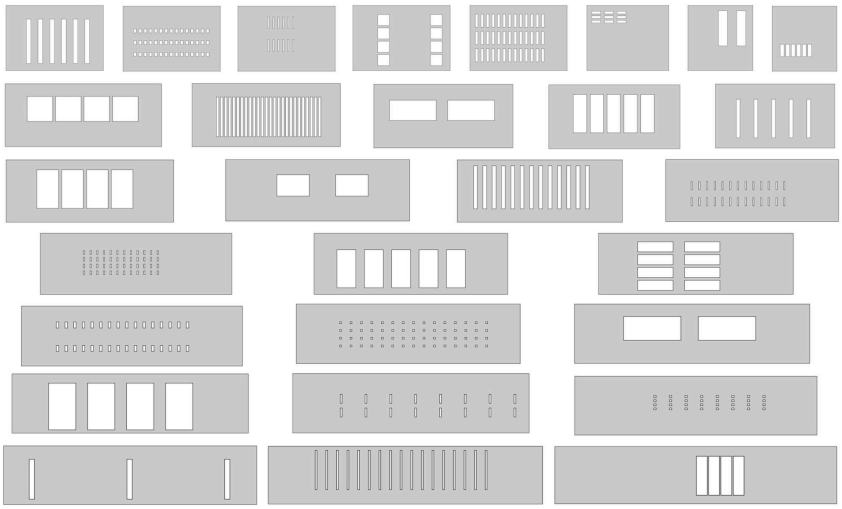
Model	I_O	I_R	I_P	I_C	I_L	I_G
1,2	4	8	9	16	10	4
3	16	18	20	40	30	6

Table 6.6: Default field parameters considered in FE models

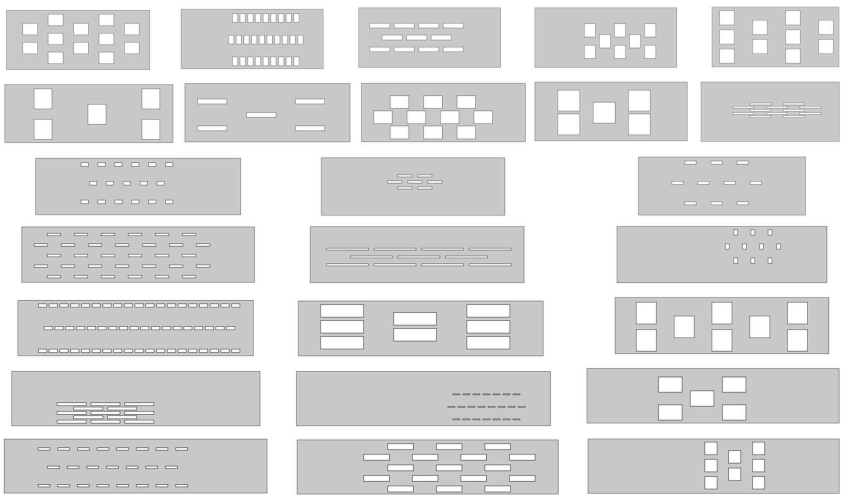
R_n^α	$\overline{q_0}^\alpha$	R_P^α	ϵ^α	C_ϵ^α	R_t^α
0.005	0.01	0.02	0.00001	0.001	0.001

6.4.2 Study Domain

A subset of the 200 models with a regular arrangement of perforations and 100 models with a staggered arrangement of perforations are displayed in Figure 6.6. It is our intention to ascertain a DSM model for plates with any arrangement of perforations, accommodating bespoke cladding and rack sections, and holes made within webs or flanges of beams or columns for maintenance or services which frequently lack practical arrangements or symmetry.



(a) Examples of models with a regular arrangement of perforations



(b) Examples of models with a staggered arrangement of perforations

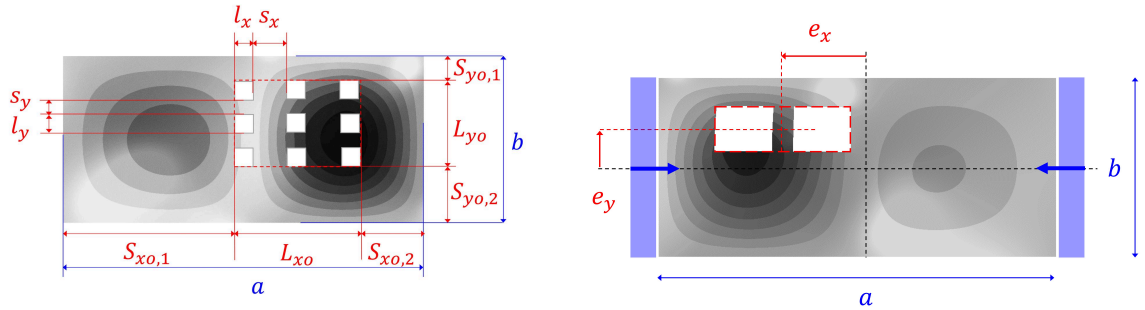
Figure 6.6: Examples of GMNIA models examined in this study

6.4.2.1 Regular Perforations

The parametric analysis follows [23]. Rectangular holes perforate an area of dimensions L_{xo} and L_{yo}, arranged in n_r rows and n_c columns, eccentric relative to the plate centroid (Figure 6.7(a)). Eccentricities in the x- and y- directions are defined as the ratios of the centroid coordinates, e_x, e_y, to the plate width b and length a, respectively (Figure 6.7(b)). Plates are uniformly compressed over b with the load-axis along a (Figure 6.7(b)). The minimum distance from this region to the plate edges are S_{xo} and S_{yo}, where $S_{\xi o} = \min(S_{\xi o,1}, S_{\xi o,2})$. All plates have a width, b, of 100 mm, and the variation domain of these geometrical variables is given in Table 6.7.

Table 6.7: Domain of variables examined in parametric study for regular perforations (see Figure 6.7 for the variable definitions)

	b (mm)	a/b	b/t	n_r	n_c	S_{xo}/a	S_{yo}/b	L_{xo}/a	L_{yo}/b	s_x/a	s_y/b	l_x/a	l_y/b	e_x	e_y
Mean	100	2.70	108	1.82	7.47	0.15	0.19	0.48	0.41	0.06	0.28	0.09	0.29	0.11	0.11
Minimum	100	1.00	100	1.00	2.00	0.05	0.05	0.15	0.05	0.01	0.02	0.00	0.02	0.00	0.00
Maximum	100	5.32	333	4.00	29.0	0.40	0.46	0.79	0.89	0.46	0.89	0.35	0.89	0.35	0.36



(a) Regular variables examined, adapted from [23]

(b) Eccentricity of perforations and load axis, adapted from [23]

Figure 6.7: Relevant variables

6.4.2.2 Staggered Perforations

In comparison to the parametric analysis outlined for plates with a regular arrangement of perforations in Section 6.4.2.1, modification to the definition of various parameters must be made for plates with a staggered arrangement of perforations. Rectangular holes again perforate an area of dimensions L_{xo} and L_{yo}, and are uniformly compressed over b with the load-axis along a as before (Figure 6.7). Perforations are arranged in n'_r rows and n'_c columns, where n'_i is the maximum number of perforations over the 'critical length' (shown dotted for each stagger style in Figure 6.8). Similarly, the spacing s_x and s_y is the least spacing between perforations over the critical length (Figure 6.8), and the plate width is also varied. There are three possible ways by which the perforations can be arranged: isolated perforations where the edges do not overlap in either direction (ISO), perforations overlapping in the y-direction (OL-y), and perforations overlapping in the x-direction (OL-x) (Figure 6.8). The domain of variables for plates with a staggered arrangement of perforations is summarised in Table 6.8.

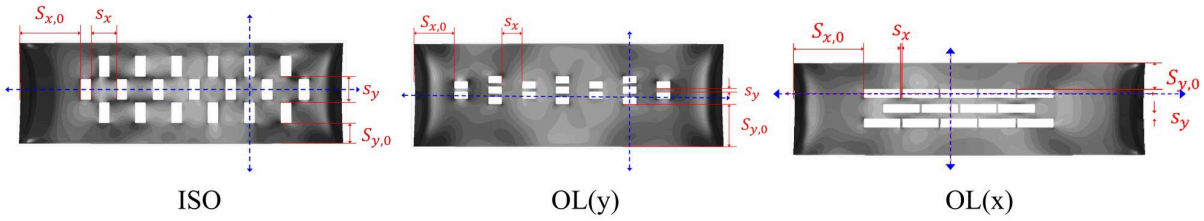
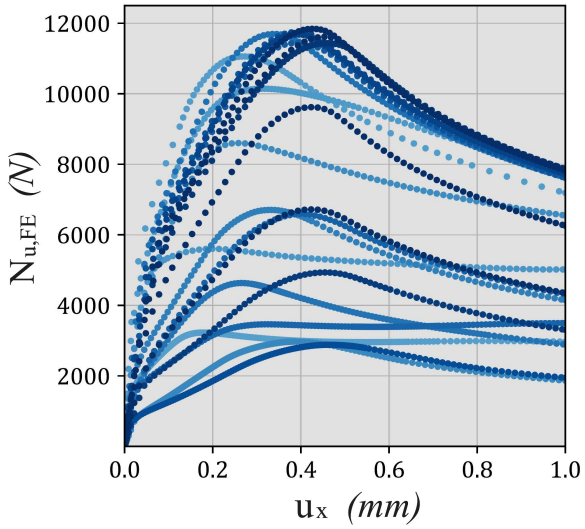


Figure 6.8: Staggered arrangements of perforations examined

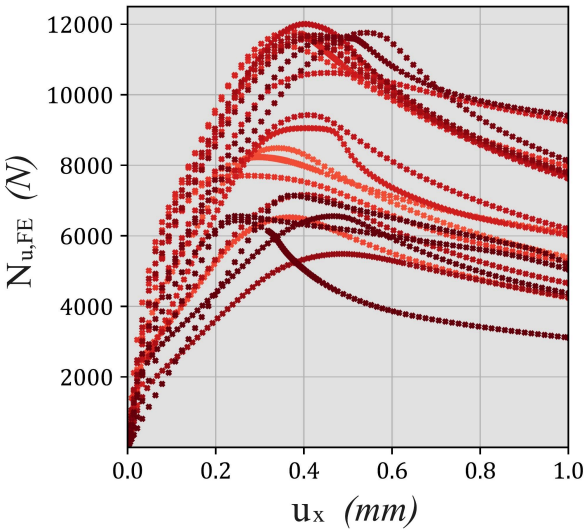
Table 6.8: Domain of variables examined in parametric study for staggered perforations (see Figure 6.7 for the variable definitions)

	b (mm)	a/b	b/t	n'_r	n'_c	S_{xo}/a	S_{yo}/b	L_{xo}/a	L_{yo}/b	s_x/a	s_y/b	l_x/a	l_y/b	e_x	e_y
Mean	97.0	3.39	104	2.99	4.42	0.130	0.138	0.541	0.577	0.076	0.165	0.092	0.121	0.099	0.073
Min	75.0	2.00	53.3	2.00	2.00	0.043	0.038	0.158	0.155	0.004	0.020	0.016	0.010	0.00	0.00
Max	100	6.92	200	5.00	19.0	0.372	0.375	0.898	0.894	0.415	0.665	0.273	0.415	0.348	0.320

6.4.3 Numerical Results



(a) Plates with a regular arrangement of perforations



(b) Plates with a staggered arrangement of perforations

Figure 6.9: Load-displacement curves

Examples of the load-displacement curves are shown for a random selection of 44 plates, half with regular (Figure 6.9a) and half with staggered perforation arrangements (Figure 6.9b). Observe that the peak point is well-approximated due to the proximity of computational datapoints at the maximum load capacity. The capacity ratio (i.e., numerically obtained ultimate capacity divided by the net section yield capacity) is plotted against the perforated

slenderness $\lambda_{l,h}$ (Equation 6.3). This is found for all 200 regular (Figure 6.10) and 100 staggered (Figure 6.11) GMNIA models, in addition to the 11 experimental results as found in the parallel study [25].

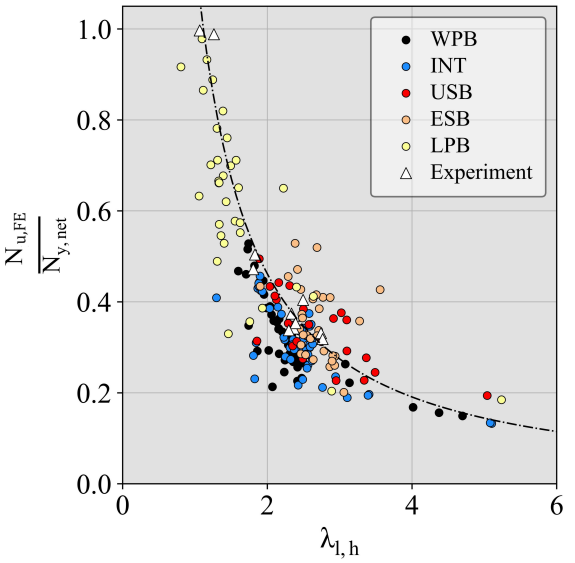


Figure 6.10: Relationship between capacity ratio and perforated slenderness ($\lambda_{l,h}$), for plates with a regular arrangement of perforations. The points are coloured by their local plate buckling mode (see Figure 4.1)

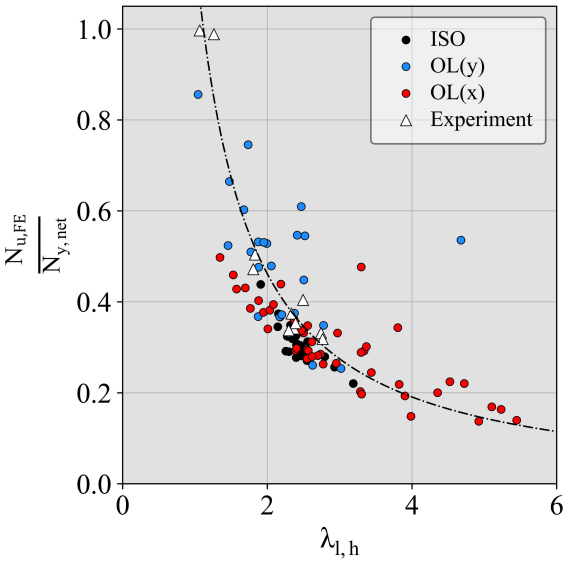


Figure 6.11: Relationship between capacity ratio and perforated slenderness ($\lambda_{l,h}$), for plates with a staggered arrangement of perforations. The points are coloured by their staggered arrangement (see Figure 6.8)

The results for plates with a regular array of perforations are coloured according to their local plate buckling modes, identified with the algorithm proposed in [23]. As no similar process for local plate buckling mode identification exists for plates with a staggered array of perforations, these results are coloured according to the geometrical subdivision shown in Figure 6.8. The trendline in each figure is found by the least square regression of the 11 experimental data. The two datasets follow similar trends as expected, although there is greater scatter for plates with a regular arrangement of perforations.

In [116], the authors showed that the ratio $N_{u,FE}/N_{y,net}$ exhibits a more stratified relationship with $\sqrt{N_{y,net}/N_0}$ than $\sqrt{N_{y,net}/N_{cr,h}}$. Building on from this expression, we introduce a 'modified slenderness' $\lambda_{l,m}$, defined as the multiple of $\sqrt{N_{y,net}/N_0}$ and the geometric slenderness $\sqrt{t/b}$, which captures the geometric stiffness of the plate. This physically-motivated factor empirically collapses the data, exhibiting a greatly reduced

scatter for both datasets (see Figure 6.12 and Figure 6.13 for regular and staggered perforations respectively). Mathematical reasoning is provided in Appendix G.

$$\lambda_{l,m} = \sqrt{\frac{t}{b}} \cdot \sqrt{\frac{N_{y,net}}{N_0}} \tag{6.9}$$

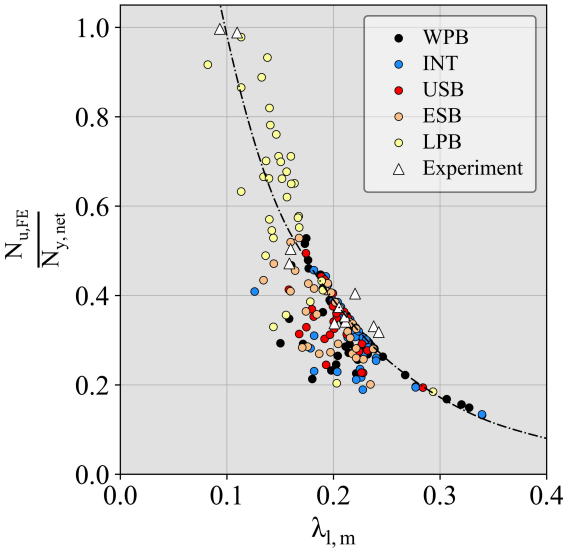


Figure 6.12: Relationship between capacity ratio and modified slenderness ($\lambda_{l,m}$), for plates with a regular arrangement of perforations. The points are coloured by their local plate buckling mode (see Figure 4.1)

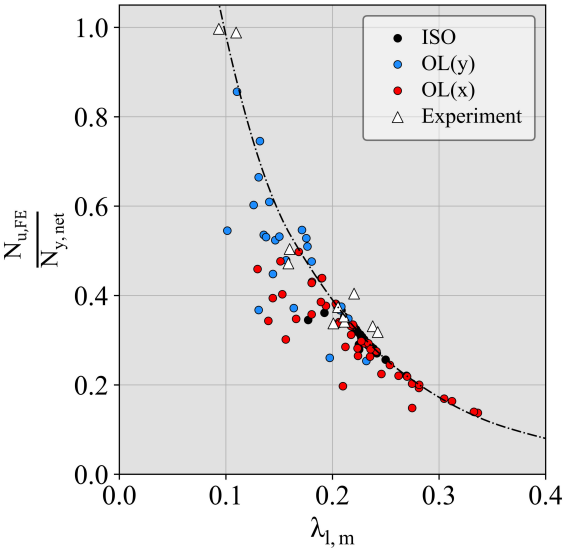


Figure 6.13: Relationship between capacity ratio and modified slenderness ($\lambda_{l,m}$), for plates with a staggered arrangement of perforations. The points are coloured by their staggered arrangement (see Figure 6.8)

Noteworthy, plates exhibiting LPB possess a greater strength than that predicted by the trendline for the experimental data (Figure 6.12). Additionally, the slenderness of OL(y) geometries is typically the least, and that of OL(x) is typically the greatest; otherwise, there is no distinction in behaviour for plates with a different geometrical arrangement of staggered holes (Figure 6.13).

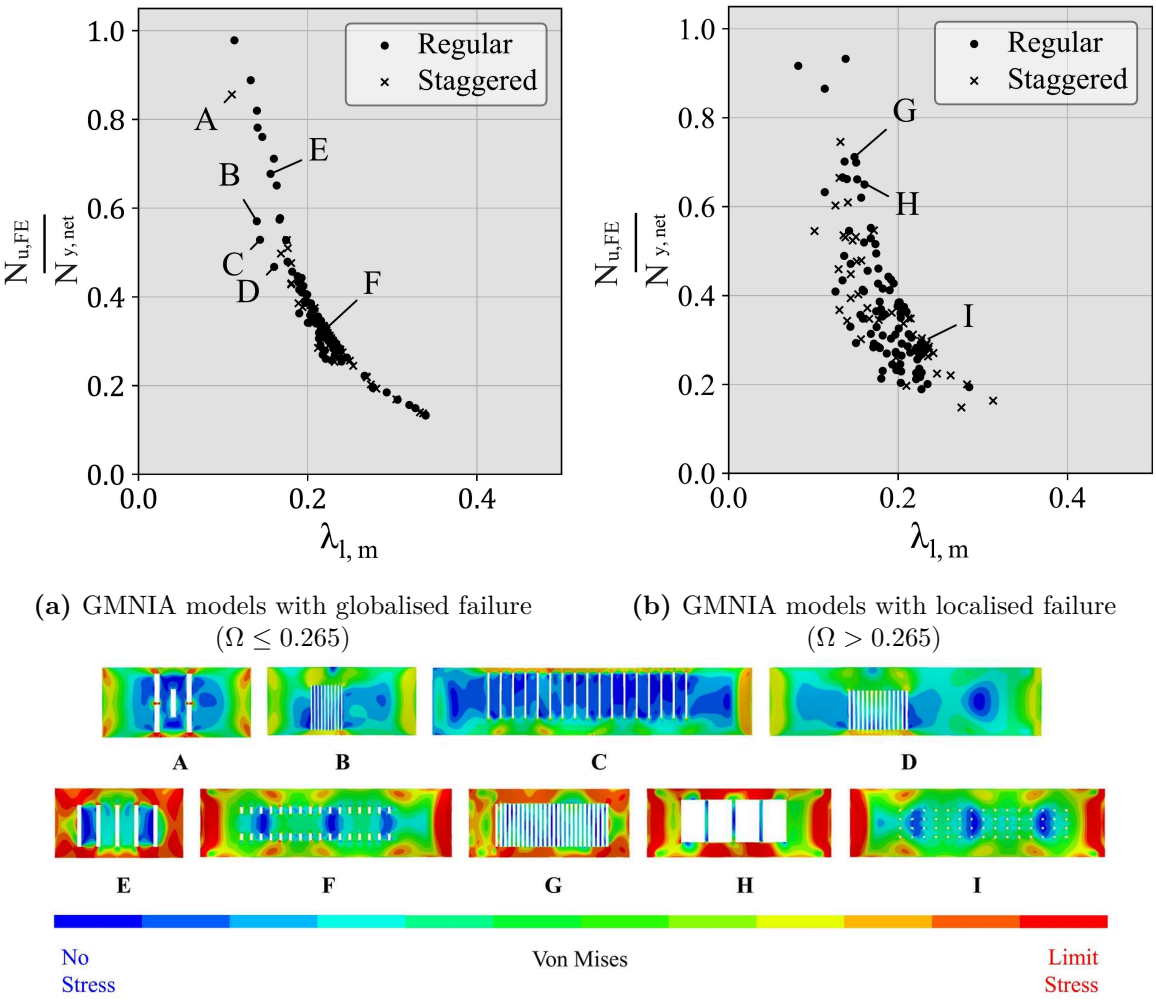
Chapter 4 showed that the elastic buckling load of globalised modes (WPB and INT), is predicted more reliably than that of localised modes, (USB, ESB and LPB). It follows that some additional parameter is required to distinguish between geometries whose ultimate response remains governed by a plate-like deformation, and those for which collapse becomes increasingly governed by localised ligament behaviour. Previous work

has shown that the perforation length l_x, perforation height l_y, transverse edge ligament S_{yo}, and longitudinal edge ligament S_{xo} are particularly important in distinguishing these behaviours (Chapter 4, Chapter 5). Furthermore, classical plate theory establishes that the aspect ratio of a plate governs its local buckling response.

Therefore, the aspect ratios of plate regions local to a perforation are expected to possess physical meaning. In particular, l_x/S_{yo} describes the tendency of the transverse ligament above or below a perforation to behave as an unstiffened strip, while l_y/S_{xo} describes the extent to which the perforation interrupts the longitudinal load path, triggering LPB. In units of mm (see Appendix G, the geometric parameter proposed in this study is:

$$\Omega = \frac{l_x}{(S_{yo})^2} + \frac{\sqrt{l_y}}{S_{xo}} \tag{6.10}$$

Noteworthy, equation 6.10 is not intended as a fundamental non-dimensional group derived from the governing differential equation, but as an empirical geometric index used to distinguish collapse mechanisms. The term l_x/S_{yo}^2 increases rapidly as the transverse ligament becomes narrow, reflecting the disproportionate sensitivity of local strip or column-like behaviour to reductions in ligament width. The term $\sqrt{l_y}/S_{xo}$ increases for taller perforations located closer to the loaded edge, and therefore represents the disturbance of the longitudinal load path. The square root indicates that this influence is sub-linear within the regression domain, consistent with the observation that increasing l_y promotes stress redistribution and localisation, but not in direct proportion to the perforation height (Chapter 4 shows that LPB is a complex failure mode). When the plate capacity ratio is plotted against $\lambda_{l,m}$, a group of 109 regular and 54 staggered GMNIA models appear to distinctly follow an exponential curve, albeit for 4 outliers A to D (see Figure 6.14a). These 163 plates are characterised by $\Omega \leq 0.265$.



(a) GMNIA models with globalised failure ($\Omega \leq 0.265$)

(b) GMNIA models with localised failure ($\Omega > 0.265$)

(c) Examples of modes at failure

Figure 6.14: The behaviour and stress distribution in perforated plates at failure

For the regularly arranged perforations characterized by $\Omega \leq 0.265$ (i.e., 109 models), 71.6% exhibit WPB or an interaction (INT) of buckling modes (both at 35.8%), as shown in Figure 6.14a. The remainder of models (91 regular and 46 staggered) exhibit a much greater variation in their capacity ratio, as displayed in Figure 6.14b. For the regularly arranged perforations characterised by $\Omega > 0.265$, only 29.7% exhibit WPB or an interaction (INT) of buckling modes (constituting 16.5% and 13.2% respectively). Such geometries are more inclined to USB, ESB and LPB buckling modes which constitute 26.4%, 22.0% and 22.0% of the total respectively. A summary of these observations is given in Table 6.9.

Table 6.9: Elastic failure mode of models with a regular arrangement of perforations

Mode	% Of Total	
	$\Omega \leq 0.265$ (109 Models)	$\Omega > 0.265$ (91 Models)
ESB	11.9	22.0
INT	35.8	13.2
LPB	12.8	22.0
USB	3.7	26.4
WPB	35.8	16.5

Examples of the Von Mises stress field at ultimate capacity is shown in Figure 6.14c. For models with $\Omega \leq 0.265$, the outliers A to D possess particularly small S_{yo} with 'WPB-LPB' borderline geometries. In comparison, plates E to I show smaller fraction of the plate at limit stress, concentrated at the strips above and below the perforation. Examples E to I in Figure 6.14c possess a more regular arrangement of perforations with smaller eccentricities about either the x- or y- axes. This is more indicative of a well-predicted ultimate failure capacity even for $\Omega > 0.265$, as these models more closely follow the exponential trend.

6.4.4 New DSM Approach

Using the geometric parameter Ω and the modified slenderness $\lambda_{l,m}$, a new DSM equation is proposed to predict the nominal ultimate capacity $\widehat{N}_{u,h}$ of perforated plates. The resulting curve is governed by three fitting constants denoted κ, α, and ρ, and is a function of the modified slenderness $\lambda_{l,m}$. Traditional DSM equations typically omit α, however its inclusion significantly improves the fit for perforated plates.

$$N_{u,h} = N_{y,net}\left(1 + \kappa\left(\frac{1}{\lambda_{l,m} + \alpha}\right)^{\rho}\right)\left(\frac{1}{\lambda_{l,m} + \alpha}\right)^{\rho} \tag{6.11}$$

Table 6.10: Optimised parameters for each subset

Subset	# Models	κ	α	ρ	R^2
$\Omega \leq 0.265$	163	3.55	1.01	7.97	0.917
$\Omega > 0.265$	137	-0.0198	0.932	8.93	0.632

Regression was conducted to determine these numerical parameters with the 'curve fit' function in the 'scipy.optimize' python module [124]. The optimised constants are shown in Table 6.10 for all 163 models with $\Omega \leq 0.265$ and all 137 models with $\Omega > 0.265$, regardless of the perforations arrangement. The poorer R^2 score for models with $\Omega > 0.265$, as illustrated in Figure 6.14b, results from the greater scatter in the data. To reduce the DSM curve to a form that is controlled by a single parameter, the equation is mathematically reformulated. The parameter α is found to be approximately equal to 1, and the optimal exponent values ρ obtained for the two subsets with $\Omega \leq 0.265$ or $\Omega > 0.265$ were similar. Therefore, a single exponent $\rho = 7.95$, more closely approximating the group with lower scatter (i.e., $\Omega \leq 0.265$), and $\alpha = 1$ were adopted. This simplification necessitated the re-optimisation using regression for the remaining coefficient, κ, as illustrated in Figure 6.15. Note that κ is quite sentsitive to this simplification. This reformulated equation is given by:

$$N_{u,h} = N_{y,net} \cdot \left(1 + \kappa \left(\frac{1}{\lambda_{l,m} + 1}\right)^{7.95}\right) \cdot \left(\frac{1}{\lambda_{l,m} + 1}\right)^{7.95}, \tag{6.12a}$$

$$\kappa = 2.77 \quad \text{if} \quad \Omega \leq 0.265, \tag{6.12b}$$

$$\kappa = 1.44 \quad \text{otherwise} \tag{6.12c}$$

Using the parameters defined in Equation 6.12, the simplified expression results in R^2 values that remain largely unchanged: $R^2 = 0.917$ or 0.624 for plates with $\Omega \leq 0.265$ or $\Omega > 0.265$ respectively. The final section of this chapter will assess the accuracy and reliability of this equation for plates with a regular arrangement of perforations as well as those with a staggered arrangement of perforations.

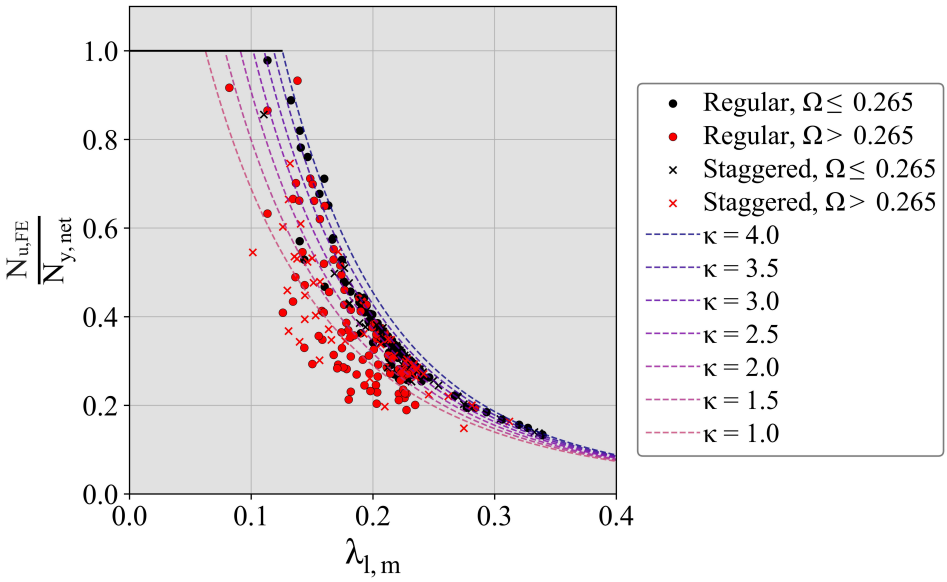


Figure 6.15: A suite of DSM curves with different κ values for all 300 models

6.5 Final Comparison and Reliability Factor

Figure 6.16 displays all 200 models with a regular arrangement of perforations (circles), all 100 models with a staggered arrangement of perforations (crosses), and all 11 experimental results (triangles).

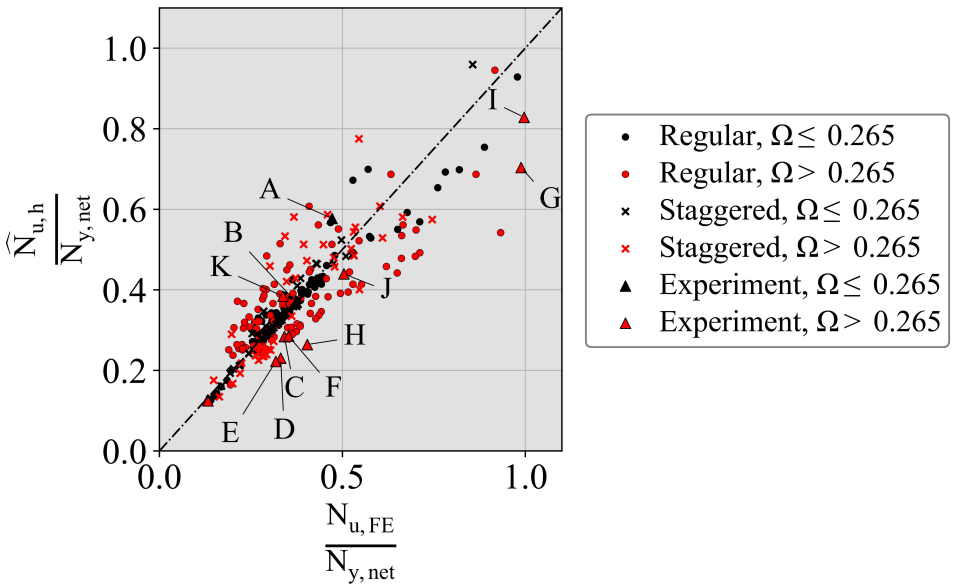


Figure 6.16: Accuracy of the prediction for all GMNIA and experimental results (labelled)

Table 6.11: Comparison of the accuracy of the modified approach by [5] ($N_{u,YR}$) and Naraidoo & Rossi ($\widehat{N}_{u,h}$, Equation 6.12), for plates with a regular (R) and staggered (S) arrangement of perforations

Perforation Pattern	Ω	# Models	$N_{u,YR}$ μ	$N_{u,YR}$ COV	$N_{u,YR}$ R^2	$\widehat{N}_{u,h}$ (Equation 6.12) μ	$\widehat{N}_{u,h}$ (Equation 6.12) COV	$\widehat{N}_{u,h}$ (Equation 6.12) R^2
Regular	≤ 0.265	109	0.835	0.228	0.470	1.00	0.075	0.927
Staggered		54	0.776	0.156	0.162	1.00	0.051	0.959
Both		163	0.815	0.212	0.412	1.00	0.068	0.935
Regular	> 0.265	91	0.794	0.303	0.189	1.02	0.231	0.625
Staggered		46	0.762	0.218	0.009	1.02	0.189	0.619
Both		137	0.783	0.279	0.144	1.02	0.215	0.624
Both	All	300	0.801	0.244	0.280	1.01	0.154	0.777

The mean (μ) and coefficient of variation (COV) obtained with Equation 6.12, ($\widehat{N}_{u,h}$), are compared to those obtained using the modified approach by [5] (i.e., $N_{u,YR}$ in Table 6.11). For both subsets, the ultimate capacity is determined much more accurately with the new equation, also achieving a better R^2 result for all 300 models. This is expected, as the prediction by [5] was developed for thin-walled columns with one row of perforations in the web, which may exhibit an interaction of local, distortional and global buckling mode. In contrast, this approach is developed for plates with multiple rows of perforations, with a regular or staggered arrangement, exhibiting only local elastic buckling. For plates with $\Omega \leq 0.265$, the new equation can determine the ultimate buckling capacity with excellent accuracy. Overall, the new DSM equations also give good predictions across the whole modified slenderness range with lower scatter. In Table 6.12, the nominal ultimate capacity for each test (Equation 6.12) is compared to the measured one ($N_{u,exp}$). Most tests are characterised by $\Omega > 0.265$. On average all experimental results possess a greater capacity than that predicted, with $\mu = 1.21$ and COV= 0.194. This may be explained by the greater rotational stiffness provided by the experimental boundary conditions which are not ideal simply supported conditions as modelled. It was seen that $\widehat{N}_{u,h}$ exhibits greater variation with $\lambda_{l,m}$ for plates with $\Omega > 0.265$, thus more experimental results are required to conclude whether Ω depends on further geometric variables, including the boundary and loading conditions. With $\mu = 1.40$ and COV= 0.514, and $\mu = 0.970$ and COV= 0.294 for $N_{u,SM}$ and $N_{u,YR}$ respectively, both are less accurate than $\widehat{N}_{n,h}$ for all 11 experimental results

(with $N_{u,YR}$ unconservative on average). Furthermore, $\widehat{N}_{u,h}$ demonstrates greater reliability for all experimental results with an R^2 score of 0.731, compared to 0.560 for $N_{u,YR}$.

Table 6.12: The accuracy of Equation 6.12 in comparison to experimental tests for plates with a regular or staggered arrangement of perforations, conducted in [23]

Test	$\lambda_{l,0}$	$\sqrt{t/b}$	$\lambda_{l,m}$	Ω	$\frac{N_{u,exp}}{N_{u,SM}}$	$\frac{N_{u,exp}}{N_{u,YR}}$	$\frac{N_{u,exp}}{\widehat{N}_{n,h}}$
A	1.83	0.0878	0.159	≤ 0.265	0.982	0.993	0.818
B	2.12	0.0879	0.205	≤ 0.265	0.881	0.876	1.01
C	2.06	0.0892	0.211	> 0.265	1.10	0.783	1.19
D	2.18	0.0867	0.238	> 0.265	1.10	0.792	1.28
E	2.14	0.0877	0.242	> 0.265	1.05	0.749	1.48
F$^{\Diamond}$	1.97	0.0882	0.211	> 0.265	1.25	0.782	1.37
G$^{\Diamond}$	1.18	0.0868	0.110	> 0.265	2.69	1.53	1.33
H$^{\Diamond}$	1.81	0.0883	0.220	> 0.265	1.35	0.843	1.52
I$^{\Diamond}$	1.15	0.0876	0.093	> 0.265	2.93	1.51	1.19
J$^{\Diamond}$	1.69	0.0873	0.160	> 0.265	1.30	1.00	1.13
K$^{\Diamond}$	2.19	0.0874	0.201	≤ 0.265	0.776	0.809	0.873
μ					1.40	0.970	1.20
COV					0.514	0.294	0.194

$\Diamond$ The perforations are staggered. n_l and n_t are the number of perforations over the 'critical length' and 'critical width', taken as the narrowest length in the longitudinal and transverse direction respectively, see Figure 6.2

The load and resistance factor method (LRFD) outlined in [7] was used to determine the resistance factor Φ to be used in conjunction with the proposed DSM equation, where:

$$\Phi = C_\Phi M_m F_m P_m \cdot \exp\left(-\beta_0 \sqrt{V_M^2 + V_F^2 + C_p V_p^2 + V_Q^2}\right) \tag{6.13}$$

In [7, 22] a value of 0.850 for non-perforated plate buckling is prescribed. This value accounts for the uncertainties and variabilities inherent in the nominal resistance R_n, such that the design resistance ($R_d = \Phi R_n$) is found with 95% confidence that the 'true' capacity exceeds R_d. The values for each coefficient is summarised in Table 6.13; the mean ratio of the experimentally determined to nominal capacity (P_m) and the coefficient of variation of these values (V_P) are shown for all 300 models. The coefficient C_P was determined for all 300 tests. The lower bound on the ratio P_m is computed as:

$$R_L = P_m - t_{0.05,n-1}\left(\frac{P_m V_P}{\sqrt{n}}\right) \tag{6.14}$$

Equation 6.13 applied on the set of 300 FE models yields $\Phi = 0.830$, and for the set of 163 models with $\Omega \leq 0.265$ it yields $\Omega = 0.904$. This means that only 83.0% of R_n is guaranteed at 95% confidence for the 300 specimens studied. However, when the 95% lower bound on the ratio is computed using Equation 6.14, it is found that R_L is equal to 0.996, therefore it is recommended to adopt the standard $\Phi = 0.850$ in conjunction with Equation 6.12 for the proposed method to be at least as reliable as the baseline AISI/AS-NZS calibration [7, 22].

Table 6.13: Coefficients used to determine the LRFD resistance factor Φ

Coefficient	AISI [7]	Value
n	K2.1.1	300
C_Φ	K.2.1.1	1.52
M_m	K2.1.1	1.10
F_m	I6.3.1, K2.1.1	1.00
P_m	K2.1.1	1.01
β_0	K2.1.1	2.50
V_M	I5.3.1, K2.1.1	0.10
V_F	I6.3.1, K2.1.1	0.05
C_P	B4.2, K2.1.1	1.01
V_P	B4.2, I6.3.1	0.16
V_Q	I6.3.1, K2.1.1	0.21

6.6 Conclusion

The Direct Strength Method (DSM) is an approach used to determine a nominal ultimate capacity of cold-formed steel members, based on their elastic buckling behaviour. Compared to the effective width method, it is particularly useful for designing complex cross-sections, including those with stiffeners or lips. For cross-sections with perforations, [114] identify that application of the current DSM approach (with $N_{y,net}$ in replace of N_y) is conservative for global and distortional failure, however it should not be used when local buckling governs failure. Furthermore, DSM approaches proposed by [113] using [3], and [5] are either more time-consuming or too conservative. Additionally, neither method can accurately find the local buckling capacity for experimental tests conducted on simply supported plates with either a regular or staggered arrangement of perforations [25]. In this chapter, a total of 300 simply supported steel plates with a regular or staggered arrangement of perforations are modelled in ABAQUS. A geometric parameter Ω is determined, and a new modified slenderness $\lambda_{l,m}$ is proposed. For a subset of 163 regular and staggered perforations characterised by $\Omega \leq 0.265$, the ratio $N_{u,h}/N_{y,net}$ exhibits a strong relationship with $\lambda_{l,m}$. Least squares regression was performed on this subset of models to determine a new DSM equation for simply supported perforated plates. This expression is controlled by κ, a variable used to distinguish between geometries characterized by Ω. The proposed expression is given as:

$$N_{u,h} = \begin{cases} N_{y,net} & \text{if } \lambda_{l,m} \leq \Lambda_{l,m}, \\ N_{y,net} \cdot \left(1 + \kappa \left(\frac{1}{\lambda_{l,m}+1}\right)^{7.95}\right) \cdot \left(\frac{1}{\lambda_{l,m}+1}\right)^{7.95} & \text{otherwise,} \end{cases} \tag{6.15a}$$

$$\kappa = \begin{cases} 2.77 & \text{if } \Omega \leq 0.265, \\ 1.44 & \text{otherwise,} \end{cases} \tag{6.15b}$$

$$\Lambda_{l,m} = \begin{cases} 0.1067 & \text{if } \Omega \leq 0.265, \\ 0.0767 & \text{otherwise} \end{cases} \tag{6.15c}$$

Equation 6.15 finds the local elastic buckling load of plates for $\Omega \leq 0.265$ with an R^2 performance of 0.935. However, it is much less reliable for plates with $\Omega > 0.265$ which have an R^2 performance of 0.624. In addition, experimental tests suggest that Ω should account for further geometric parameters, boundary conditions, and loading

distribution to better distinguish how a plate will fail. Yet overall, this research has demonstrated a significant improvement to existing predictions for the ultimate buckling capacity of simply supported perforated plates. The expression proposed in Equation 6.15 can find a nominal ultimate capacity $\widehat{N}_{u,h}$ of all 300 GMNIA models with a mean $\mu = 1.01$, COV= 0.154, and a good overall R^2 performance of 0.777. These results are substantially more accurate than the best-performing prediction for the nominal ultimate capacity of a buckling plate with a row of concentric perforations, proposed by [5] with a mean $\mu = 0.801$, COV= 0.244 and an R^2 performance of 0.280. Finally, the resistance factor proposed by [7, 22] can still be used in conjunction with the method proposed Equation 6.15, to determine the design nominal buckling capacity of simply supported plates with a regular or staggered arrangement of perforations.

7

The Prediction and Classification of Local Buckling for Plates with Tall Perforations

7.1 Commentary

In Chapter 4, two standard-ready empirical formulations for the elastic buckling load of perforated plates were proposed [23]. These were found for plates with a regular arrangement of perforations in n_c columns and n_r rows. The expressions performed with an overall R^2 coefficient of 0.840 for 775 models. However, the accuracy of this prediction reduces significantly when used to predict $N_{cr,h}$ for models with one row of tall perforations, exhibiting LPB or USB.

LPB is prevalent in plates and sections with tall perforations, such as castellated or cellular members. Section 2.6.1 gave an introduction to the geometric limits imposed by [6], however it will be discussed in Section 7.2 that the design equations proposed by [6] and [119] are inapplicable to cases of pure axial compression. Additionally, for a simply supported plate with a tall hole under uniform compression, [53] is the only method summarised in Table 2.1 that can reliably determine $N_{cr,h}$, but only for a single concentric perforation.

By first categorising the types of buckling seen in plates with large perforations with a feed-forward neural network, this paper seeks to find a definitive prediction for the elastic buckling load $\widehat{N}_{cr,h}$. It is found that that the spacings S_{xo} and s_x play a significant role in optimising the elastic buckling load of a member with large openings, which may exceed

that of the equivalent unperforated member. Additionally, Chapter 5 [25] found that the only test to exhibit LPB failed at a capacity closest to the elastic buckling load, suggesting that plates with tall holes fail elasto-plastically. Therefore, this chapter investigates how the variable Ω introduced in Chapter 6 [24] affects the critical buckling capacity $N_{cr,h}$, and can be used to successfully distinguish whether plasticity will be globalised or localised in the plate at ultimate failure. This chapter concludes the investigation into problem (i) outlined in Section 1.2.

7.2 Introduction: Positioning the Problem

Structural members with perforations can be categorised into two types; (i) cellular and castellated beams with large, widely spaced openings; and (ii) rack sections with multiple small perforations over the member width. Plates with 'tall' perforations are the subject of this paper, which provide cellular and castellated beams with a greater second moment of area per unit weight than a solid section, while permitting service integration and maintenance access. However, as large openings greatly reduce the web stiffness and effective depth, such members can exhibit one of seven types of failure (Figure 7.1). These are governed by the applied loading conditions (shear, bending or compression), the opening shape (circular, rectangular or elongated circular), and key geometric variables (including opening position, spacing, size and web slenderness) [119] (Section 2.6.1).

This study concentrates on simply supported plates with tall holes subjected to compression, for which buckling between perforations (web-post buckling with capacity $N_{wp,Ed}$ [6]) and buckling between the perforation and loaded edge (web buckling with capacity $N_{w,Ed}$ [6]) are the predominant modes of failure, such that the elastic buckling load $N_{cr,h} = \min(N_{wp,Ed}, N_{w,Ed})$. The webs of such sections can be conservatively approximated as simply-supported plates, when loaded in axial compression (Section 2.4). A 'tall' hole is defined as one with $l_y \geq 0.6b$ in this paper (geometric variables are defined in Figure 7.2); plates with such holes buckle with 'lateral perforation buckling' (LPB) [26]. There is a significant research gap of methods that can accurately find $N_{cr,h}$; none of the methods proposed by [2–4, 23, 26, 53] can appropriately model this distinct buckling deformation.

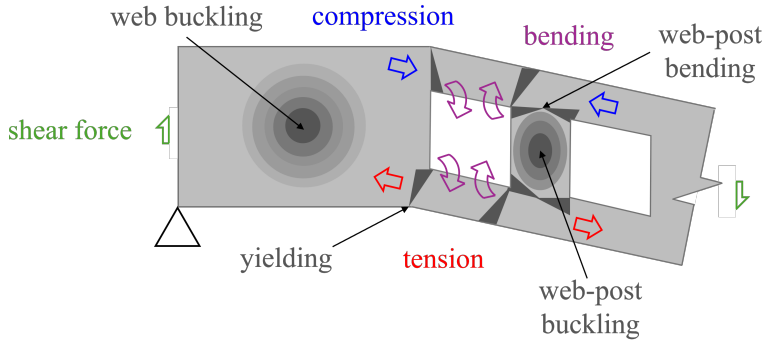


Figure 7.1: Modes of failure in webs with tall perforations, adapted from [119]

fre

EN 1993-1-13 [6] codifies buckling design resistance (N_{Rd}); however this calculation makes two oversimplified assumptions. Firstly, the effective width of the buckling web is arbitrarily approximated as half the hole height $b_w = l_y/2$ (Figure 7.2a [6, 119]).

Secondly, the design resistance of the web-post ($N_{wp,Rd}$) is an empirical function of the relative slenderness of the web-post (λ_{wp}). This formulation assumes that the web and web-post behave as simply supported columns, with the effective lengths shown in Figure 7.2. This assumption is a considerable oversimplification of the true plate buckling behaviour, based on the "simpl" compression field model developed from an approach presented in[119, 159]. As expected, the elastic buckling load predicted with this model is conservative relative to finite element analysis, and was only calibrated for $0.3l_y \leq s_x \leq l_y$ [119, 159].

Most importantly, $N_{wp,Ed}$ and $N_{w,Ed}$ are only intended for sections buckling under flexure; there is no provision for uniform axial compression [6], which will be the dominant force in columns containing large web openings.

In response to all of these issues, this paper seeks to find an accurate approximation for the elastic buckling load of members with tall web openings subject to uniform compression. Additionally, this study seeks to find whether the arrangement of perforations can be optimised to maximise the buckling capacity, and if such arrangements obey the geometrical constraints imposed by [6].

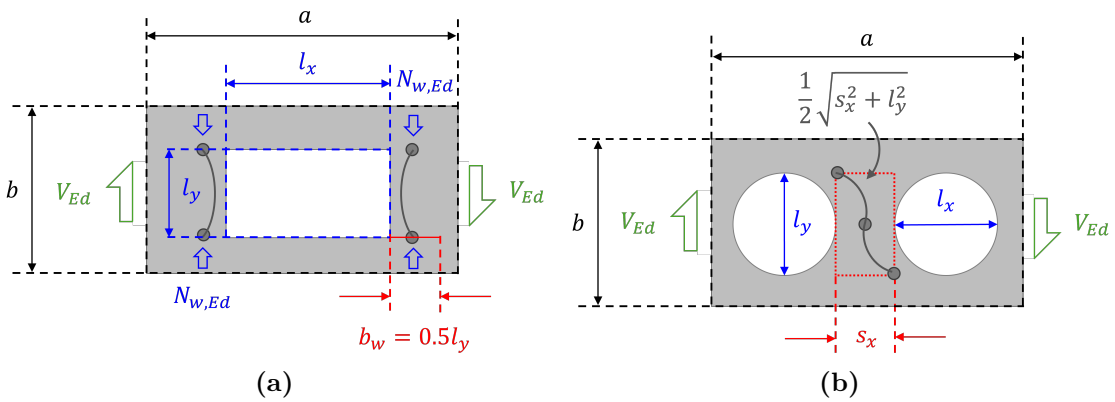


Figure 7.2: Model for buckling of web between (a) widely spaced rectangular openings and (b) closely spaced circular openings, adapted from [6]

7.3 Finite Element Model

To support the development of mode-specific expressions for simply supported plates with a series of tall perforations exhibiting LPB, we will undertake a parametric study using a finite element (FE) model developed next.

Using finite element (FE) analysis in ABAQUS [150], the elastic buckling load of simply supported plates with a series of tall perforations will be found with a linear perturbation assessment following [23], and the ultimate buckling capacity will be found with a geometric and material nonlinear analysis (GMNIA) according to [24]. In accordance with [152], the elasto-plastic stress-strain behaviour is modelled using the two-stage Ramberg-Osgood model, commonly adopted for modelling cold-formed steel. The adopted material properties are summarised in Table 7.1; the Young's modulus (E) is taken in accordance with [23]; the 0.2% yield stress (f_y) and ultimate strength (f_u) agree with the values considered by [116]; the exponents m and n are proposed by [155] and [152] respectively, with the ultimate strain ϵ_u also in accordance with the latter author.

Table 7.1: Material parameters considered in FE models, in accordance with [24]

E (MPa)	f_y (MPa)	f_u (MPa)	$E_{0.2}$ (MPa)	n	m	ϵ_u
203400	450	500	24700	8	3.97	0.060

Given that this study investigates plates with large slenderness (b/t), the use of thin shell quadratic elements is justified. In this context, the only modelling advantage of using S9R5 over S4 finite elements is that the former can model the bending of a thin curved shell more accurately [140] which is not of relevance for the present study. Accordingly, provided that there is a sufficiently large number of nodes, meshing with S4 elements can achieve the same level of accuracy as meshing with S9R5 elements [77, 140]. Consistent with this, a convergence study undertaken by [25] found that at least 15000 S4 elements are required for sufficient accuracy. Using these material and finite element parameters, the ABAQUS input files were meshed with python (Appendix B). The maximum ratio of the element-to-plate width was set to 0.005, and the limit on the minimum number of elements was imposed. In addition, the python meshing script restricted the aspect

ratio of the S4 elements to be square in accordance with [77] who demonstrated that this further improves convergence accuracy. The boundary conditions were applied as nodal constraints, and loading was applied as a continuum distributing coupling constraint between the reference point and the loaded edge as shown in Figure 7.3.

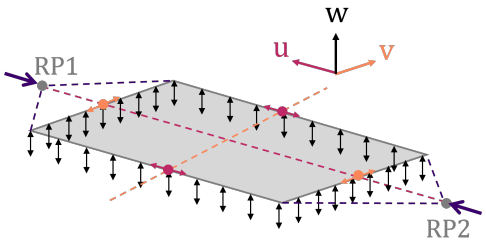


Figure 7.3: Nodal constraints and loading in FE model. Nodes are restrained in the coloured direction shown. Loads are applied at reference points and constrained to nodes along the plate width

For the GMNIA, the Newton-Raphson solution scheme is adopted, following [140] and [116], since ABAQUS/Standard does not provide a stabilisation algorithm for the Riks Arc Length-Method, which does not represent the true buckling behaviour when localised instabilities are present [140]. The models in this study use an adaptive damping factor ('allsdtol') of 0.005, which was increased by 0.005 in the event of non-convergence (Appendix F). The remaining control parameters are given in Table 7.2. Finally, the geometric imperfection is taken as the first compressive eigenvalue obtained in a linear perturbation assessment, with an amplitude equal to $\min(a/200, b/200)$, as per [18].

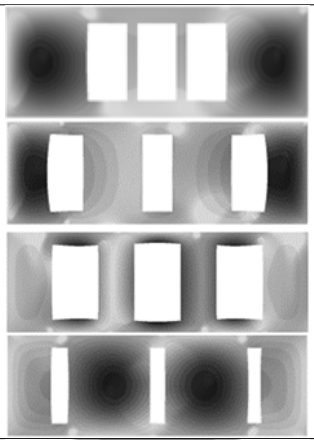
Table 7.2: Control parameters considered in FE models

Initial Step Size	Min Step Size	Max Step Size	Total Period	Max Disp. (mm)
0.001	0.000001	0.015	1	1

7.4 Prediction for Elastic Buckling Load

To enable reliable design, this study first seeks to develop predictive formulae for the elastic buckling load of plates perforated with tall openings. To this end, the observed buckling response is subdivided into three distinct behaviours (Table 7.3). Observe that buckling designations 1(a), 1(b) are akin to the lateral perforation buckling (LPB) mode defined in Chapter 4[23], whereas buckling designation 1(c) is akin to the unstiffened strip buckling (USB) mode. In Chapter 4, it was observed that USB could occasionally benefit the elastic buckling capacity of the plate. This Chapter establishes that to be true, provided the hole is tall. For this reason, it is grouped under 'designation 1' defined as out-of-plane web buckling between the plate and hole (plate-to-hole) edges. The particular type of LPB where out-of-plane buckling occurs between two adjacent holes (hole-to-hole) is termed 'designation 2', and finally in-plane buckling is termed 'designation 3'. A separate prediction for the elastic buckling load ($\widehat{N}_{cr,h}$) will be found for each designation, presented in terms of a plate buckling coefficient $k_{f,i}$ where i denotes the designation (i.e., 1, 2, 3), such that such that $\widehat{N}_{cr,h} = k_{f,1}N_0$.

Table 7.3: Description of observed buckling deformation

Observation	Direction of Buckle	Location of Buckle	Designation
	out-of-plane	plate-to-hole edge (full halfwave)	1(a)
		plate-to-hole edge (partial halfwave)	1(b)
		plate-to-hole edge (hole length halfwave)	1(c)
		hole-to-hole edge	2
	in-plane	N/A	3

7.4.1 Designation 1: Out-of-Plane Plate-to-Hole elastic buckling load

Firstly, buckling deformation with Designation 1 is examined, characterised by a full buckling halfwave between the plate and hole edge (i.e., 'plate-to-hole edge', Table 7.3). To quantify this, a semi-analytical expression for the out-of-plane plate-to-hole edge elastic buckling load is sought in terms of a single plate buckling coefficient $k_{f,1}$ (such that $\widehat{N}_{cr,h} = k_{f,1}N_0$). Accordingly, a parametric study was conducted. A total of 200 models with one row of n_c holes and l_y/b of 0.6, 0.7, 0.8 and 0.9 were examined; the variable domain is summarised in Table 7.4.

Table 7.4: Domain of variables examined in Section 7.4.1 (perforation variables summarised in Figure 1.4)

	a/b	b/t	n_c	l_x/l_y	l_x/a	l_y/b	S_{xo}/b	s_x/b	e_x	e_y
Mean	2.80	80.4	2.71	0.320	0.085	0.749	0.00862	0.442	0	0
Minimum	1.25	80	2	0.0458	0.015	0.6	0.00157	0.0638	0	0
Maximum	5	100	4	0.660	0.177	0.9	0.0196	0.82	0	0

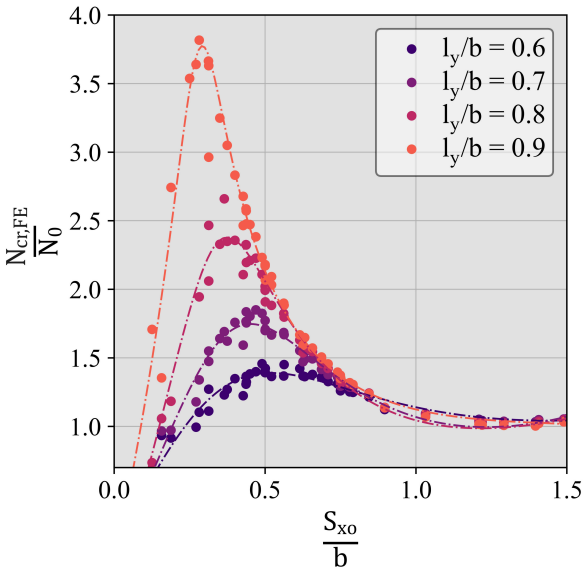


Figure 7.4: Relationship between $N_{cr,FE}/N_0$ and S_{xo}/b with the best-fit curves shown for four values of l_y/b

7. The Prediction and Classification of Local Buckling for Plates with Tall Perforations

To evaluate the influence of the outer plate-to-hole edge distance ($S_{xo,i}$), the elastic buckling to unperforated plate buckling $N_{cr,FE}/N_0$ ratio is plotted with respect to $S_{xo,i}/b$ in Figure 7.4. Note that in this case, $S_{xo,1} = S_{xo,2} = S_{xo}$ since holes are distributed symmetrically about the plate centre ($e_x = e_y = 0$). The relationship between $N_{cr,FE}/N_0$ and $S_{xo,i}/b$ is characterised by three distinct features; (i) an initial, almost linear, increase in the elastic buckling load ratio as $S_{xo,i}/b$ increases, (ii) a pronounced local maximum, and (iii) an asymptotic decay $N_{cr,FE}/N_0 \to 1$ as $S_{xo,i}/b \to \infty$ (i.e., behaviour approaching that of an unperforated plate). A purely polynomial representation inadequately represents these three features, as it cannot capture both (ii) and (iii); and a purely logarithmic expression cannot accurately capture (i). Therefore a composite functional form was adopted, consisting of a quadratic polynomial (to capture features (i) and (ii)), coupled with a logarithmic decay term to enforce the asymptotic trend (iii). This relationship is formulated as:

$$k_{f,1} = \left(A \cdot \left(\frac{S_{xo}}{b}\right)^2 + B \cdot \left(\frac{S_{xo}}{b}\right) + C\right) - \ln\left(D \cdot \left(\frac{S_{xo}}{b}\right)^2 + E \cdot \left(\frac{S_{xo}}{b}\right) + F\right) \tag{7.1}$$

Where the coefficients A to F are found with regression for each l_y/b using SciPy [160], and $\widehat{N}_{cr,h}$ is the predicted elastic buckling load. This gives the following expressions for $k_{f,1}$, for $l_y/b = 0.6$ and $l_y/b = 0.9$ (for simplicity, expressions for $l_y/b = 0.6$ and $l_y/b = 0.7$ are not shown, although they were also found):

$$\begin{aligned} k_{f,1}\left(\frac{l_y}{b} = 0.6\right) = &\left(-0.13\left(\frac{S_{xo}}{b}\right)^2 + 1.94\left(\frac{S_{xo}}{b}\right) + 0.78\right) \\ &- \ln\left(6.90\left(\frac{S_{xo}}{b}\right)^2 - 4.57\left(\frac{S_{xo}}{b}\right) + 1.95\right), \end{aligned} \tag{7.2a}$$

$$\begin{aligned} k_{f,1}\left(\frac{l_y}{b} = 0.9\right) = &\left(-0.68\left(\frac{S_{xo}}{b}\right)^2 + 3.60\left(\frac{S_{xo}}{b}\right) + 0.43\right) \\ &- \ln\left(17.6\left(\frac{S_{xo}}{b}\right)^2 - 10\left(\frac{S_{xo}}{b}\right) + 1.51\right) \end{aligned} \tag{7.2b}$$

Let the equation found with regression for each of the four hole height ratios investigated ($l_y/b = 0.6, 0.7, 0.8, 0.9$) be $f_1(l_y, S_{xo}, b)$. It is desired to find a set of curves $f_2(l_y, S_{xo}, b)$ that can approximate $f_1(l_y, S_{xo}, b)$, while allowing for interpolation between them (i.e., approximate the relationship between S_{xo}/b and $k_{f,1}$ for hole height ratios where $f_1(l_y, S_{xo}, b)$ is unknown). The series of curves $f_2(l_y, S_{xo}, b)$ is given in terms of three scale factors ψ_i,

determined such that when $l_y/b = 0.6$, $k_{f,1}$ is equal to Equation 7.2a, and when $l_y/b = 0.9$, $k_{f,1}$ is equal to Equation 7.2b. The scale factors must also approximate the location of the first turning point (when $k_{f,1}$ is a maximum) for $0.6 \leq l_y/b \leq 0.9$. Greater detail on the mathematical derivation of these scale factor equations is presented Appendix G.

$$\begin{aligned} k_{f,1} = \psi_1 \Bigg\{ \Bigg(&- 0.13 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right)^2 + 1.94 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right) + 0.78 \Bigg) \\ &- \ln\left(6.90 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right)^2 - 4.57 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right) + 1.95\right) \Bigg\} \\ + \psi_2 \Bigg\{ \Bigg(&- 0.68 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right)^2 + 3.60 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right) + 0.43 \Bigg) \\ &- \ln\left(17.6 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right)^2 - 10 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b}\right) + 1.51\right) \Bigg\}, \end{aligned} \tag{7.3a}$$

$$\psi_1 = -91.667 \cdot \left(\frac{l_y}{b}\right)^3 + 207.5 \left(\frac{l_y}{b}\right)^2 - 157.83 \left(\frac{l_y}{b}\right) + 40.8, \tag{7.3b}$$

$$\psi_2 = 66.667 \cdot \left(\frac{l_y}{b}\right)^3 - 145 \left(\frac{l_y}{b}\right)^2 + 106.83 \left(\frac{l_y}{b}\right) - 26.3, \tag{7.3c}$$

$$\psi_3 = -50 \cdot \left(\frac{l_y}{b}\right)^3 + 125 \left(\frac{l_y}{b}\right)^2 - 102 \left(\frac{l_y}{b}\right) + 28 \tag{7.3d}$$

The second turning point (when $k_{f,1}$ is a minimum) is defined by $S_{xo,i} = S_{xo,T}$, after which point the values predicted using Equation 7.3 no longer describe the true physical behaviour. To ensure that $\widehat{N}_{cr,h} \to N_0$ as $S_{xo,i}/b \to \infty$, $k_{f,1}$ is limited by $k_{f,1,T}$, the elastic buckling load at $S_{xo,T}$.

$$k_{f,1} = \begin{cases} \min(k_{f,1,1},\, k_{f,1,2}) & \text{if } S_{xo,i} < S_{xo,T} \\ k_{f,1,T} & \text{if } S_{xo,i} \geq S_{xo,T} \end{cases} \tag{7.4}$$

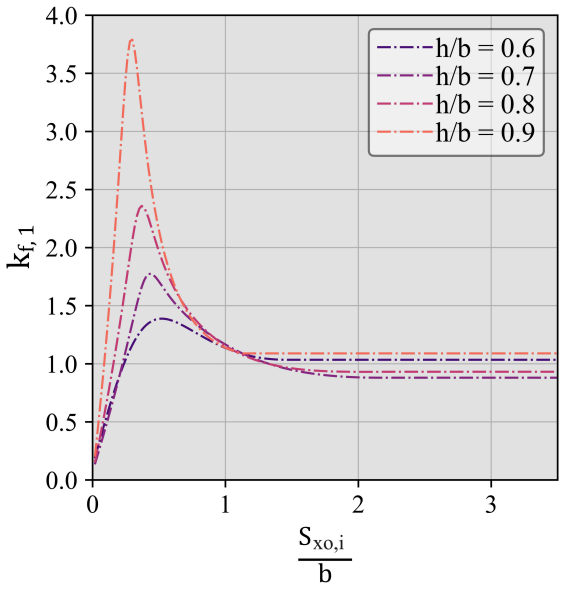


Figure 7.5: The elastic buckling coefficient $k_{f,1}$ corrected for the limiting ratio $S_{xo,T}$

The unperforated plate capacity $N_0 \propto k_0$ where k_0 is the unperforated plate buckling coefficient, and $k_0 \approx 4$ for simply supported plates. Therefore the perforated plate capacity can be expressed as $\widehat{N}_{cr,h} \approx 4k_{f,1}N_0$, and the set of curves shown in Figure 7.6 can be compared to the results of [35] in Figure 7.7. In [35], the authors examined plates with a row of circular perforations with diameter l_{xy} and centre-to-centre spacing s_x. They present a perforated plate buckling coefficient k_h as a function of $\frac{s_x}{l_{xy}}$, considering $s_x = 2S_{xo}$.

Comparing the two graphs demonstrates that the observations made here are not unique and it may be concluded that the expressions given by Equation 7.3 can accommodate a continuous height ratio in the range $0.6 \leq l_y/b \leq 0.9$. The set of equations given by Equation 7.3 possess an R^2 score of 0.971 with a mean accuracy ($\frac{N_{cr,FE}}{\widehat{N}_h}$) of 1.01, and coefficient of variation (COV) of 0.0632 for the 200 models examined (Table 7.4).

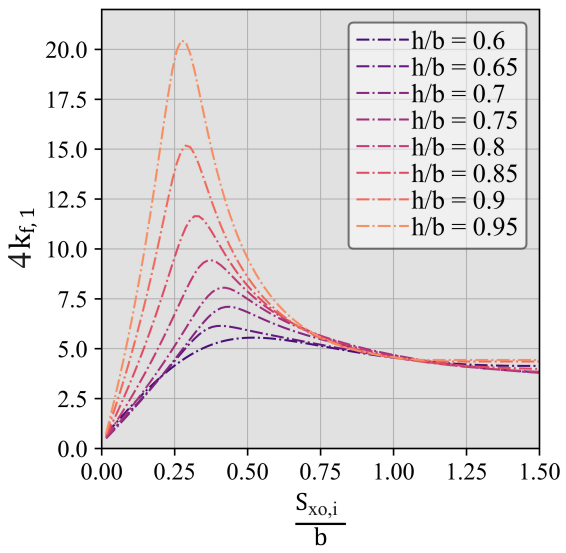


Figure 7.6: Equation 7.3 plotted as the side aspect ratio varies

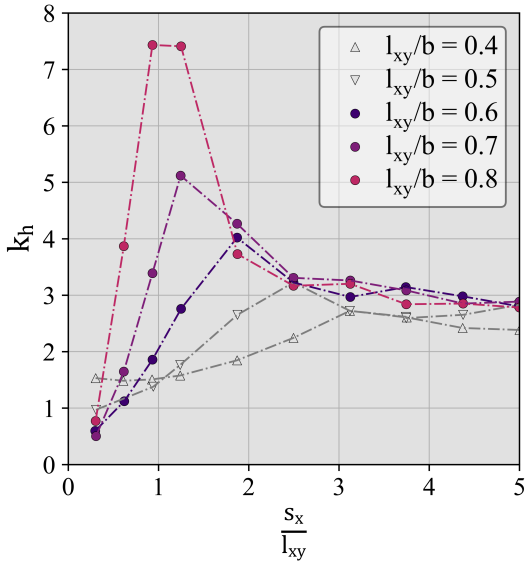


Figure 7.7: Computational results reproduced from [35]

7.4.2 Designation 2: Out-of-Plane Hole-to-Hole elastic buckling load

In contrast to designation 1, designation 2 corresponds to buckling at the hole-to-hole edge (Table 7.3). To find a semi-analytical expression for out-of-plane hole-to-hole edge buckling in terms of a plate buckling coefficient $k_{f,2}$ (such that $\widehat{N}_{cr,h} = k_{f,2}N_0$), a second parametric study was undertaken. The variable domain of the 40 models is summarised in Table 7.5. The average eccentricity of the perforations about the centre in the length (x) and width (y) directions are given as e_x and e_y respectively.

Table 7.5: Domain of variables examined in Section 7.4.2 (perforation variables summarised in Figure 1.4)

	a/b	b/t	n_c	l_x/l_y	l_x/a	l_y/b	S_{xo}/b	s_x/b	e_x	e_y
Mean	3.07	97.1	2.73	0.157	0.0407	0.795	0.361	1.21	0.0172	0
Minimum	2.25	66.7	2	0.0707	0.0163	0.6	0.15	0.785	0	0
Maximum	4	100	4	0.2	0.0567	0.92	0.75	2.36	0.159	0

In this case, the buckling region of length s_x is approximated as a simply supported plate, for which the unperforated plate buckling coefficient k_0 is given by:

$$k_0 = \left(\frac{mb}{a} + \frac{a}{bm}\right)^2, \tag{7.5a}$$

$$\sqrt{m(m+1)} \leq \frac{\sqrt{a(a+b)}}{b} \leq \sqrt{(m+1)(m+2)} \tag{7.5b}$$

Figure 7.8 displays the buckling curves of an unperforated simply supported plate, which closely resemble the trend of the 40 results shown as black points in Figure 7.9. Motivated by this observation, regression analysis was performed using the unperforated plate expressions, yielding the following expression for $k_{f,2}$:

$$k_{f,2} = \left(\frac{m}{\frac{s_x}{b} - 0.45} + \frac{\frac{s_x}{b} - 0.45}{m}\right)^{0.9} - 0.8, \tag{7.6a}$$

$$0.45 + \sqrt{m(m+1)} \leq 0.45 + \frac{\sqrt{s_x(s_x+b)}}{b} \leq 0.45 + \sqrt{(m+1)(m+2)} \tag{7.6b}$$

Where m are the number of buckling halfwaves between the hole edges.

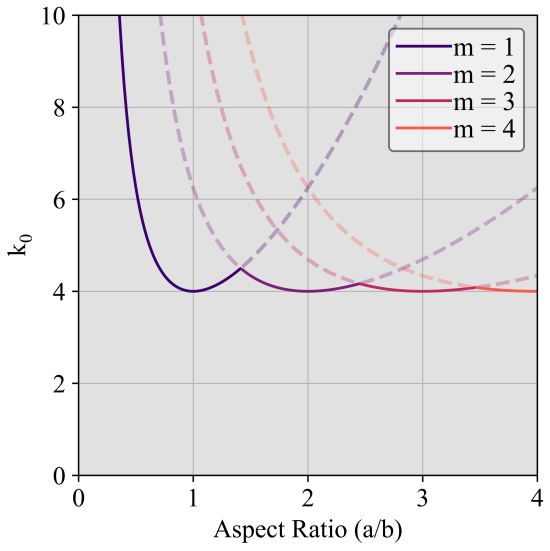


Figure 7.8: Simply supported plate buckling curves for a plate of length a, and width b

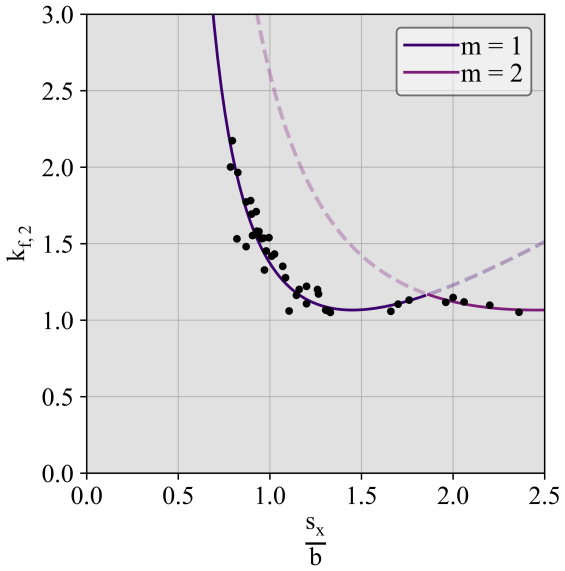


Figure 7.9: Proposed buckling curves for plates exhibiting Designation 2 buckling

The set of equations given by Equation 7.6 possess an R^2 score of 0.867 with a mean accuracy ($\frac{N_{FE}}{\widehat{N}_{cr,h}}$) of 1.01, and coefficient of Variation (COV) of 0.0667 for the 40 models examined (Table 7.5).

7.4.3 Designation 3: In-Plane elastic buckling load

When rectangular perforations are particularly tall and elongated, or when the spacing s_x between two tall perforations is small, the plate may buckle in-plane, thereby suppressing out-of-plane buckling. This behaviour is fundamentally localised Euler buckling, where the end-boundary conditions depend on the width of the buckling strip, and its location within the plate.

To investigate this final designation, a third parametric investigation is undertaken on 40 FE models, with the corresponding domain summarised in Table 7.6, where $S_{xo}/b = \min(S_{xo,1}, S_{xo,2})$.

Table 7.6: Domain of variables examined in Section 7.4.3 (perforation variables summarised in Figure 1.4)

	a/b	b/t	n_c	l_x/l_y	l_x/a	l_y/b	S_{xo}/b	s_x/b	e_x	e_y
Mean	2.47	80	3	0.427	0.163	0.9	0.4	0.198	0	0
Minimum	1.63	80	1	0.236	0.085	0.9	0.28	0	0	0
Maximum	3.75	80	6	1.179	0.386	0.9	1.29	0.563	0	0

Since buckling is localised at the strips of plate between the perforation and the longitudinal supported edges, in-plane buckling is a function of the Euler column-like behaviour of these strips. For the special case of only one perforation, the elastic buckling load $\widehat{N}_{cr,h}$ is given by:

$$\widehat{N}_{cr,h} = \frac{\pi^2 E I_S}{S_{yo} \cdot (k_I l_x)^2} \cdot b \left\{ 1 - \frac{l_y}{b} \right\}, \tag{7.7a}$$

$$I_S = \frac{t \cdot S_{yo}^3}{12} \tag{7.7b}$$

In this configuration, $k_I = 0.523$ provides exact agreement with the FE result. However, the accuracy of the expression given by Equation 7.7 reduces exponentially as s_x and n_c increase. To account for this, least-squares regression was performed on the 40 models given in Table 7.6, introducing an additional factor $k_{f,3}$, such that the final expression

for $\widehat{N}_{cr,h}$ is given by:

$$\widehat{N}_{cr,h} = k_{f,3} \cdot \frac{\pi^2 E I_s}{0.274 \cdot S_{yo} L_{x,o}^2} \cdot b \left\{ 1 - \frac{l_y}{b} \right\}, \tag{7.8a}$$

$$k_{f,3} = \left(\frac{\alpha^2}{\beta (\ln s_x)^\gamma + \alpha^2} \right), \tag{7.8b}$$

$$\alpha = \begin{cases} 2.44 & \text{if } n_c = 2 \\ \frac{1}{0.184(n_c - 2.649)^2} & \text{otherwise,} \end{cases} \tag{7.8c}$$

$$\beta = \begin{cases} 0.0112 & \text{if } n_c = 2 \\ \frac{1}{0.218(n_c - 2.894)^2} & \text{otherwise,} \end{cases} \tag{7.8d}$$

$$\gamma = \begin{cases} 5 & \text{if } n_c = 2 \\ 2.5 & \text{otherwise} \end{cases} \tag{7.8e}$$

The proposed formulation achieves a mean accuracy of $\mu = 0.939$, COV $= 0.149$ and an R^2 score of 0.80 for the 40 models given in Table 7.6.

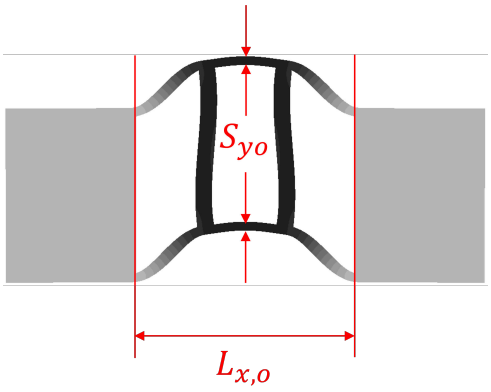


Figure 7.10: Lattice-like Buckling

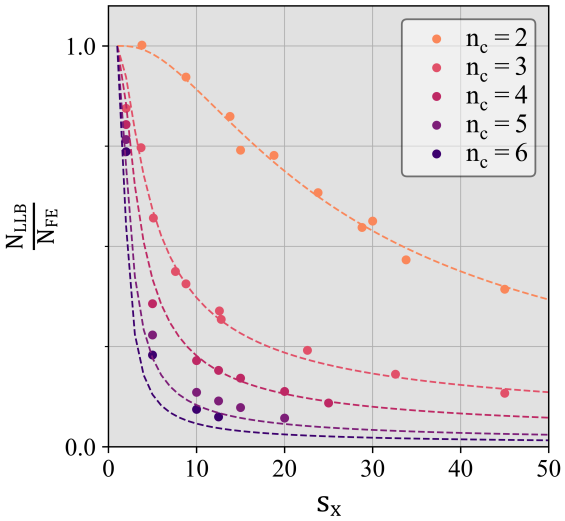


Figure 7.11: The relationship between the capacity predicted with Equation 7.8 and the spacing s_x depending on n_c

7.4.4 Classification of elastic buckling load

In [23], the authors present an algorithm for the classification of local buckling modes. Although this is highly interpretable, easy to implement, and standard-ready, this rule-based approach can be brittle for unseen combinations. Although it is less interpretable, a feed-forward neural network (FNN) is more generalisable, and scales to large datasets with many features. For this reason, a FNN is trained to classify the out-of-plane buckling deformation according to its designation (Table 7.3). As a preliminary step, a random forest classifier was used to determine the feature importance of every geometric parameter, finding that $S_{xo,1}/b$, $S_{xo,2}/b$, s_x/b and l_y/b have the greatest significance. Notably, these are the same parameters that appear in the analytical expressions developed in the previous sections, supporting their physical relevance and credibility.

Based on these findings, the FNN was configured with 3 hidden layers, with 20 neurons per layer, and was trained for 1000 epochs on a random 80% of the parametric domain. Each hidden layer has ReLU activation, and SoftMax activation is used for the output layer. Across 25 independent runs with different random seeds, the mean (μ) classifier accuracy was high (88.6 $\pm$ 5.16 %, Figure 7.12); the results for a typical random test group are given in Figure 7.13.

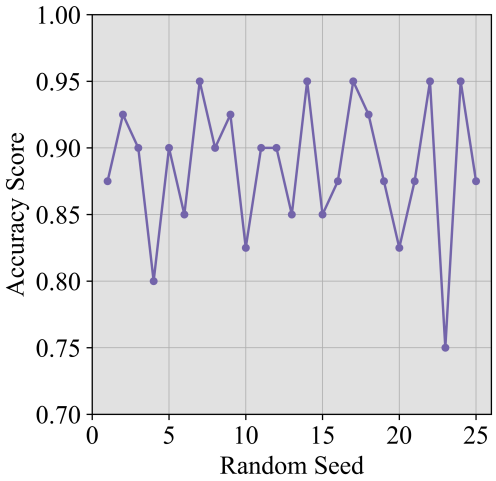


Figure 7.12: Accuracy score for 25 random seeds

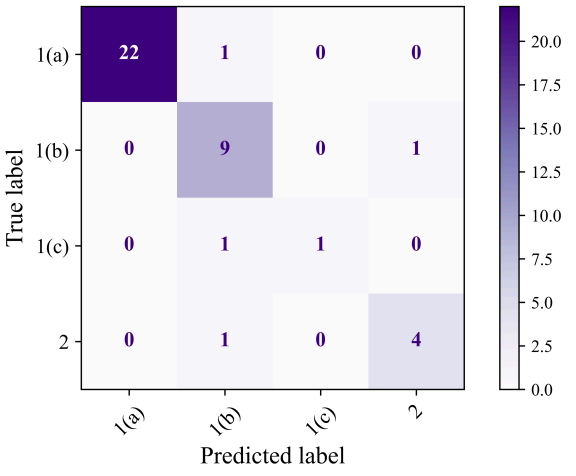


Figure 7.13: Example of results for a random seed of 25

7.5 Overall Performance

Together, the formulae developed in Sections 7.4.1 to 7.4.3 provide a reliable prediction when used with the classifier developed in Section 7.4.4 on unseen data, demonstrated herein on 100 test cases.

The out-of-plane buckling mode is first identified with the FNN trained in Section 7.4.4 (illustrated in Figure 7.14). Then, the corresponding elastic buckling load is determined using the relevant equation (summarised in Table 7.7). Additionally, the elastic buckling load for in-plane deformation (designation 3) is also calculated and compared to the out-of-plane capacity. Of the two calculations, the smaller value is selected, and the geometry is re-classified as designation 3 if this was found to be the weaker mode of failure.

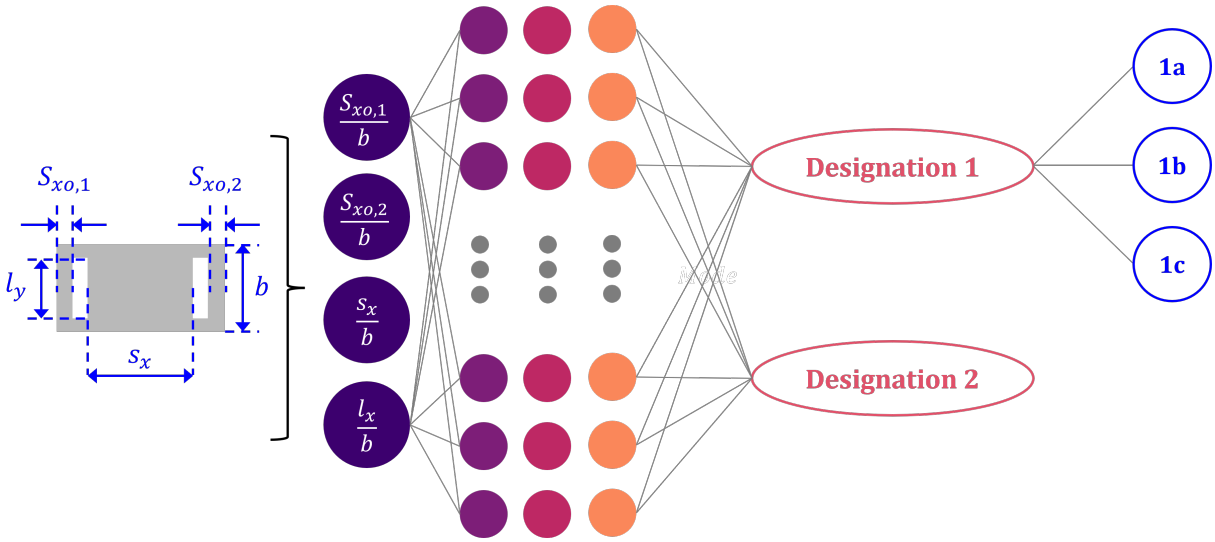


Figure 7.14: Neural network for out-of-plane buckling mode classification

Table 7.7: Summary of equations for each buckling mode

Designation	Capacity (N/mm)	Buckling Coefficient
1	$\widehat{N}_{cr,h} = k_{f,1} N_0$	$k_{f,1} = \begin{cases} \min(k_{f,1,1},\, k_{f,1,2}) & \text{if } S_{xo,i} < S_{xo,T} \\ k_{f,1,T} & \text{if } S_{xo,i} \geq S_{xo,T} \end{cases}$
2	$\widehat{N}_{cr,h} = k_{f,2} N_0$	$k_{f,2} = \left(\frac{m}{\frac{s_x}{b} - 0.45} + \frac{\frac{s_x}{b} - 0.45}{m} \right)^{0.9} - 0.8$
3	$\widehat{N}_{cr,h} = k_{f,3} \cdot \frac{\pi^2 E I_s}{0.274 \cdot S_{yo} L_{x,o}^2} \cdot b \left\{ 1 - \frac{l_y}{b} \right\}$	$k_{f,3} = \left(\frac{\alpha^2}{\beta (\ln s_x)^\gamma + \alpha^2} \right)$

To verify the combined accuracy of both the classifier and the suite of proposed equations, a fourth parametric dataset of 100 models was generated, with the corresponding variable domain summarised in Table 7.8.

Table 7.8: Domain of variables examined in section 7.5 (perforation variables summarised in Figure 1.4)

	a/b	b/t	n_c	l_x/l_y	l_x/a	l_y/b	$S_{xo,1}/b$	$S_{xo,2}/b$	s_x/b	e_x	e_y
Mean	3.13	94.6	4.22	0.281	0.0742	0.797	0.346	0.667	0.553	0.0539	0
Minimum	1.7	53.3	2	0.0745	0.0176	0.6	0.055	0.259	0.0933	0	0
Maximum	5.67	105	22	0.933	0.264	0.967	1.23	2.74	2.52	0.278	0

By comparing the variable domain for the validation dataset (Table 7.8) to those examined in Section 7.4.1 (Table 7.4), Section 7.4.2 (Table 7.5) and Section 7.4.3 (Table 7.6), it is seen that both the mean and max(a/b) for the validation dataset exceed those in the training datasets. Similarly, there is a greater range in b/t, n_c, $S_{xo,i}/b$, s_x/b and e_x examined in the validation dataset. Yet despite extrapolating outside of the domain on which it was trained, 89.1% of the dataset is correctly classified (Figure 7.15), and $N_{cr,FE}$-to-$\widehat{N}_{cr,h}$ has an R^2 score of 0.976 (Figure 7.16). The average $N_{cr,FE}/\widehat{N}_{cr,h}$ is 1.06 with a COV of 0.262.

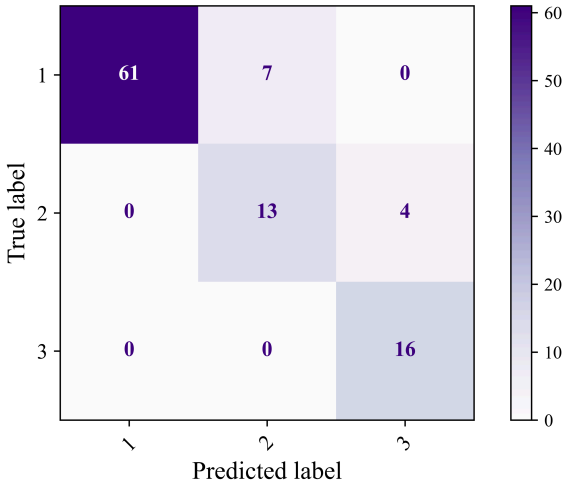


Figure 7.15: Confusion matrix for parametric database in Table 7.8

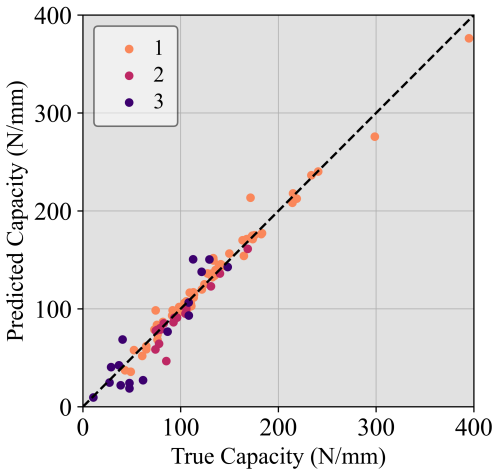


Figure 7.16: Relationship between the predicted capacity and true capacity predicted with equations proposed in Table 7.7, for models examined in Section 7.5

The capacity predicted for designation 3 exhibits the greatest uncertainty, and average accuracy improves to 1.02 with a reduced COV of 0.118 when these 16 cases are excluded. This highlights that the proposed framework is particularly robust for out-of-plane buckling modes, the primary focus of this paper.

Although designation 1 exhibits the greatest spread in elastic capacity $N_{cr,FE}$, Figure 7.17 shows that it also offers the most significant enhancement on average, relative to the unperforated plate elastic buckling load ($N_{cr,FE}/N_0$). In particular, plates buckling according to designation 1 can achieve $N_{cr,h} \approx 3N_0$. The results are plotted with respect to the perforation ratio r_p (i.e., the ratio of the perforated area to the gross area of the plate) in Figure 7.17, to emphasise how irrelevant the perforation area is on $N_{cr,h}$ for plates with tall holes exhibiting LPB.

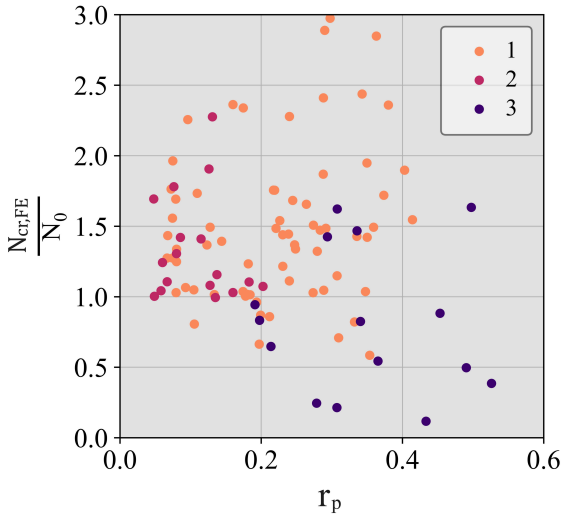


Figure 7.17: Perforated-to-unperforated elastic buckling load ratio as perforation ratio varies

7.6 Optimising Capacity

The previous section demonstrated that the framework proposed in this paper can accurately (i) classify how a plate with tall perforations will deform, and (ii) predict its elastic buckling load $N_{cr,h}$. Next, this framework will be used to optimise the arrangement of tall perforations in a simply supported plate subject to uniform compression.

7.6.1 Elastic Capacity

As shown in Figure 7.6, for a given side aspect ratio $S_{xo,1}/b$ and hole height ratio l_y/b, the elastic buckling coefficient $k_{f,1}$ exhibits a distinct peak. However, this graph does not display the opposite side aspect ratio $S_{xo,2}$, which will become governing when $k_{f,1,2} < k_{f,1,1}$. The combined effect of each side aspect ratio on the buckling coefficient $k_{f,1}$ is shown in Figure 7.18. The maximum attainable buckling coefficient $k_{f,1,max}$ occurs when the perforations are symmetrically arranged, such that $S_{xo,1} = S_{xo,2}$, corresponding to $e_x = 0$.

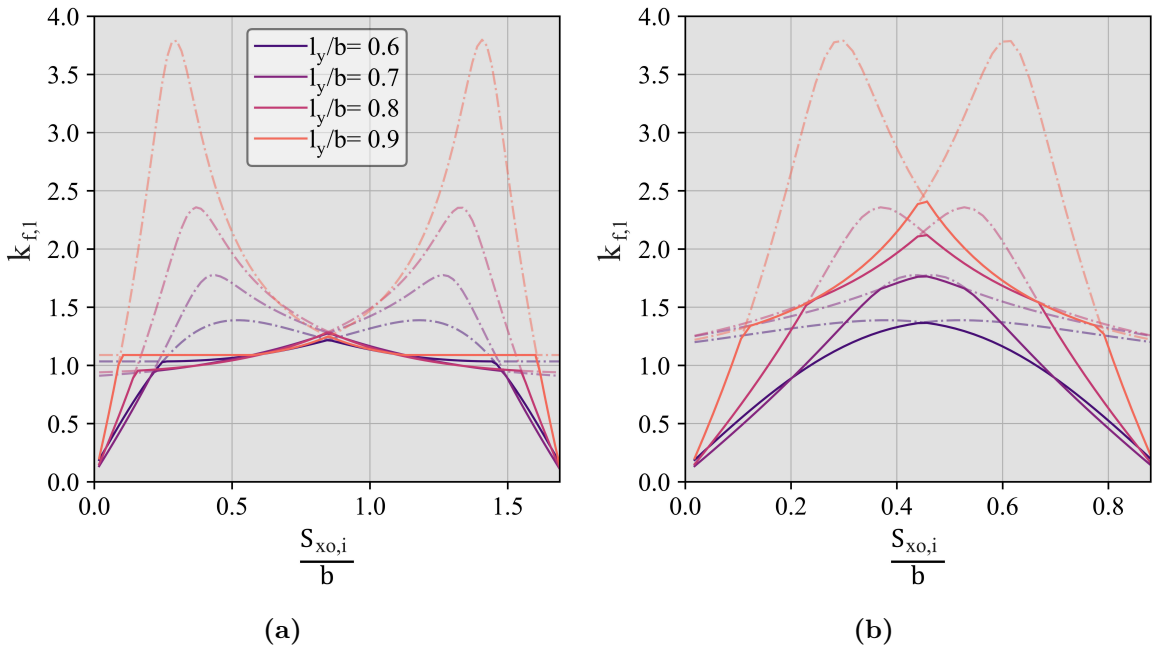


Figure 7.18: $k_f = \min(k_{f,1}, k_{f,2})$ for $a = 200$mm, $b = 100$mm, $l_x = 10$mm, $s_x = 10$mm (a) $n_c = 2$, (b) $n_c = 6$.

Since the spacing S_{xo} is a function of the hole length l_x, number of columns n_c and hole spacing s_x, it follows that the elastic buckling load can be optimised by appropriate selection of s_x, while l_x and n_c were controlled. This optimisation is depicted in Figure 7.19 for a plate with arbitrary dimensions.

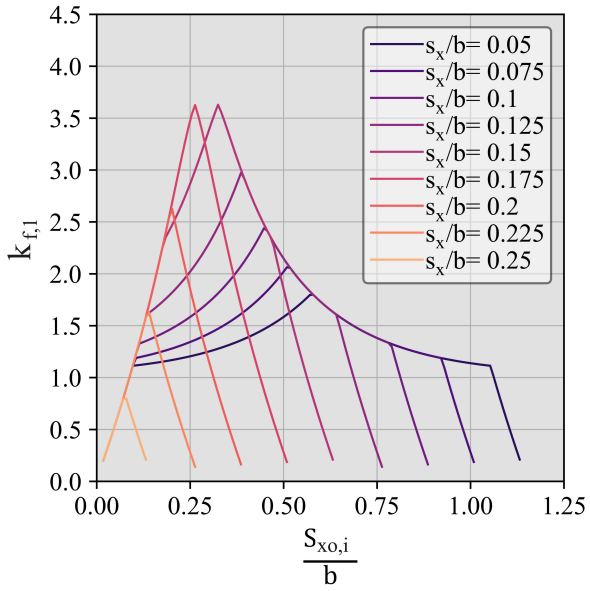


Figure 7.19: $k_{f,1}$ - to - $S_{xo,i}/b$ relationship as s_x is varied

A python script was developed to trial 50 discrete values of s_x while the variables l_x and n_c are controlled. This procedure was repeated for plates with different plate dimensions (a, b and t), and hole geometries (l_y, l_x and n_c); the final database is summarised in Table 7.9. Perforated columns subject to buckling are typically strengthened with longitudinal stiffeners along their length, thus the plate aspect ratio a/b examined in this study represents this smaller buckling length, where a is the length between the stiffeners.

Deformation corresponding to designation 1 was shown to provide the greatest enhancement to elastic buckling load (Figure 7.17), thus this mode is targeted with the classifier developed in Section 7.4.4.

Table 7.9: Domain of variables examined in Section 7.6.1 (perforation variables summarised in Figure 1.4)

	a/b	b/t	n_c	l_x/l_y	l_x/a	l_y/b	$S_{xo,1}/b$	$S_{xo,2}/b$	s_x/b	e_x	e_y
Mean	3.19	96.6	3.94	0.312	0.078	0.798	0.542	0.543	0.667	0	0
Minimum	1.68	75	2	0.111	0.0267	0.605	0.269	0.265	0.096	0	0
Maximum	4.81	110	10	0.643	0.182	0.904	1.07	1.07	1.40	0	0

The mean (μ) elastic buckling load of the 50 'optimised' models in this database is μ=156 N/mm with COV=0.414, an improvement over the randomly generated dataset from Section 7.5, with μ=125 N/mm and COV=0.625 (Table 7.8).

Except for the spacings, the mean of all variables in the 'optimised' and 'random' datasets are very similar (Table 7.9 and Table 7.8 respectively). For reasons discussed, we observe that the mean $E_x \approx 0$ for the optimised models, and that the mean value for S_{xo}/b is similar for both datasets (0.543 and 0.507 for the 'optimised' and 'random' datasets respectively). However, despite a similar minimum and a smaller maximum, the optimised dataset has a greater mean s_x/b. This demonstrates the critical role this variable plays in optimising $N_{cr,h}$ for members with tall perforations, and should therefore be exploited in the design and manufacture of cellular members.

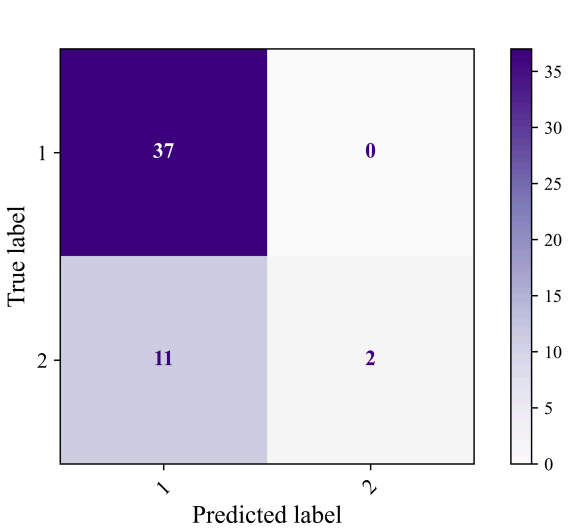


Figure 7.20: Confusion matrix for parametric database in Section 7.6.1 (Table 7.9)

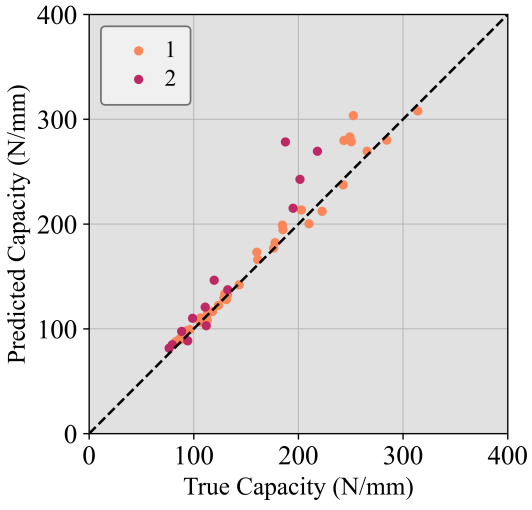


Figure 7.21: Relationship between the predicted capacity and true capacity predicted with equations proposed in Table 7.7 for models examined in Section 7.6.1

Of these models, 22% are misclassified by the neural network developed in Section 7.4.4, and exhibit designation 2 buckling (Figure 7.20). Although the proportion of misclassified models is quite large, the average capacity of these misclassified models actually exceeds that of the models exhibiting designation 1(a) and 1(c) buckling. Such modes must lie at the transition between different designations, and enable a mean $N_{cr,h}/N_0 > 1$, despite misclassification (Table 7.10). Geometries corresponding to designation 1(b) exhibit the greatest enhancement, with an average $N_{cr,FE}/N_0$ of 2.82, therefore representing the most favourable configuration to optimise $N_{cr,h}$ (Table 7.10).

Table 7.10: Observed failure and relative elastic buckling load

Buckling Designation	1(a)	1(b)	1(c)	2
# Models	13	14	10	13
Average $N_{cr,FE}/N_0$	1.34	2.82	1.67	1.81
COV of $N_{cr,h}$	0.45	0.286	0.237	0.422

The relationship between predicted and finite element elastic buckling capacities for the optimised dataset is shown in Figure 7.21. Notably, of the 14 models which exhibit designation 1(b), 12 exhibit multiple tensile modes prior to compressive buckling. In general, geometries with a greater number of tensile modes possess higher elastic buckling capacities than those with fewer such modes. The relationship observed in Figure 7.21 gives an R^2 score of 0.891, with μ=0.952 and COV=0.0777. As the mean $N_{cr,FE}/\widehat{N}_{cr,h} < 1$, a safety factor should be applied if this approach were used to optimise $N_{cr,h}$ in practice.

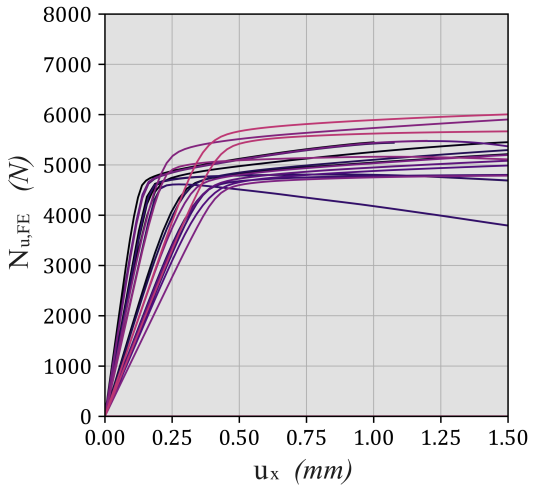
7.6.2 Plastic Capacity

Although the previous section showed that elastic buckling performance can be significantly enhanced through geometric optimisation, ultimate structural performance is governed by plastic capacity. Naraidoo & Rossi [24] characterised the ultimate failure of perforated plates by a variable Ω, and developed a DSM-style expression that can accurately predict the ultimate buckling capacity ($N_{u,h}$) when $\Omega \leq 0.265$ (provided that l_x, l_y, S_{yo} and S_{xo} are in units of mm). This variable will be used to categorise the optimised models

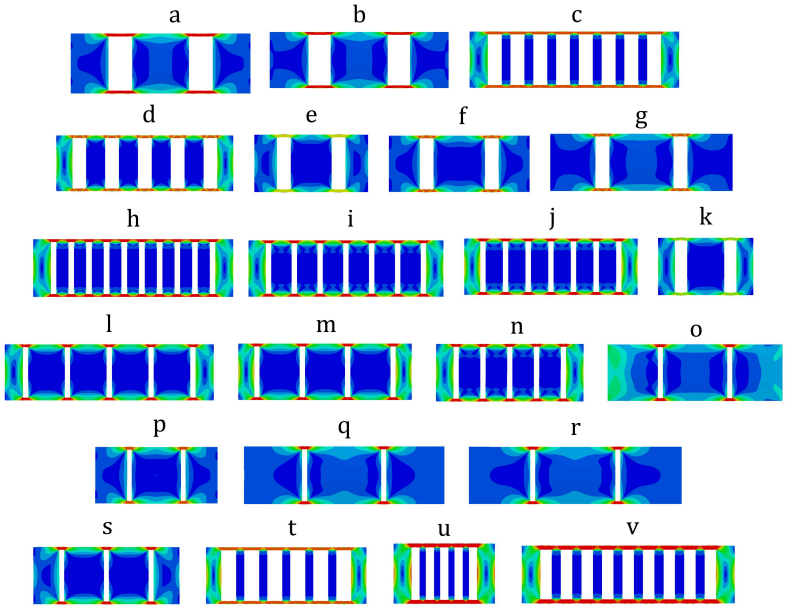
from Section 7.6.1, and examine how $N_{u,h}$ is related to perforation geometry. This will determine whether geometries that maximise $N_{cr,h}$ also maximise $N_{u,h}$, providing a complete picture on how manufacturers should design members with tall perforations subject to uniform compression.

$$\Omega = \frac{l_x}{(S_{yo})^2} + \frac{\sqrt{l_y}}{S_{xo}} \tag{7.9}$$

The force-displacement ($N_{u,FE}$-u_x) relationships of the 50 'optimised' models in Table 7.9 are therefore examined next, to assess whether gains in elastic buckling load translate into improved ultimate resistance. It is seen that 22 models possess an elasto-plastic capacity less than 7000 N (Group I, Figure 7.22), and 28 possess an elasto-plastic capacity between 7000 N and 12000 N (Group II, Figure 7.23). The models displayed in Figure 7.22b and Figure 7.23b are labelled alphabetically in order of increasing ultimate (plastic) capacity. Only 8 models possess $\Omega \leq 0.265$, all of which fail with a capacity between 7000 N and 12000 N (Group II), and the model with the smallest value of $\Omega = 0.151$ fails with the greatest capacity of 11700 N (Figure 7.23b).

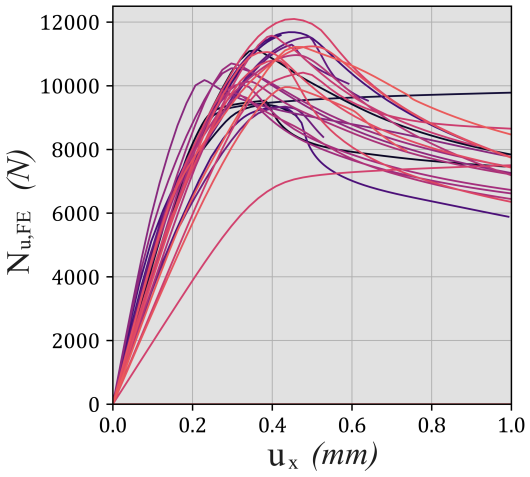


(a) Force-displacement relationship

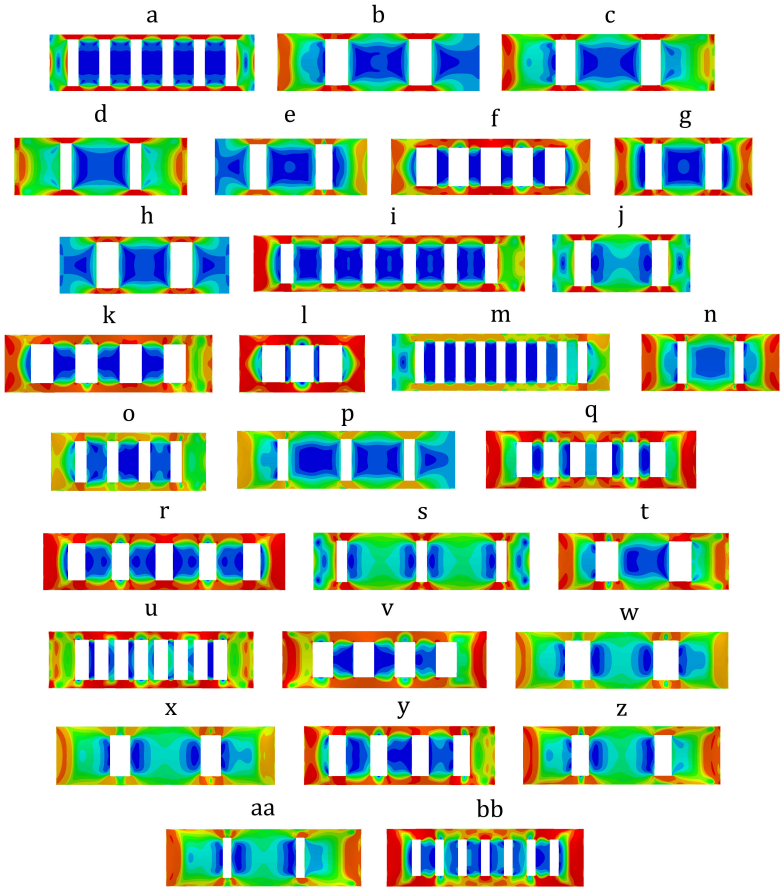


(b) Von-Mises stress at ultimate capacity

Figure 7.22: Group I (perforated plates with $N_{u,FE} < 7000$ N)



(a) Force-displacement relationship



(b) Von-Mises stress at ultimate capacity

Figure 7.23: Group II (perforated plates with $7000 \leq N_{u,FE} < 12000$ N)

The force-displacement trends for models in Group I with $N_{u,h} < 7000$ N (Figure 7.22) closely resemble the behaviour of an imperfect column deforming under second-order theory. These curves display a smooth nonlinear increase in load with displacement and no bifurcation. The tangent stiffness decreases progressively with increasing displacement, characteristic of a second-order response with geometric softening (i.e., the plate is becoming less effective at resisting additional load).

Comparatively, the majority of models in Group II with $7000 \leq N_{u,h} < 12000$ N (Figure 7.23) possess a distinct ultimate buckling capacity, followed by a limit-point instability and unstable post-buckling response, where the load drops sharply with little additional displacement (this softening behaviour is typical of slender plates with membrane action, Section 1.3.3).

The different mechanical behaviour in the two groups is governed by the perforation geometry, which differs on average. For models in Group I and Group II, the average ratio l_y/b is 0.90 and 0.72 respectively, and the average l_x/l_y ratio is 0.20 and 0.40 respectively. Thus, Group I plates contain taller openings, and narrower remaining ligaments (with width S_{yo}), promoting more one-dimensional load paths. Whereas Group II plates retain wider effective plate regions capable of two-dimensional stress redistribution.

The progressive stiffness degradation observed in Group I is characteristic of the P-delta effect in columns. This suggests that failure is governed by localised behaviour at the column-like strips of width S_{yo} above and below the perforations. The ultimate capacity of plates in Group II is greater because stresses are more uniformly spread over the entire geometry, due to post-buckling membrane redistribution.

These observations support the research by [39] who assert that perforated plates with a greater ultimate capacity enable a 'better distribution of the maximum stress i.e., there are more regions submitted to the maximum stress in the optimal geometries' [39]. Henceforth for clarity, 'Group I' is renamed 'localised failure' and 'Group II' is renamed 'globalised failure'.

The ratio S_{yo}/b is of additional importance to $N_{u,h}$, unlike $N_{cr,h}$; all models in Group I possess $S_{yo}/b = 0.05$ except for t, u and v which possess increasing S_{yo}/b (0.054, 0.063 and 0.072 respectively). This very small geometrical property is enough to cause a difference

in ultimate buckling capacity of 1220 N between model v and model c, despite them otherwise looking geometrically similar (Figure 7.22b).

Plates with either localised or globalised failure form two distinct clusters of data points when the ratio of $\min(N_{cr,FE}, N_{u,FE})$ to N_0 is plotted with respect to Ω (Figure 7.24). This shows that if $\Omega < 0.45$, the addition of perforations is beneficial to the elastic buckling load of an equivalent unperforated plate. The trends shown in Figure 7.24 and Figure 7.25 are coloured according to the out-of-plane local elastic buckling designation, shown for reference in Figure 7.26.

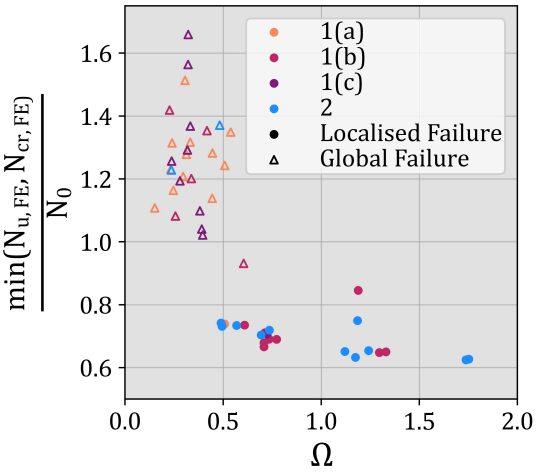


Figure 7.24: The ratio of $\min(N_{cr,FE}, N_{u,FE})$ to N_0 of plates exhibiting localised failure (circles) and global failure (triangles) form two clusters

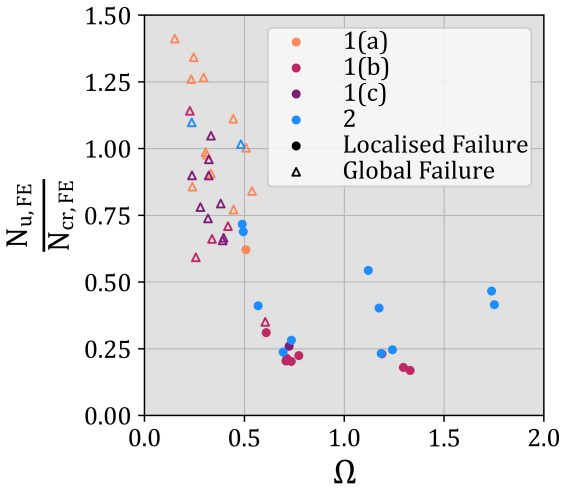
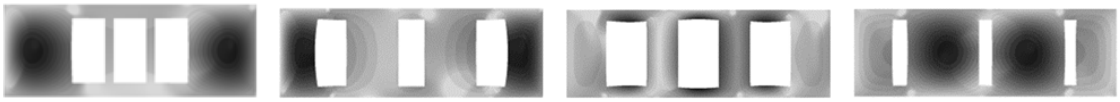


Figure 7.25: The plastic-to-elastic capacity ratio of models exhibiting localised failure (circles) and global failure (triangles)



(a) 1(a) **(b)** 1(b) **(c)** 1(c) **(d)** 2

Figure 7.26: The different out-of-plane buckling designations

As 80% of the dataset fail with $N_{u,h} < N_{cr,h}$ (Figure 7.25), this demonstrates that $N_{cr,h}$ alone is insufficient as a design criterion, and $N_{u,h}$ is governed by stress redistributions dependent on the perforation geometry.

Table 7.11 summarises the models with the greatest and smallest $N_{u,FE}/N_{cr,FE}$, and Table 7.12 summarises the models with the greatest and smallest $\min(N_{cr,FE}, N_{u,FE})$.

7. The Prediction and Classification of Local Buckling for Plates with Tall Perforations

It follows that perforated plates failing by designation 1(c) (i.e., plate-to-hole-edge out-of-plane buckling) can withstand the greatest compressive force (Table 7.12), whereas a geometry with many tensile elastic buckling modes is least optimal (Table 7.11 and Table 7.12). Please refer to Figure 7.22b and Figure 7.23b for images of the named models.

Table 7.11: Geometries with least and greatest $N_{u,FE}/N_{cr,FE}$ (shown in increasing order)

Name	Elastic Buckling Mode	First Compressive Eigenmode	Elastic Capacity (N/mm)	Plastic Capacity (N/mm)	Ω	a/b	l_x (mm)	l_y (mm)	S_{xo} (mm)	S_{yo} (mm)
Group I-d	1(b)	13	284	48.0	1.33	3.20	25	90	28.8	5
Group I-c	1(b)	11	266	47.9	1.230	3.75	25	90	32.1	5
Group I-j	1(b)	11	251	50.8	0.734	3.10	10	90	28.4	5
Group I-h	1(b)	11	243	49.9	0.708	3.50	10	90	30.8	5
Group I-i	1(b)	11	248	50.8	0.708	3.50	10	90	30.8	5
Group II-bb	1(b)	1	106	121	0.227	3.40	15.2	63	43.5	18.5
Group II-z	1(a)	1	91.9	116	0.233	3.40	30	70	83.6	15
Group II-w	1(a)	1	89.2	113	0.296	3.75	45	70	87.2	15
Group II-x	1(a)	1	85.9	115	0.246	3.75	35	70	92.2	15
Group II-aa	1(a)	1	82.7	117	0.151	3.40	15	70	98.6	15

Table 7.12: Geometries with least and greatest $\min(N_{cr,FE}\,,\, N_{u,FE})$ (shown in increasing order)

Name	Elastic Buckling Mode	First Compressive Eigenmode	Elastic Capacity (N/mm)	Plastic Capacity (N/mm)	Ω	a/b	l_x (mm)	l_y (mm)	S_{xo} (mm)	S_{yo} (mm)
Group I-a	2	4	11	46.1	1.75	3.00	40	90	62.7	5
Group I-b	2	4	98.9	46.1	1.74	3.20	40	90	69	5
Group I-k	2	14	201	46.9	1.18	1.68	24.6	99.4	32.3	5.3
Group I-c	1(b)	11	266	47.9	1.30	3.75	25	90	32.1	5
Group I-d	1(b)	13	284	48.0	1.33	3.20	25	90	28.8	5
Group II-i	1(b)	1	210	124	0.260	4.81	18.1	80	38.6	11.9
Group II-l	1(c)	1	160	127	0.381	2.19	31.9	80	31.8	14.3
Group II-q	1(c)	1	143	129	0.238	3.65	22.5	85	45.4	16.8
Group II-k	1(c)	1	203	135	0.396	3.60	29	75	34.2	12.4
Group II-r	1(c)	1	176	138	0.280	4.25	24.3	80	35	13.9

7.7 Conclusion

Structural members with tall perforations permit service integration and maintenance access, providing cellular beams with a greater second moment of area per unit weight than a solid section. However, the buckling response of these members subject to uniform compression is under-researched; many methods used to determine $N_{cr,h}$ are inapplicable to members exhibiting LPB, and design standards either make over-conservative approximations for plate behaviour between perforations, or only make provisions for buckling under flexure [6, 119, 159].

This study is the first to systematically categorise the buckling behaviour of plates subject to pure compression, with a series of tall rectangular openings. This study identifies five distinct buckling modes; four out-of-plane modes and one in-plane mode. Out-of-plane modes 1(a), 1(b) and 1(c) buckle at the plate-to-hole edge, and mode 2 buckles at the hole-to-hole edge.

A semi-analytical expression for $N_{cr,h}$ is derived for each designation of buckling failure, developed with three parametric studies. $N_{cr,h}$ for designation 1 geometries exhibits a strong relationship with the aspect ratio of the web ($S_{xo,i}/b$) characterised by (i) an almost-linear increase in capacity, (ii) a local maximum and (iii) an asymptotic decay towards the behaviour of a simply-supported unperforated plate. Conversely, the elastic buckling load of plates failing by designation 2 exhibits a strong relationship with the aspect ratio of the web-post (s_x/b), similar to the buckling curves of an unperforated plate. Finally, the behaviour of plates failing by designation 3 is similar to Euler column buckling.

A simple feed-forward neural network can reliably classify these failure types, informed by only four geometric variables with an error of only 10-20%. The larger error occurs when out-of-plane deformation at the plate-to-hole edge (designation 1) is targetted (Section 7.6.1), to optimise the geometry. In this case, future work should determine a statistical safety factor for reliable application of the classifier in practice.

On a validation set of 100 models, the expressions for $\widehat{N}_{cr,h}$ summarised in Table 7.7 determine $N_{cr,h}$ highly accurately, with $R^2 = 0.976$, $\mu = 1.06$ and COV$= 0.262$ in comparison to $N_{cr,FE}$. The capacity predicted for designation 3 exhibits the greatest uncertainty.

7. The Prediction and Classification of Local Buckling for Plates with Tall Perforations

Although geometries exhibiting a partial plate-to-hole buckling halfwave (designation 1(b)) can enable $N_{cr,h} \approx 3N_0$, these models fail with $N_{u,h} << N_{cr,h}$. The equivalent unperforated geometry would fail with $N_u > N_0$, thus this confirms the findings of [25] and [34], who showed how tall perforations reduce the 'slenderness' of a perforated plate. Therefore, $N_{u,h}$ must be considered when designing perforations in thin-walled sections, and this study finds that the variable Ω proposed by [24] can be used to distinguish models with a greater $N_{u,h}$; a designer should target geometries with $\Omega < 0.45$ to maximise $\min(N_{cr,h}, N_{u,h})/N_0$. However, note that the ultimate capacity of the unperforated plate will be greater than the perforated geometries.

When comparing $N_{cr,h}$ to $N_{u,h}$, the designer should target geometries with a full elastic buckling halfwave at the plate-to-hole edge (designation 1(a)), as this maximises $N_{u,FE}/N_{cr,FE}$. Plastic failure is not localised above and below the perforations in these stronger geometries. This aligns with the observations by [39], whereby stronger geometries enable a better globalised distribution of maximum stress around the perforation, particularly dependent on the variable S_{yo}/b.

Importantly, the ratio l_y/b for the most optimal geometries in Table 7.12 differ to the maximum recommended in Table 2.3 for rectangular perforations, indicating that modifications to design standards should be made. Additionally, the results of this study are valid only for simply supported plates with rectangular perforations subject to pure uniform compression, a case that is not currently included in [6]. Where additional actions are present, [6] specifies additional instability checks (critical when either flexure or shear is present). Of particular importance, Vierendeel bending may arise due to the transfer of shear forces across a given opening. In addition, serviceability checks are necessary to ensure that the maximum deflection remain within span/250 – span/150, depending on structural function [1].

8

Shape Annealing Optimisation of Perforations in Rack Sections Subject to Buckling

8.1 Commentary

The third challenge with the design and assessment of perforated plates lies in their optimisation. As introduced in Section 1.2, extensive parametric finite element (FE) studies are typically required to determine the size, shape and distribution of perforations that maximises $N_{cr,h}$. This design space is high-dimensional, and the buckling response is highly sensitive to geometric variation; deformation could be one of four sub modes, or an interaction of these, as shown in Chapter 4 and Chapter 7. Thus, systematic design space exploration is time-consuming and complex; global optimisation is impractical in routine design.

As discussed, Minera, et al. (2021) [46] and Gracia & Rammerstorfer (2019) [54] propose semi-analytical alternatives to guide perforation placement in buckling plates. Yet although these methods present an important step towards optimisation in absence of FE, they are constrained to plates with relatively few, small perforations. Numerical intervention is necessary if the perforations are large or many, reintroducing the computational burden these methods are intended to mitigate. By replacing the numerical aspects of optimisation with with heuristic ones, there is the ability to navigate complex design spaces and find

near-optimal solutions much faster; heuristic optimisation methods do not require gradient information, and are not restricted by analytical assumptions (Section 2.3).

Consequently, this chapter proposes to use the semi-empirical methods developed in Chapters 4 and 7 in replace of numerical methods traditionally used in optimisation, within a heuristic framework. These chapters introduced mode-classification methods and semi-empirical predictions for $N_{cr,h}$ that are reliable within a clearly defined domain, which encompasses a wide range of hole sizes and spacings typical of manufactured sections. As the elastic buckling load is typically less than the ultimate capacity (unless the plate is stocky, or perforations are particularly tall - see Chapter 7), it is safe practice to optimise structures at the serviceability limit state (i.e., elastic buckling) rather than the ultimate limit state. Nevertheless, future work should involve the variable Ω, and expressions developed for ultimate capacity (Chapter 6), to ensure that perforations do not cause stocky behaviour.

The optimisation framework presented in this chapter was conceptualised into 'Abalone' [29, 30], a plug-in for Grasshopper [31] (Appendix I). The code for this plug-in was independently written in Python by myself, and I collaborated with Dr. Ioannis Mirtsopoulos at ETH Zurich to translate my entire Python code into C#. I proposed the invention of this tool because Grasshopper is an intuitive visual programming language that is widely used in architecture, engineering, product design, construction, and digital fabrication workflows. Moreover, optimisation tools are more beneficial in practice when they are integrated within software with which structural engineers are already familiar; this minimises the learning curve, a key barrier to implementing optimisation in a real-world context [161]. At the time of development, there is no tool that enables engineers to optimise the arrangement of perforations within rack sections or cellular members subject to axial compression, which is a significant research gap. This open-source tool is protected by copyright under Oxford University Innovation project number 24955.

8.2 Introduction

Shape annealing is a stochastic heuristic optimisation technique concerned with optimising the arrangement of shapes according to a rule-based language specified by a 'shape grammar' i.e., an iterative application of shape transformation rules to a starting shape [96, 97]. It is particularly advantageous for the optimisation of the elastic buckling load ($N_{cr,h}$) of perforated plates, as numerical optimisation struggles to compute the many design variables and their spatially-varying relationship with $N_{cr,h}$. Most importantly, shape annealing is stochastic, which enables exploration of the design space, without converging prematurely to local optima.

To assess how well shape annealing can optimise a structural member, rack sections are considered as a case study. Rack sections are slender steel members used in warehouses for the storage of goods with medium to high loads. In practice, column and beam members are bolted together in diverse assemblages, enabled by the periodic arrangement of perforations along the member web. As a result, column members must support high loads in compression, and must be restrained against global buckling. At the same time, bracing must not obstruct access to shelving, and instead restrain members in the transverse direction (Figure 8.1).

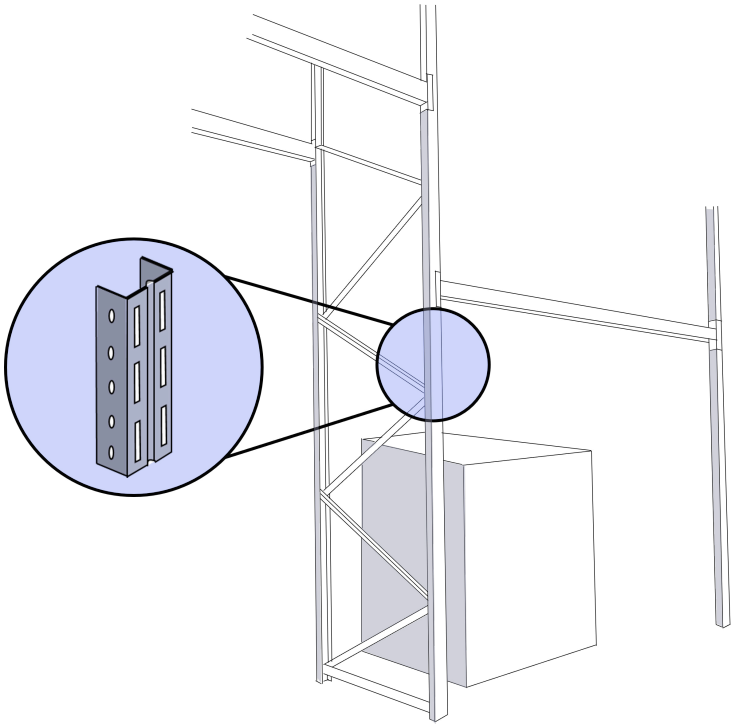


Figure 8.1: Rack frame assembly with perforated column highlighted, and storage access shown with the arrow

When loaded in compression, the webs of these members can be conservatively approximated as simply-supported plates (Section 2.4), in which the parallel rows of perforations required to locate beam-column fixtures may introduce modes of local buckling. Moreover, the perforation spacing is typically standardised by manufacturers to suit shelf increments and production, rather than for particular structural merit. As it is too expensive to stiffen each perforation locally, local elastic buckling resistance is instead achieved by stiffening the cross section.

Given the widespread use of rack systems, and the highly material-efficient nature of thin-walled members, even modest improvements in local buckling performance have the potential to deliver significant economic and safety benefits at scale. This paper therefore uses shape annealing to investigate whether the geometric arrangement of perforations can be systematically modified to enhance $N_{cr,h}$, in absence of cross-sectional stiffening, and while remaining compatible with practical manufacturing constraints.

8.3 Shape Annealing Procedure

Cagan & Mitchell (1993) [96] first proposed shape annealing optimisation, using geometric properties to formulate the objective function, rather than the random application of rules (as was the conventional shape grammar approach). In this procedure, an eligible rule is selected and used to modify the design state d_i. The 'energy' E_i at state d_i is compared to the 'energy' E_{i+1} at the resulting state d_{i+1}; if $E_{i+1} < E_i$ then the resulting design state is more feasible (for objective minimisation) and is accepted. However, if $E_{i+1} \geq E_i$ then the resulting design state may only be accepted according to a probability $p(E_{i+1})$, given in terms of the 'temperature profile'. A random number $r \in [0, 1]$ is generated and compared with $p(E_{i+1})$; if $r < p(E_{i+1})$, then the resulting state is accepted (despite it being 'less optimal' than the previous state) [96]. In this paper, a shape grammar algorithm is developed to optimise the elastic buckling efficiency of a plate with rectangular perforations, where the buckling efficiency is defined as:

$$\phi = \frac{N_{cr,h}}{N_0} \cdot \frac{A_0}{A_0 - A} \tag{8.1}$$

Here, $N_{cr,h}$ and $N_{cr,0}$ denote the elastic buckling load of the perforated and unperforated plate respectively, A_0 denotes the unperforated plate area, and A denotes the total area of the perforations. At each optimisation step, the efficiency at the end of the current step is compared to the efficiency of the previous step; it is then accepted according to the simulated annealing probability (p) [96]:

$$p(E_{i+1}) = \exp\left(\frac{\phi_{i+1} - \phi_i}{T}\right) > r \tag{8.2}$$

In Equation 8.2, the exponent $(\phi_{i+1} - \phi_i)/T$ controls the size of p, and T is a parameter which depends on the step number since the start. As the exponent reduces, it becomes more likely that a less efficient result will not be accepted, as $p < r$. More generally, it has been proven that if T is reduced sufficiently slowly, and a solution is reached at each T, the algorithm is guaranteed to determine the global optimum [162]. By selecting the best option in a population of solutions (generated by running the algorithm multiple times), our investigation will find near-optimal solutions.

This investigation expresses the simulated annealing probability $T(i)$ as either an exponential or a reciprocal function, given in terms of the 'cooling rate' (α), and the step number x.

$$T(i) = \alpha^x, \tag{8.3a}$$

$$T(i) = \frac{1}{1 + \alpha x} \tag{8.3b}$$

α is a parameter between 0 and 1; as $\alpha \to 1$, this increases the probability that a less efficient perforation arrangement is selected by the shape annealing framework. Thus, a value of $\alpha = 0.5$ is selected in this paper, as it balances efficiency with exploration of the design space.

The probability of acceptance $p(E_{i+1})$ is plotted with respect to step number x in Figure 8.2, considering an arbitrary difference in efficiency $\phi_{i+1} = 0.9\phi_i$.

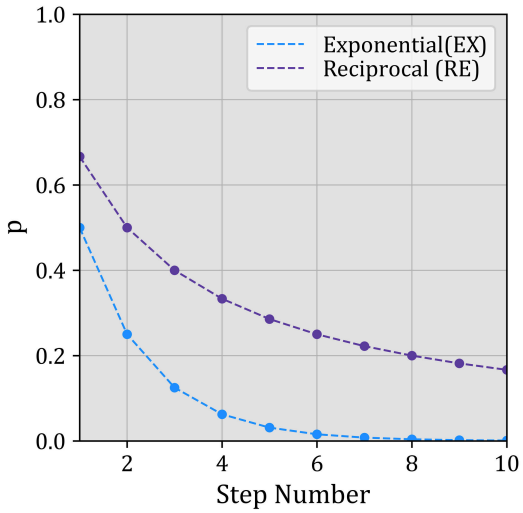


Figure 8.2: Probability of acceptance for $\phi_{i+1} = 0.9\phi_i$ as step number (x) varies

Within this framework, the geometry is optimised in an iterative stepwise manner; in any given step, a geometrical transformation is applied. This transformation is defined by a set of 'rules' that are independent of the current geometric state. In total, four possible transformation rules are considered; the addition of holes in the x-direction (i.e., an additional 'column' of perforations); the addition of holes in the y-direction (i.e., an additional 'row' of perforations); a reduction to the hole length by 10% and an increase to the hole length by 10%. These rules are denoted C, R, S and G respectively. This

allows one to optimise the size and number of perforations, but control their shape and aspect ratio ($r_a = l_x/l_y$), which is specified by the user in the primary state d_0; any change in size (S, G), or addition of perforations (C, R) will not affect r_a. Additionally, the configuration of perforations is one that is regular and symmetric about the x- and y-axes of the plate, as experimental tests have determined that an irregular arrangement of perforations is less structurally efficient [25]. Additionally, the regular configuration complies with traditional manufacturing rules (Figure 8.3).

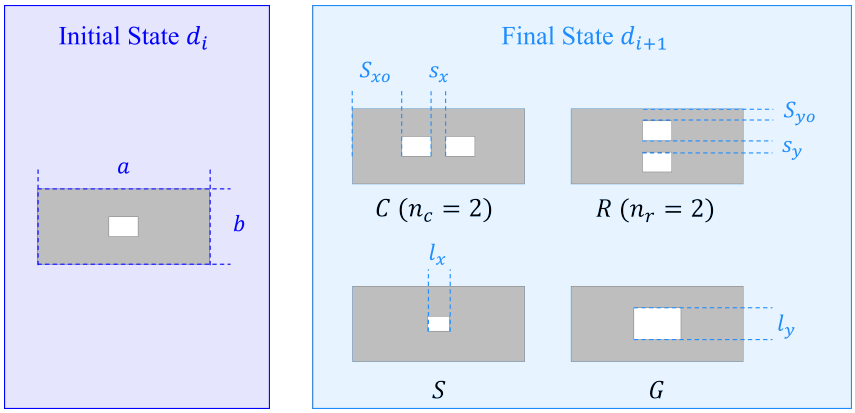


Figure 8.3: The initial state could be transformed to one of four final states depending on whether rule application violates geometric constraints. The number of columns is n_c and the number of rows is n_r

The optimisation procedure is controlled by geometrical and computational constraints. If application of a randomly chosen rule violates these geometrical constraints, then the rule is discarded, and an alternative rule is selected. These constraints are the minimum and maximum limits on inter-hole spacing (s_x, s_y), the spacing between a hole and an edge (S_{xo}, S_{yo}), and the size of the hole (l_x, l_y). If all rules violate the geometric constraints in a given step, then the procedure terminates. Otherwise, the procedure will only terminate after the maximum number of steps defined by the user. Once the geometry is successfully transformed, the efficiency of the new arrangement is predicted with Equation 8.1; the predicted efficiency is $\widehat{\phi}$.

As stated in the commentary to this chapter, heuristic optimisation requires repeated evaluations of the efficiency objective, which would be slow if elastic buckling load were determined with FE. For this reason, $N_{cr,h}$ is determined using equations derived in Chapter 4 [23], and Chapter 7, which provide reliable predictions within the domain

of interest. To find a near-optimal solution, a candidate pool of 50 results is collected for a given set of parameters which generates a range of solutions, and the best result is selected. The capacity predicted for the final result ($\widehat{N}_{cr,h}$) is verified with FE, and compared to that of an unperforated plate N_0. The FE linear perturbation model follows the linear perturbation study conducted in Section 7.3, and the material is steel, with a Young's modulus of 203.4 GPa and a Poisson's ratio of 0.3 (assumed to adhere to the stress-strain relationships governing isotropic materials).

8.4 Derivation of Spacing Heuristics for Optimisation

As detailed, the semi-empirical expressions derived in Chapters 4 and 7 will be used in the optimisation procedure to evaluate the elastic buckling load $N_{cr,h}$ at each optimisation step. However, these expressions do not prescribe where newly added perforations (from selection of the C or R rule) should be positioned, to maximise $N_{cr,h}$. To identify geometric trends between perforation spacing and $N_{cr,h}$, a parametric FE study was conducted offline, i.e., outside of the optimisation loop; no FE analyses are performed during optimisation. This study was used to formulate deterministic geometric heuristics and spacing rules that guide transformation rules C and R.

Rather than exhaustively sampling the entire mathematical design space, the domain of investigation reflects practically relevant rack section geometries, while accommodating other perforation arrangements (as this optimisation framework is not restricted to rack section design). Specifically, the selected ranges for a/b, n_r and n_c correspond to typical braced upright segments, where a small number of rows and columns are present along the effective buckling length. The hole dimensions (l_x, l_y) are based on standard rack connection apertures (Appendix H), with some extension beyond this range to establish the bounds of the spacing heuristics and accommodate other perforated members. Similarly, the majority of spacing parameters investigated (s_x, s_y, S_{xo}) reflect practical constraints on minimum edge distances, bolt clearances, and regular pitch spacing used in commercial rack systems.

The minimum-maximum ranges given subsequently in Table 8.1 and Table 8.3 therefore bound both tightly perforated configurations (small spacing, larger perforation density) and sparse layouts (larger spacing, fewer perforations), ensuring that the study captures

the full range of practically admissible designs. Importantly, these designs comply with the range of applicability for which the semi-empirical predictions used within the optimisation loop were developed.

8.4.1 Column Spacing s_x

To determine the optimal spacing s_x, 62 models with a slenderness of 100 and s_y/b of 0, 0.1 or 0.2 were examined; the variable domain is summarised in Table 8.1.

Table 8.1: Domain of variables examined to determine optimal spacing s_x (relevant parameters summarised in Figure 1.4)

	a/b	n_r	n_c	l_x/a	l_y/b	S_{xo}/b	s_x/b	s_y/b
Average	3.26	2.07	4.07	0.04	0.15	0.69	0.48	0.08
Minimum	3	1	3	0.03	0.05	0.025	0.05	0
Maximum	4	3	6	0.08	0.25	1.71	1.70	0.20

When $n_r = 1$, and $l_y/b \geq 0.6$, Chapter 7 identified that for a particular l_y/b, there exists an optimum ratio S_{xo}/b which maximises the elastic buckling load. However, when $l_y/b < 0.6$, this study finds that the elastic buckling load depends on the total distance between the centre of a perforation ($c_{x,i}$) and the nearest buckling node (c_0), i.e., the nodal distance spacings $\sum |c_{x,i} - c_0|$ (Figure 8.4). When $n_r = 1$, $N_{cr,h}$ increases as $\sum |c_{x,i} - c_0|$ reduces (C, G, in Figure 8.5(a)). However, when $n_r > 1$, the relationship between $\sum |c_{x,i} - c_0|$ and $N_{cr,h}$ depends on n_c. If $\sum |c_{x,i} - c_0|$ is minimised, $N_{cr,h}$ is maximised for plates with $n_r \leq 2$ and $n_c = 3$ (Figure 8.5(b)), whereas $N_{cr,h}$ is minimised for plates with $n_r > 3$ and $n_c = 4$ (Figure 8.5(c)), as l_y/b and l_x/l_y are varied.

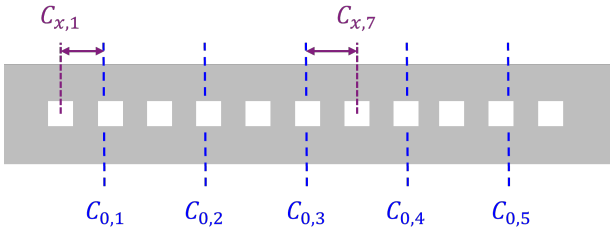


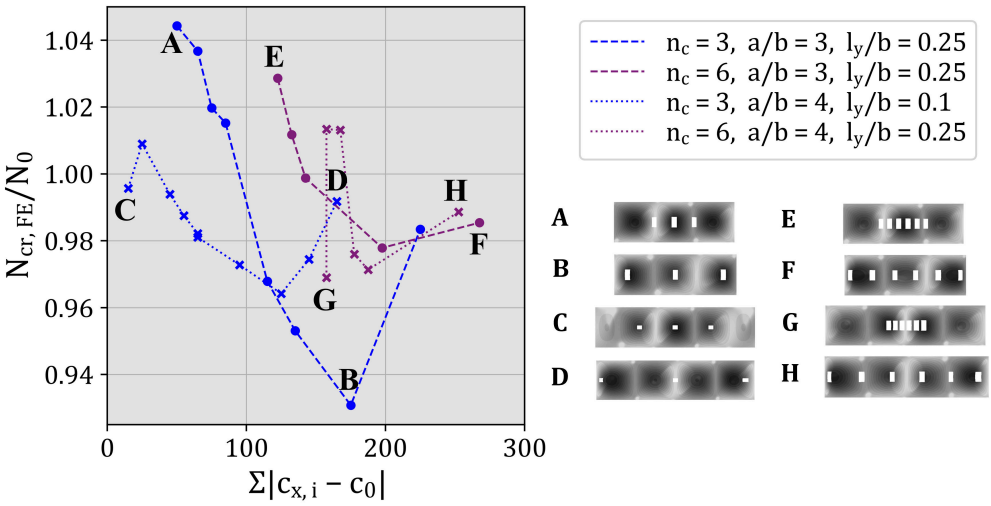
Figure 8.4: The nodal distance is shown by the arrow for the 1st and 7th hole with a centre at $c_{x,i}$. The location of the nodes for a plate buckling into 6 halfwaves are $c_{0,j}$

Based on the full parametric study, the spacing rules summarised in Table 8.2 are implemented within the heuristic optimisation loop. For clarity, only a representative subset of the analysed plates is presented in Figure 8.5; however, the trends underpinning these rules were observed consistently across the entire dataset explored. While the selected examples illustrate the governing behavioural patterns, further investigation over a wider range of n_r and n_c would strengthen the generality of the proposed inequalities. For plates with $l_y/b \geq 0.6$, $S_{xo,1}/b$ should be optimised, following Section 7.6.

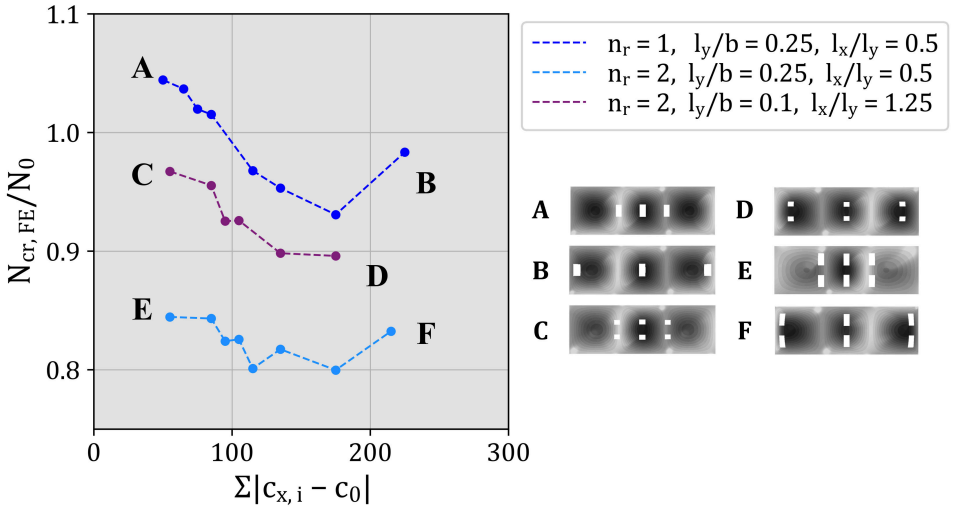
Table 8.2: Heuristic rules for s_x (relevant parameters summarised in Figure 1.4)

Case	n_c	n_r	l_y/b	l_x/l_y	m	$\sum \lvert c_{x,i} - c_o \rvert$
(i)	Any	1	< 0.6	Any	Any	Minimise$^{\Diamond}$
(ii)	≤ 3	> 1	< 0.6	Any	Any	Minimise$^{\Diamond}$
(iii)	> 3	> 1	< 0.6	Any	Any	Maximise
(iv)	Any	1	≥ 0.6	Any	Any	N/A

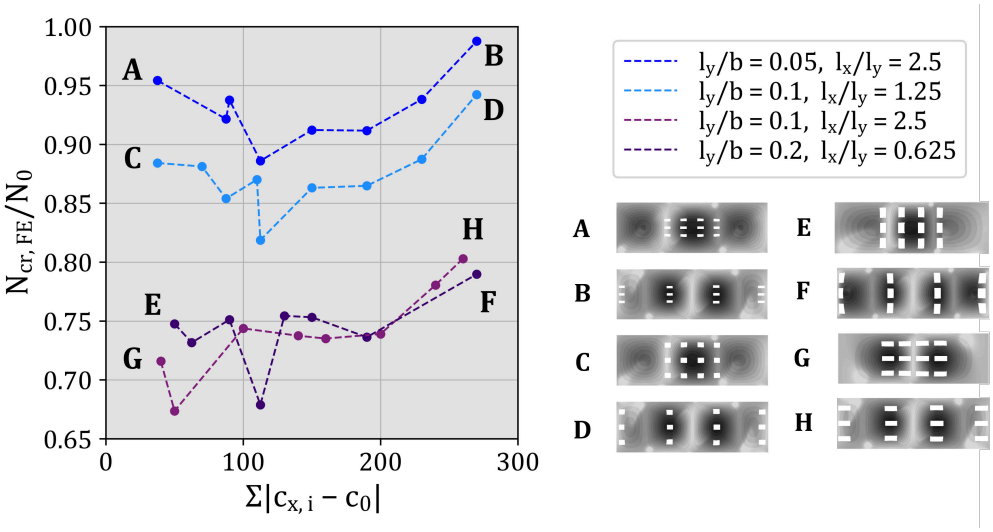
$^{\Diamond}$ if $l_x/b \leq 0.15$, limit $s_x > l_x$



(a) $\sum |c_{x,i} - c_o|$ for plates with $n_c = 1$



(b) $\sum |c_{x,i} - c_o|$ for plates with $a/b = 3$, $n_c = 3$



(c) $\sum |c_{x,i} - c_o|$ for plates with $a/b = 3$, $n_c = 4$ and $n_r = 3$

Figure 8.5: The relationship between capacity ratio and $\sum |c_{x,i} - c_o|$ for models examined in Section 8.4.1

8.4.2 Row Spacing s_y

To determine the optimal spacing s_y, 66 models with a slenderness of 100 were examined (Table 8.3); the relationship between s_y and the critical buckling ratio ($N_{cr,h}/N_0$) is shown in Figure 8.6. Observe that the narrow and small perforations displayed in Figure 8.6(b) are not strictly 'practical' for rack section design. However, this is necessary to establish the bounds of the heuristic relationships, as discussed. The optimal row spacing s_y is found to depend on the aspect ratio of the plate (the number of buckling halfwaves m), the hole height ratio (l_y/b) and the number of columns (n_c).

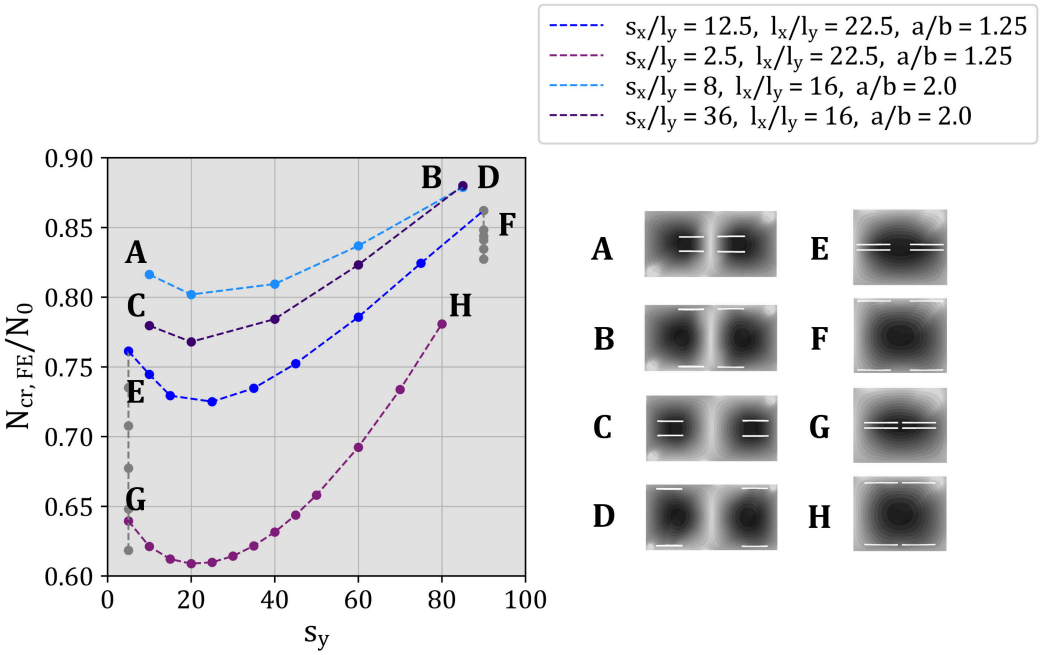
Table 8.3: Domain of variables examined to determine optimal spacing s_y (relevant parameters summarised in Figure 1.4)

	a/b	n_r	n_c	l_x/a	l_y/b	S_{xo}/b	s_x/b	s_y/b
Average	1.89	1.85	2.30	0.27	0.06	0.22	0.33	0.33
Minimum	1.25	1	2	0.144	0.008	0.01	0.01	0
Maximum	3	2	4	0.36	0.3	0.7	1.4	0.9

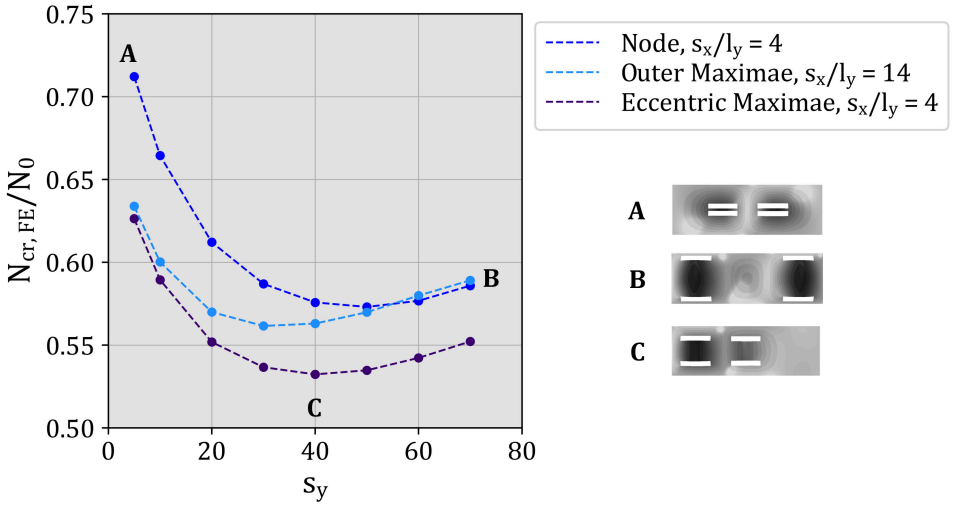
Unlike the previous section, the optimal s_y depends explicitly on the plate aspect ratio (through the number of buckling halfwaves m), in addition to other geometric parameters. Although only representative results are presented in Figure 8.6, the complete parametric study again revealed consistent behavioural trends across the full dataset. These trends lead to the following spacing rules summarised in Table 8.4. In all other configurations not included in this table, minimising s_y yielded higher elastic buckling capacities.

Table 8.4: Heuristic rules for s_y (relevant parameters summarised in Figure 1.4)

Case	n_c	n_r	l_y/b	l_x/l_y	m	s_y
(i)	1	1	Small (≤ 0.05)	any	1	Maximise
(ii)	> 1	1	Any	≤ 1	1	Maximise
(iii)	> 1	Any	Small (≤ 0.05)	any	2	Maximise
(iv)	> 1	2	Small (≤ 0.05)	any	≥ 3	Maximise



(a) $l_y/b = 0.02$, $n_c = n_r = 2$ and perforations located with $e_x = e_y = 0$



(b) $l_y/b = 0.1$, $l_x/l_y = 6$ and $a/b = 3$. Perforations located with $e_y = 0$ as e_x varies

Figure 8.6: The relationship between capacity ratio and s_y for models examined in Section 8.4.2

8.5 Shape Annealing Optimisation

It is desired to optimise the elastic buckling load to area ratio of a rack section web (Equation 8.1). This immediately imposes three practical design constraints. Firstly, there must be at least two rows of perforations over the web width, to permit connections with sufficient rotational restraint; secondly, the size of the perforations must accommodate common fixture sizes and meet manufacturing tolerances; and thirdly, there should be multiple perforations over the member length, to allow shelf beams to be positioned at various height increments.

In the following investigation, the result found with the shape annealing optimisation algorithm developed in Section 8.3 in combination with the spacing heuristics proposed in Section 8.4 is firstly compared to a deterministic rule-based optimisation procedure, to demonstrate that shape annealing outputs superior results. Then, shape annealing is used to optimise the arrangement of perforations within a rack section subject to buckling, constrained by design specifications and manufacturing standards.

It was chosen to optimise the elastic buckling load rather than the ultimate capacity in the following investigation The following optimisation investigation pertains to elastic buckling load only, additional checks should be completed by the designer to ensure that the perforated member does not fail plastically, or exhibit other instability phenomena such as flexure or shear. The result is verified using finite element (FE) analysis.

8.5.1 Comparison to a Deterministic Algorithm

Deterministic or 'greedy' rule-based optimisation procedures differ from shape annealing optimisation wherein the best result is always selected at the end of a given step, rather than randomly selecting any rule that does not violate design state criteria. Furthermore, if the final state is not more efficient than the initial state, the deterministic procedure will terminate; there is no ability to accept a less efficient rule according to a 'temperature profile'. Shape annealing optimisation would only be worthwhile if its stochastic properties cumulate in a more efficient final state (Figure 8.7). A case study compared the two approaches for a plate with geometric parameters summarised in Table 8.5.

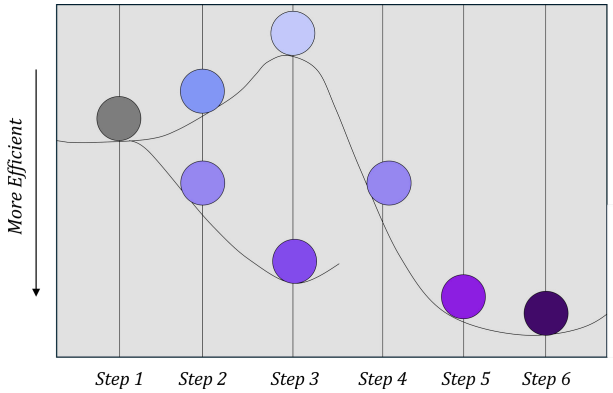


Figure 8.7: Depiction on how an initially less efficient path could terminate more efficiently

Table 8.5: Domain of variables examined in Section 8.5.1 (relevant parameters summarised in Figure 1.4). Units of *mm*.

a	b	t	$s_{x,min}$	$s_{x,max}$	$S_{xo,min}$	$s_{y,min}$	$s_{y,max}$	$S_{yo,min}$	$l_{x,min}$	$l_{y,min}$
250	100	1	10	250	10	5	100	5	0	20

The iteration begins with a single concentric perforation with $l_y/b = 0.75$, and $l_x/l_y = 0.5$ is targeted throughout. It takes less than one minute to run the shape annealing optimisation procedure 50 times, and the results with the three greatest values for the predicted efficiency ($\widehat{\phi}$) are shown in Table 8.6. The result found with the deterministic approach terminates after only four steps, and the result is equivalent to the worst result in Table 8.6. At this state, rules C, R and G cannot be applied without disobeying constraints on the geometry, and the efficiency of the final state after application of rule S is less than the previous state (Figure 8.8).

Table 8.6: The most efficient results found with shape annealing

Rank	Buckling Behaviour	Rules Chosen	$\widehat{\phi}$	ϕ
1		S, S, C, G, G, G, G, C, C	7.16	6.46
2		C, C, C, S, G, G, G	5.95	5.46
3		C, C, G, C	4.88	4.55

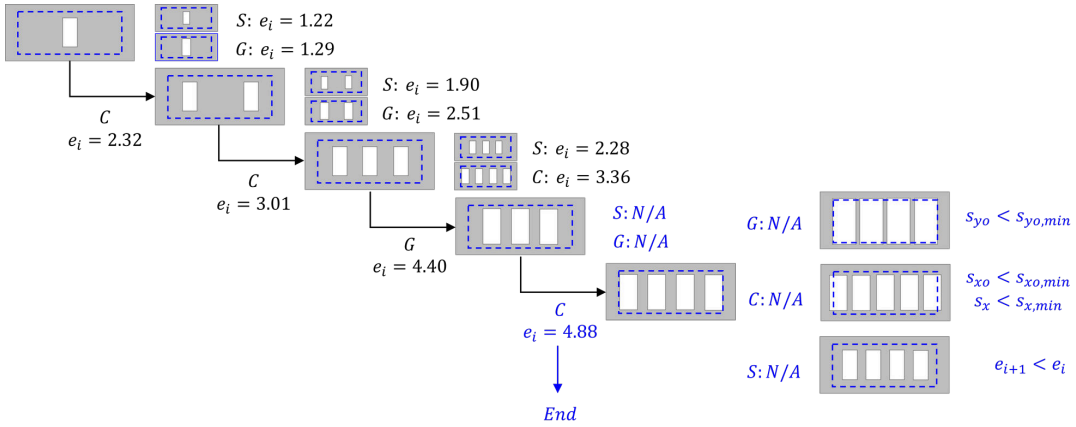


Figure 8.8: The most efficient result found with the deterministic approach

Observe that 2nd best result involves the growth (G) of perforations following a shrink (S) step. This growth does not 'undo' the shrink, as the two rules are not perfect opposites; the S rule multiplies the perforation length by 0.9, and the G rule multiplies the length by 1.1 (i.e., if $l_{x,i} = 1$ then $l_{x,i+2} = 0.99$ after G, S).

The efficiency of the shape grammar results are slightly over predicted (Table 8.1), which is attributed to the variability in the in-plane buckling mode prediction (Chapter 7). Nevertheless, this demonstrates that the stochastic and probabilistic shape grammar optimisation procedure produces more optimal results than a simple 'greedy' deterministic procedure.

8.5.2 Optimisation of a Rack Section

This chapter investigates whether shape grammar can be used to optimise the arrangement of perforations in a rack section. A total member length of $a = 1$ m is considered, corresponding to typical upright dimensions. A representative web width $b = 120$ mm and thickness $t = 2$ mm ensure that thin-shell Kirchhoff-Love theory can be considered, while complying with various rack section catalogues (Appendix H)(Table 8.7).

Although the resulting plate aspect ratio lies slightly beyond the parameter domain considered in the development of the semi-empirical predictions (Chapters 4 and 7), and the spacing heuristics study (Section 8.4), the geometry remains within the same structural regime (slender, regularly perforated plates in axial compression); thus the extrapolation is therefore limited and mechanically consistent, and the accuracy of the results is assessed later.

The spacings $s_{x,min}$, $S_{xo,min}$ and $S_{yo,min}$ are constrained by the recommended limits for cellular beams imposed by [6] and [119] (Table 8.8). Three case studies are assessed, each restricted by different $l_{x,min}$, $l_{y,min}$, $l_{y,i}$ and $n_{r,i}$ which are discussed in the relevant forthcoming section. The optimisation procedure is assessed for 100 steps, or until no rule can be applied; both exponential ('EX') and reciprocal ('RE') cooling are considered. Without changing the primary state d_0, the procedure is run 25 times, and the most optimal result is selected from the pool of candidate designs. The prediction found with shape annealing is compared to the FE result, using ABAQUS [150].

Table 8.7: Dimensions of rack section web to optimise

a(mm)	b(mm)	t(mm)	E(MPa)
1000	120	2	203400

Table 8.8: Geometric constraints on the rack section web to optimise (relevant parameters summarised in Figure 1.4)

$s_{x,min}$	$s_{x,max}$	$S_{xo,min}$	$s_{y,min}$	$s_{y,max}$	$S_{yo,min}$
$\max(0.5l_x, l_y)$	b	$\max(l_x, b)$	$0.125b$	$b/2$	$0.125b$

8.5.2.1 Case Study 1 - One Row of Perforations

To ease the transition of unperforated into perforated plate buckling behaviour, the member is initialised with a single concentric perforation with $l_y/b = 0.2$ (i.e., exhibiting WPB deformation, with $N_{cr,h} \approx N_0$). This provides a neutral baseline from which the influence of perforation geometry can be isolated as the geometry is optimised. To determine the relationship between ϕ and aspect ratio (r_a), holes with $r_a = 0.4, 0.6, 0.8, 1.0, 2.0$ and 4.0 are examined. The most efficient result for each aspect ratio determined with exponential and reciprocal cooling is shown in Figure 8.9. It is expected that different elastic buckling modes will become more prevalent as r_a is increased; for example, after successive perforation growth (G rule), plates with $r_a \leq 1$ are hypothesised to exhibit the LPB mode, whereas plates where $r_a > 1$ are hypothesised to exhibit the USB mode.

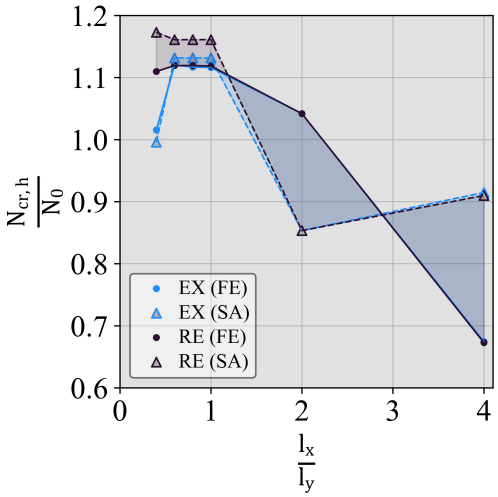
This hypothesis is predominantly true, and there is little difference in how the different cooling functions perform (Figure 8.9(a)). The accuracy of $N_{cr,FE}/N_0$ found with shape

annealing (SA) for perforations with $l_x/l_y < 2$ is high, and the absolute percentage error to the finite element (FE) result does not exceed 6%. However, the absolute % error becomes too significant for perforations with $l_x/l_y \geq 2$, unconservative by 35% for perforations with $l_x/l_y = 4$. This results in an average absolute error for $N_{cr,FE}/N_0$ of $10.8\% \pm 13.0\%$.

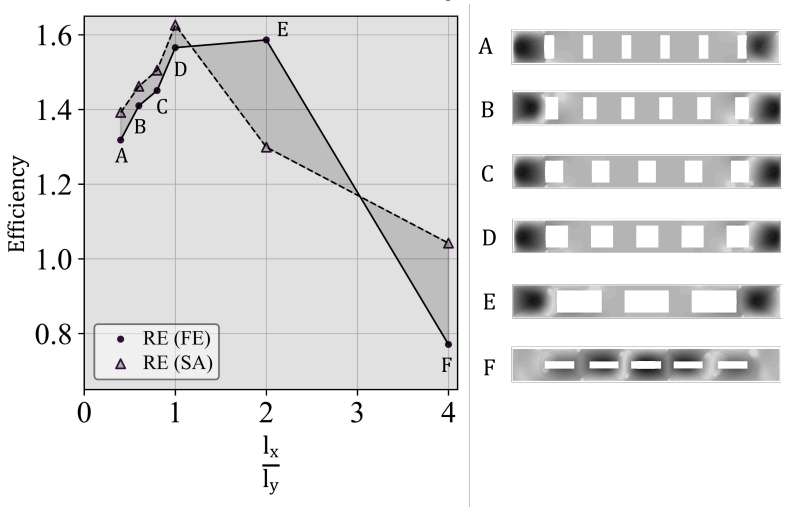
Table 8.9: Additional geometric constraints imposed on optimisation in Case Study 1 (Section 8.5.2.1) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm)

$l_{x,min}$	$l_{y,min}$	$l_{y,i}$	$n_{r,i}$	$n_{c,i}$
$5 \cdot \frac{l_x}{l_y}$	5	24	1	1

Reciprocal cooling (RE) gives slightly more efficient results for each r_a examined, shown in Figure 8.9(b). For each of these most optimal results A to E, the objective appears to be to maximise l_y subject to $S_{yo,min}$, which produces geometries that fail with the strongest elastic buckling mode (LPB-1(a), Chapter 7). However, the plate with the largest hole aspect ratio fails by USB (Figure 8.9(b)-F), suggesting that the perforation could not grow tall enough to incite LPB without violating geometric constraints. The final geometry is the least efficient of all results, suggesting that large holes with perforations where $r_a \approx 1$ are most optimal.



(a) Capacity ratio of best result in Case Study 1 found with different cooling functions as l_x/l_y varies



(b) Efficiency of best result in Case Study 1 as l_x/l_y varies

Figure 8.9: Results found in Case Study 1

8.5.2.2 Case Study 2 - One Column of Perforations

None of the optimal results in Section 8.5.2.1 can be distinguished as a rack section, which are characterised by at least two rows of perforations for fixture attachment. Therefore, Case Study 2 will initialise optimisation with two rows of perforations, in an attempt to steer optimisation towards eligible results (Table 8.10). Again, aspect ratios l_x/l_y of 0.4, 0.6, 0.8, 1.0, 2.0 and 4.0 are examined.

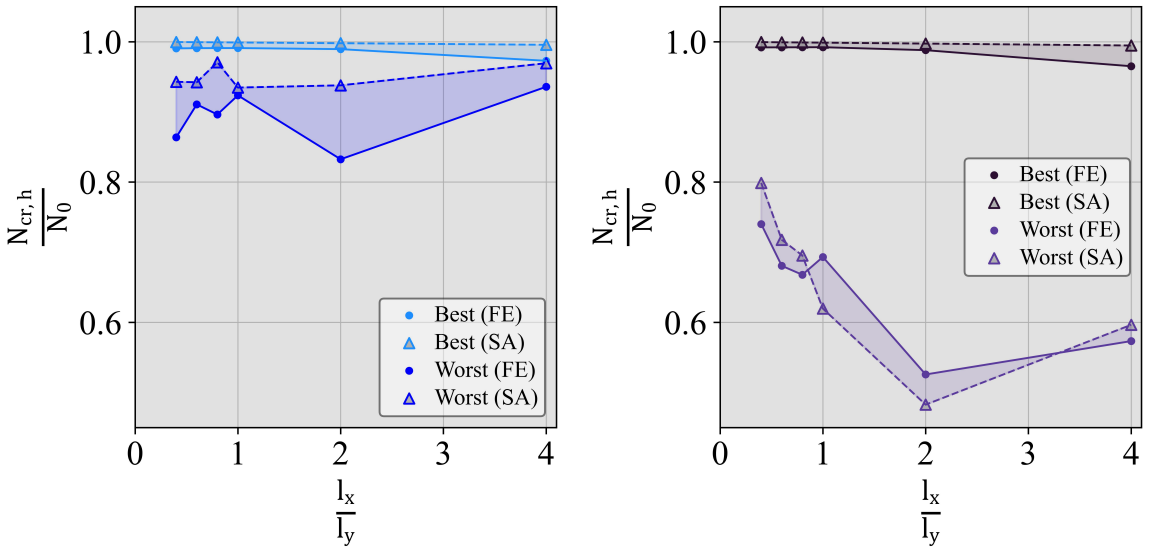
Table 8.10: Additional geometric constraints imposed on optimisation in Case Study 2 (Section 8.5.2.2) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm)

$l_{x,min}$	$l_{y,min}$	$l_{y,i}$	$n_{r,i}$	$n_{c,i}$
$5 \cdot \frac{l_x}{l_y}$	5	24	2	1

Recall that, for each aspect ratio examined, a pool of candidate configurations is generated and the configuration with the greatest predicted efficiency is selected (Section 8.5.2). To assess whether the optimisation procedure identifies genuinely high-performing solutions, the most efficient (best) and least efficient (worst) predictions within each candidate pool are compared to FE predictions for each cooling function (Figure 8.10). The results show that configurations predicted to be more efficient consistently exhibit higher FE buckling capacities than those predicted to be less efficient, indicating that the shape annealing framework reliably distinguishes between favourable and unfavourable geometries, and converges toward local optima.

Of the least efficient ('worst') results, those found with exponential cooling perform significantly better than those found with reciprocal cooling, suggesting that exponential cooling can find a local optimum in fewer runs (compare (a) and (b) in Figure 8.10).

Of the most efficient results, the capacity predicted with shape annealing is found with less than 3% error to the FE results, and the total perforated area is very small (Figure 8.11). Subject to the geometric constraints in Table 8.8, no perforated configuration with two rows exceeds the efficiency of the unperforated plate in this case study. This suggests that two-row perforation layouts are unlikely to be more efficient than an unperforated plate, under the geometric constraint that $l_{y,min} = 5$mm.



(a) Exponential cooling function

(b) Reciprocal cooling function

Figure 8.10: 'Best' and 'worst' prediction in the result pool as l_x/l_y varies

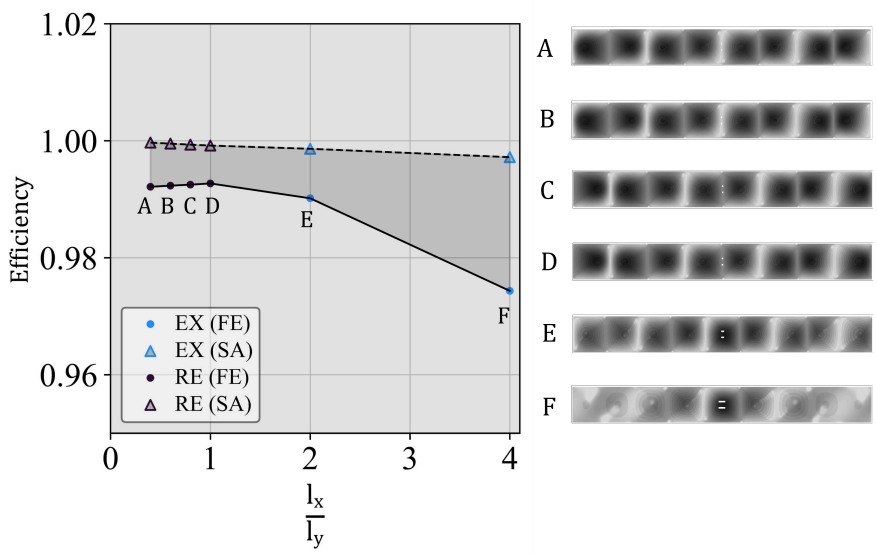


Figure 8.11: Efficiency of best result in Case Study 2 as l_x/l_y varies

8.5.2.3 Case Study 3 - Multiple Rows and Columns of Perforations

The optimised geometries from the previous two sections are not eligible rack sections; the objective function given in Equation 8.2 does not favour more perforations, and the minimum hole dimensions examined are not feasible in practice. Under these constraints and objectives, practical two-row perforation arrangements are unlikely to be optimal.

Thus, to force the solution towards one representative of a rack section with multiple columns of perforations in two rows, the minimum hole dimension and objective function must change. This third case study uses the sigmoid function to encourage a greater number of columns n_c.

$$\phi = \lambda \cdot \frac{P_{cr}}{P_{cr,0}} \cdot \frac{A_0}{A_0 - A}, \tag{8.4a}$$

$$\lambda = \frac{1}{1 + e^{(k - n_c)}} \tag{8.4b}$$

Increasing k shifts the sigmoid transition to larger n_c, delaying the increase in λ and producing a more gradual variation of λ over the practical range of column numbers (Figure 8.12); a value of $k = 4$ is adopted in this study, as this encourages the optimiser to explore layouts with n_c typical of rack section uprights, without an excessive bias towards very large n_c.

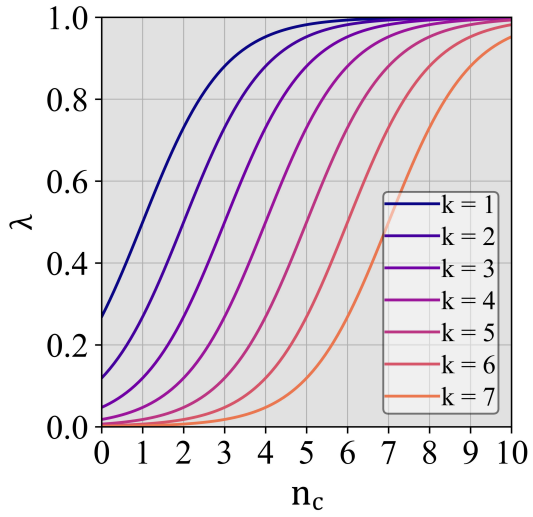


Figure 8.12: How the sigmoid factor λ on the objective function varies with n_c (Equation 8.4)

As considered in the previous case study, the plate is initialised with $n_{r,i} = 2$, and additional geometrical constraints are given in Table 8.11. The minimum hole dimension $l_{x,min}$, $l_{y,min}$ is raised to 35mm (depending on the hole aspect ratio), forcing the search away from small hole solutions, and arrangements that are more feasible for rack section fixtures. The starting hole dimension $l_{x,i}$, $l_{y,i}$ is equal to this minimum dimension, and holes with aspect ratios of 0.4, 0.6, 0.8, 1.0, 2.0, 3.0 and 4.0 are examined.

Table 8.11: Additional geometric constraints imposed on optimisation in Case Study 3 (Section 8.5.2.3) (relevant parameters summarised in Figure 1.4 and all lengths (l) in mm)

$l_{x,min}$	$l_{y,min}$	$n_{r,i}$	$n_{c,i}$
$35 \cdot \frac{l_x}{l_y}$ if $\frac{l_x}{l_y} < 1$	35 if $\frac{l_x}{l_y} < 1$	2	1
35 if $\frac{l_x}{l_y} \geq 1$	$35 \cdot \frac{l_y}{l_x}$ if $\frac{l_x}{l_y}$	2	1

Figure 8.13 displays the results found in Case Study 3 in terms of the original efficiency function (Equation 8.1), so that the geometric efficiency of the results can be compared independently of the modified objective weighting. All results are less efficient than those in Case Study 2 (Section 8.5.2.2), except for the plate with $r_a = 0.4$, which even exceeds the efficiency of an unperforated plate. This is the only result where perforations are concentrated away from the 'end panel' (Figure 2.4 and Figure 8.14), supporting the findings of [46] and [72].

This beneficial topology was found because the optimiser favoured plates with $n_c > 1$, due to the sigmoid weighting in the objective function (Equation 8.4). Consequently, the search explored regions of the design space which the optimiser in Case Study 2 either avoided or did not 'see', as the design space was crowded out by solutions with very small perforations, or which the objective did not favour. This finding demonstrates the importance of objective weighting, search behaviour, and stochastic sampling in determining which regions of the design space are explored.

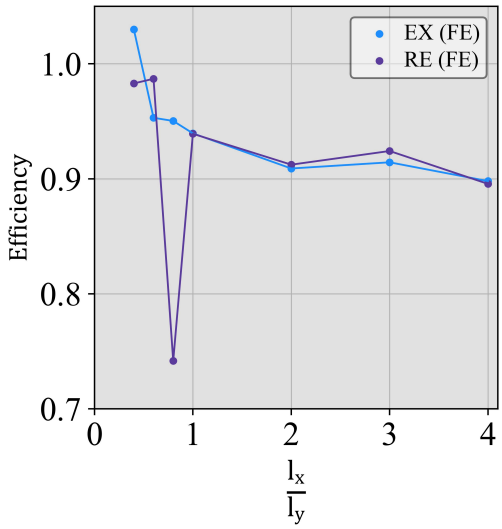


Figure 8.13: Efficiency given by Equation 8.2 for the best results found in Case Study 3, as l_x/l_y varies

The empirical predictions for $N_{cr,h}$ that underpin the shape annealing framework result in a conservative prediction for the efficiency in this case study on average, when compared to the FE result (Figure 8.14). Each of the optimal geometries contain perforations with the minimum permissible l_x and l_y, and are spaced at the smallest permissible s_y.

The results found with reciprocal and exponential cooling exhibit little difference, except for $r_a = 0.4$ and $r_a = 0.8$ (Figure 8.15). Fewer perforations with $r_a = 0.4$ spaced with smaller s_x exhibit a 4.5% improved efficiency. However, a greater difference in efficiency is seen for $r_a = 0.8$, where only the row spacing changes; the reciprocal cooling result has s_y that is 2.9 times greater than the exponential cooling result, performing 21% less efficiently. These findings demonstrate how sensitive the elastic buckling load of a perforated plate is to various geometric variables, but that this sensitivity is correctly identified with the shape annealing procedure proposed in this chapter.

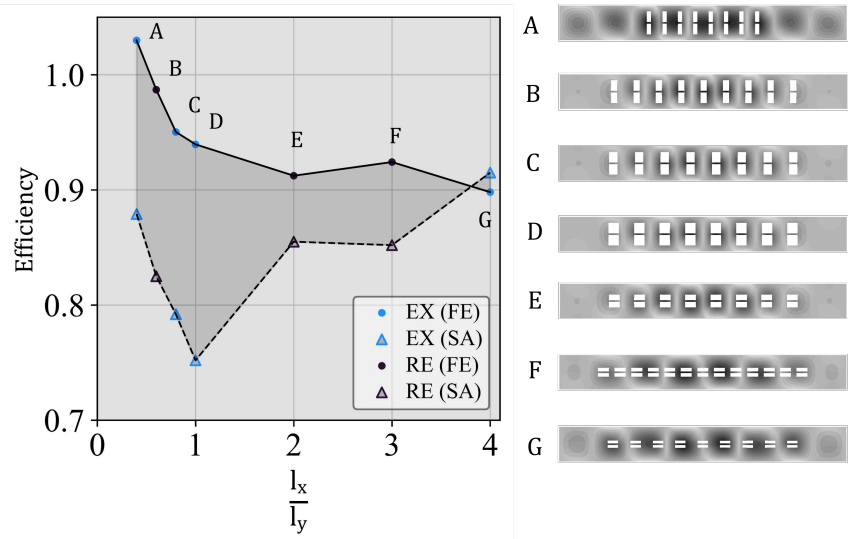
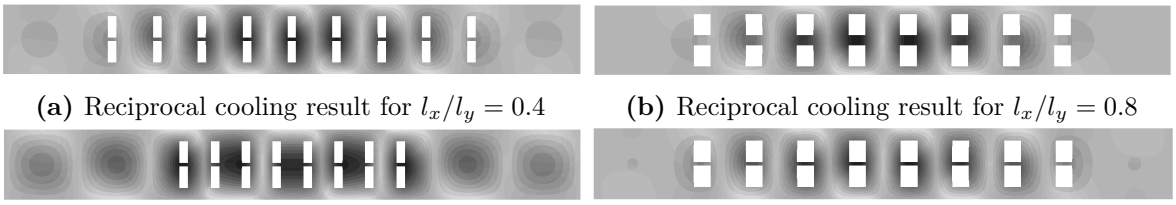


Figure 8.14: Efficiency of best result in Case Study 3 as l_x/l_y varies



(a) Reciprocal cooling result for $l_x/l_y = 0.4$

(b) Reciprocal cooling result for $l_x/l_y = 0.8$

(c) Exponential cooling result for $l_x/l_y = 0.4$, see Figure 8.14-A

(d) Exponential cooling result for $l_x/l_y = 0.8$, see Figure 8.14-C

Figure 8.15: The optimal result found with reciprocal and exponential cooling for (a) $l_x/l_y = 0.4$ and (b) $l_x/l_y = 0.8$

8.5.3 Blended Geometries

The shape annealing optimisation pipeline developed in this paper is restricted to plates with perforations at uniform spacing with uniform size. However, although rack sections require two rows of perforations, these do not have to be uniformly sized, or distributed over the entire member length. Indeed, the results of Case Study 3 suggest that distributing perforations over the full member length may be suboptimal (compare all results in Figure 8.14, for which A is most optimal).

This final section therefore explores whether the favourable characteristics of the optimal geometries identified in Case Study 1 (g_1, Section 8.5.2.1) and Case Study 3 (g_3, Section 8.5.2.3) by using the shape annealing pipeline, can be combined. The results

identified in Case Study 2 are not considered, as these perforations are too small and too few, as discussed. Specifically, g_1 and g_3 are 'blended' to produce layouts satisfying rack section manufacturing requirements (i.e., two rows of perforations), while permitting non-uniform hole size and distribution over the member length.

There are infinitely many ways in which the 'parent' geometries g_1 and g_3 could be combined. Therefore, this study examines only a limited set of representative 'blended' configurations. The inter-hole spacing is denoted by the subscript 'ij' (for example $s_{x,11}$ is the spacing between two g_1 holes).

The efficiency determined with FE is compared to the results in case studies 1 to 3 with respect to the average hole aspect ratio in Figure 8.16. The results indicate that the efficiency of the blended geometries lies between those of the parent configurations, exceeding the efficiency of results found in case studies 2 and 3. Only two of the nine blended geometries exhibit efficiencies lower than that of an unperforated plate.

The best result (blended geometry B, Table 8.12) combines the most optimal geometry from Case Study 1 with the most optimal geometry from Case Study 3, achieving a 35% greater efficiency than an unperforated plate. Physically, this improvement arises because the tall perforations from Case Study 1 isolate where deformation can occur; deformation is located towards the plate centre in the stronger blended geometries, as two rows of smaller perforations are 'bounded' by the taller ones, which dampen deformation [2, 35] (A, B, E and G, Table 8.12). This benefit outweighs the benefit observed in Case Study 3, i.e., members with a greater S_{xo} do not strictly perform more efficiently (such as geometry C).

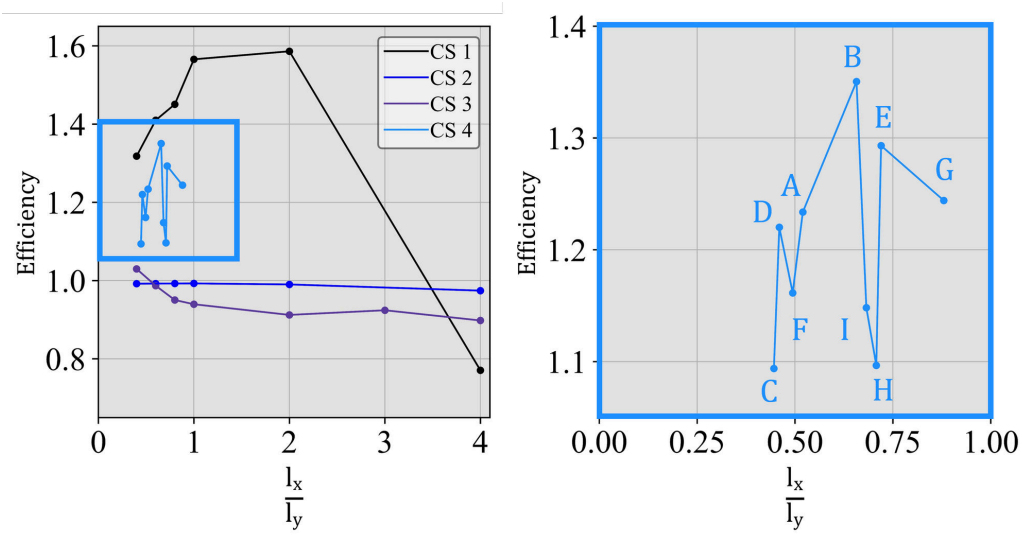


Figure 8.16: Efficiency of all Case Studies (CS) found with FE, plotted with respect to average hole aspect ratio

Table 8.12: Blended geometries in Case Study 4

Name	$g_1^{\diamondsuit}$	$g_3^{\spadesuit}$	Blended Geometry
A	D	A	 $s_{x,12} = 85.5$ mm $s_{x,22} = 19.5$ mm $s_y = 6$ mm
B	D	A	 $s_{x,12} = 120.6$ mm $s_{x,22} = N/A$ $s_y = 6$ mm
C	D	A	 $s_{x,12} = 39.1$ mm $s_{x,22} = 39.1$ mm $s_y = 6$ mm
D	D	A	 $s_{x,12} = 46.3$ mm $s_{x,22} = 46.3$ mm $s_y = 6$ mm
E	E	A	 $s_{x,12} = 85.6$ mm $s_{x,22} = 35.8$ mm $s_y = 6$ mm
F	E	A	 $s_{x,12} = 85.6$ mm $s_{x,22} = 35.8$ mm $s_y = 6$ mm
G	E	B	 $s_{x,12} = 81.9$ mm $s_{x,22} = 28.4$ mm $s_y = 6$ mm
H	E	B	 $s_{x,12} = 61.9$ mm $s_{x,22} = 55.9$ mm $s_y = 6$ mm
I	E	B	 $s_{x,12} = 78.2$ mm $s_{x,22} = 28.4$ mm $s_y = 6$ mm

$\diamondsuit$ see Figure 8.9, $\spadesuit$ see Figure 8.14

8.6 Conclusion

This work uses equations for $N_{cr,h}$ developed in Chapter 4 [23], and Chapter 7 as a surrogate for finite element analysis, within a shape annealing optimisation pipeline. By replacing repeated FE evaluations with semi-empirical predictions, the proposed pipeline enables a rapid exploration of perforation layouts, while retaining sensitivity to the governing geometric parameters. Importantly, shape annealing imposes stochasticity, which can find more efficient results than a deterministic optimisation algorithm.

Regarding the optimisation procedure itself, the choice of cooling schedule influences both the diversity of candidate solutions, and the final efficiency obtained. Although reciprocal cooling produces slightly more efficient results in some cases, exponential cooling yields more consistent performance when rack-type geometries are required. It is recommended to use the exponential cooling function with the parameters given in Section 8.3 if a smaller candidate pool is to be generated, as this gives a smaller range of best-to-worst candidates than reciprocal cooling (Section 8.5.2.2).

For geometries with particular geometric constraints initialised with $n_{r,i} = 1$, Case Study 1 (Section 8.5.2.1) finds that it is most optimal for perforations to be large, and arranged in a single row along the plate centreline. However, this study demonstrates that the shape annealing optimisation pipeline is only reliable for perforations with $r_a < 2$ when $l_y/b \geq 0.6$ and $n_r = 1$. These results can enable a 50% increase in efficiency compared to an unperforated plate, but these arrangements are not characteristic of rack section webs, which require multiple rows of small perforations to accommodate beam-column fixtures.

When the same geometric constraints are considered with $n_{r,i} = 2$, Case Study 2 (Section 8.5.2.2) shows that the optimisation procedure converges to two perforations of the minimum allowable size, indicating that the most efficient configurations minimise perforation area when multiple rows are required. To steer optimisation towards practical rack geometries, Case Study 3 (Section 8.5.2.3) introduces a sigmoid weighting factor λ in the objective function, and increases the minimum permissible hole dimension. Although this restricts the feasible geometry set, it exposes alternative topology families that are not explored in Case Study 2. Despite these modifications, Case Study 2 finds more

efficient results within the range of parameters examined, except for the configuration with $r_a = 0.4$, which is more efficient than an unperforated plate.

The difference in outcomes between Case Studies 2 and 3 arise primarily from optimisation dynamics, rather than feasible set size alone. In Case Study 2, the optimiser concentrated its search in regions of the design space dominated by low perforation-area solutions, thus did not explore the family of larger-hole and greater-column topologies that were forced (and rewarded) in Case Study 3. Thus, different objective functions and geometric constraints should be considered when shape annealing optimisation is implemented in practice.

In all case studies, it is generally more optimal to have perforations with $r_a \leq 1$. Yet, as optimisation becomes more constrained to rack section geometries, the average efficiency reduces from 1.3 in Case Study 1, to 0.99 in Case Study 2, and 0.93 in Case Study 3. This reflects the structural penalty associated with practical manufacturing constraints (subject to the geometric constraints of this study).

However, if the conventional requirement for uniform perforation size and spacing is relaxed, the results of Section 8.5.3 demonstrate that more efficient rack-compatible geometries can be achieved. By blending the optimal geometries from Case Study 1 with those in Case Study 3, efficiencies of approximately 1.2 can be obtained (thus exceeding the efficiency of unperforated plates). Large openings bring the merits of Case Study 1 (high efficiency), whereas small openings bring the merits of Case Study 3 (fixture attachment zones). This demonstrates that non-uniform perforation arrangements may offer a promising design strategy for improving the structural efficiency of rack section webs, and shape annealing can inform the best 'parents' for these blended geometries.

Overall, this study establishes both a practical optimisation methodology, and new insights into the structural behaviour of perforated rack section webs. The results demonstrate that shape annealing optimisation provides a powerful and computationally efficient framework for exploring perforation layouts in thin-walled plates. It can successfully identify efficient perforation arrangements, provided that the objective function, cooling schedule, and geometric constraints are chosen appropriately. Future work should implement equations developed in Chapter 6 [24] within the shape annealing pipeline,

to ensure that $N_{u,h} > N_{cr,h}$ (particularly for plates with tall perforations, as this is shown to be critical in Chapter 7).

9

A Physics Informed Neural Network for the elastic buckling load of a Perforated Plate

9.1 Commentary

Section 1.2 outlined three issues with the assessment of perforated plate buckling, of which chapters 4 to 8 address the first and third. The second (and perhaps the most critical) issue is first investigated in Chapter 3, and now concluded in this final results chapter. Chapter 3 introduced the MEP which was found to accurately predict $N_{cr,h}$ when deformation is WPB, and rivals the empirical prediction by [4]. Yet it remains far from contending with the accuracy, and versatility of numerical FE procedure.

Since the turn of the century, there has been an exponential growth in deep-learning research, of which physics-informed neural networks (PINNs) form a significantly younger, smaller subset. Conceptualised less than a decade ago, PINNs encompass several advantages set to challenge the domination of FE in structural engineering analysis. PINNs are mesh-free, (particularly beneficial for modelling complex geometries, such as those with perforations) and provide a comparably inexpensive prediction once trained. This will greatly advantage structural optimisation, which is impeded by the computational demand of FE.

This final chapter demonstrates these benefits for the prediction of the elastic buckling load $\widehat{N}_{cr,h}$ of a plate with an eccentric square perforation. The MEP proposed in Chapter 3 [26], the standard-ready empirical prediction proposed in Chapter 4 [23], and the PINN

developed herein are compared to FE results. The PINN estimate is this closest of the three to the FE results, within the domain of applicability for which it was developed. Therefore, this research does not propose a complete replacement for numerical analysis, but rather demonstrates the potential for a suitably trained PINN to act as a surrogate for FE analysis within a defined problem domain. The results presented in this thesis demonstrate that PINNs can reproduce FE predictions with reasonable accuracy for the cases considered, although further validation is required before equivalent accuracy can be assumed more generally. Furthermore, unlike purely data-driven machine learning models, PINNs incorporate governing physical relationships into the training process, providing a degree of physical interpretability and consistency with established mechanics. The python scripts used to develop this PINN are presented in Appendix J

9.2 Introduction

Conventional neural networks rely exclusively on large quantities of labelled data, whereas physics informed neural networks (PINNs) embed governing differential equations directly into the training process through a loss function. By enforcing equilibrium, compatibility, and satisfying boundary conditions, PINNs produce solutions that are physically admissible by construction. This physics-based regularisation significantly reduces data requirements, and improves generalisation beyond the training domain.

Since their introduction less than a decade ago, research on PINNs is growing, from 22 papers published in 2019, to 366 in 2022, and over 800 in 2025 [163, 164]. This research applies to a vast domain, including quantum physics, fluid mechanics, biomedical engineering, electromagnetism, and renewable energy. Despite this rapid growth, the application of PINNs to classical stability problems in structural mechanics remains comparatively underexplored.

Differential equations can be solved to determine the elastic buckling load of unperforated plates. However, perforations further complicate the analytical problem by introducing geometric discontinuities that disrupt stress fields and invalidate many simplifying assumptions considered in analytic formulations. Engineers therefore undertake numerical finite element (FE) simulations to determine the elastic buckling load of

perforated plates. However, FE is computationally expensive, and sensitive to modelling choices, and complex geometries can be difficult to mesh.

Yet PINNs completely mitigate issues with mesh artefacts, selecting the correct element, and mesh sensitivity studies typical of FE modelling. This is because PINNs are 'mesh-free'; collocation points (where the differential equation or residual is evaluated) exist in a bounded region, but do not come with element connectivity, and thus do not compose a mesh in a FE sense. Moreover, as they do not require labelled data from simulations or experiments, PINNs could accelerate the design of perforated plated structures by converting the complex problem of directly solving analytical equations, or the slow problem of FE analysis, into a fast and interpretable loss function optimisation problem.

This research develops a PINN trained to predict the elastic buckling load of a thin steel plate with a perforation. The purpose of this study is to determine whether a PINN can (i) accurately predict the elastic buckling load of a simply-supported perforated plate subject to uniform compression, and (ii) accurately predict the buckling modeshape of the plate. The objective is to challenge the accuracy of FE analysis, without compromising its benefits.

9.3 Literature Review

Examples of how PINNs have been used in structural engineering are discussed in Section 2.1.4. In addition to this, the next sections frame the decisions made for the PINN design later.

9.3.1 Architecture

To model complicated relationships, deep PINNs consist of more than two hidden layers (a hidden layer is a layer between the input and output layers of a neural network that transforms data, helping the model learn patterns). To determine the optimal number of hidden layers and neurons per layer, many authors employ grid search. In this approach, finite sets of candidate layer counts and neuron counts are specified a priori, and the model is trained for every possible combination. The best-performing architecture is then selected [165]. Guo, et al. (2025)[64] investigate a PINN with a different number of neurons in each hidden layer, finding that more layers increases

the computational cost, whereas fewer layers limits the capacity to capture complex features. Furthermore, Liu, et al. (2025) [66] find that the accuracy of the model does not absolutely increase with the number of hidden layers, because of overfitting. Moreover, [165] and [166] highlight that a semi-automated grid search or random hyperparameter sampling should be used to refine accuracy, particularly for cases with a large number of hyperparameters. Alternatively, [62] combined grid search with Bayesian Optimisation (BO) for hyperparameter (λ) tuning [167, 168]. Hyperparameters are user-specified model settings that govern the structure of a machine learning model or the training process, and are not learned directly from the training data.

At the start of training, network parameters are typically initialised using the Xavier scheme [169], which preserves the variance of activations and gradients across layers, while reducing vanishing or exploding gradients thus improving convergence. During training, parameters are updated via backpropagation, and the loss is repeatedly re-evaluated over the training epochs. A training epoch in a PINN is one complete pass through all training data, including both the collocation points and any supervised data points (e.g. boundary or initial conditions). During each epoch, the network parameters are updated by minimising the total loss, which typically combines data mismatch and physics-based residual terms.

Hyperparameter tuning is required to determine the weighting by which soft constraints are enforced. Many authors find these weights manually, which is laborious and time consuming. However, in recent years, authors have proposed various types of adaptive weight schemes for automatic tuning. Most recently, ReLoBRaLo (Relative Loss Balancing with Random Lookback) was proposed to stabilize training by adaptively reweighting losses using relative improvement rates over time [62]. ReLoBRaLo combines the benefits of (i) SoftAdapt's [170] concept of operating on loss statistics rather than gradient statistics, (ii) GradNorm's [171] concept of calculating progress by dividing the loss at the current iteration by the loss at the previous iteration, (iii) Learning Rate Annealing's [172] concept of updating the scalings using exponential decay with (iv) a 'random lookback' which decides whether to use the loss statistics from the previous step to compute the scalings, or whether to look all the way back to the first epoch [62]. The authors find that ReLoBRaLo

achieves competitive results at lower computational demand than vanilla gradient-based methods, and overcomes numerical instability challenges with Learning Rate Annealing.

To solve any physical condition using PINNs, the network's output must be differentiable with respect to the network's inputs. This is achieved with a smooth activation function, which is applied to the weighted sum of the inputs to a neuron, in addition to a bias factor [173]. Generally, authors consider either a hyperbolic tangent activation function [59, 61, 62, 174, 175], an adaptive inverse tangent function [176], ReLU or 'leaky' ReLU [64–66, 70], or sigmoid [177].

9.3.2 Optimisation

The PINN is trained to satisfy the differential operator, initial condition and boundary conditions using stochastic gradient descent at randomly sampled spatial points. Cuomo, et al. (2022) [178] finds that almost all examples of PINNs in literature use automatic differentiation (AD), and this investigation uses PyTorch AD. Numerical differentiation methods such as finite differences estimate derivatives through small perturbations and therefore introduce truncation and round-off errors. By contrast, AD records a computation graph during the forward pass, and by tracing the graph from roots to leaves, derivatives are evaluated to machine precision by using the chain rule in the backward pass [173, 179]. Unlike symbolic differentiation, AD avoids the expression growth associated with manipulating analytical expressions, making it well suited to complex neural network models. The trade-off is that this can be particularly memory and computationally-intensive, especially for higher order derivatives.

The loss function minimisation process is called 'training'. Most PINNs optimise the loss functions either with Adam minibatch sampling [180], or with limited-memory Broyden-Fletchen-Goldfarb-Shanno (LBFGS) algorithm [181, 182]. LBFGS is a second order quasi-Newton method which tends to converge more precisely for problems with well-defined gradients than first order methods such as Adam, which combines adaptive learning rate and momentum methods. Furthermore, LBFGS tends to be faster [175], better for noisy data [183], and requires fewer iterations than Adam which is slower but more robust and scalable.

9.3.3 Collocation Point Sampling

It is typical to sample collocation points uniformly, however fixed sampling can miss regions with large solution gradients. Wu, et al. (2023)[184] compare six uniform sampling methods and propose two residual-based adaptive sampling methods to improve the accuracy of PINNs. Of (i) equispaced uniform grid, (ii) uniformly random sampling [185], (iii) Latin hypercube sampling [186, 187], (iv) Halton sequence [188], (v) Hammersley sequence [189], and (vi) Sobol sequence [190], the authors find that the low-discrepancy sequences (iv, v, vi) perform better than (i, ii, iii). They also find that residual-based adaptive sampling methods significantly improve the accuracy of PINNs with complicated solutions (such as the Burgers' and multi-scale wave equation) but are not necessary for smooth PDEs (such as the Kirchhoff-Love plate equation) [184]. Bastek & Kochmann (2023) [63] choose Sobol sampling for a faster rate of convergence.

In Nabian, et al. (2021) [71] and Celaya, et al. (2025) [191], the authors develop schemes whereby collocation points are chosen based on the loss or residuals of training data. The authors find that their loss or error-aware sampling approaches improve the efficiency of PINN training and accelerates convergence [71, 191, 192]. [193] develop a residual and gradient-based adaptive sampling, targeting PDEs with solutions exhibiting locally steep gradients. More collocation points are located at regions where the differential is the greatest, and this outperforms the accuracy, efficiency and stability of vanilla sampling methods [193].

On examining PINNs for structural health monitoring of Kirchhoff-Love plates, [61] explain the importance of selecting an appropriate number of collocation points, which affects both precision and computational time. They find that model performance is more stable and consistent with a greater number of interior collocation points, but the optimum number of boundary collocation points depends on the physical constraint [61]. Goraya, et al. (2023) [194] and Leiteritz & Pflüger (2021) [195] demonstrate that a greater number of collocation points improves the accuracy, and reduce the likelihood of trivial solutions, but that convergence rates vary by norm and field.

9.4 Mathematical Background to the Problem

9.4.1 Strong-form

Classical small deflection thin-plate theory adheres to the Kirchhoff-Love thin-plate theory, which assumes that normals to the plate mid-surface remain straight, normal, and inextensible during deformation, such that transverse shear deformation is neglected. The theory is applicable to plates whose thickness is small relative to their in-plane dimensions and is commonly formulated assuming linear elastic material behaviour [27]. Using these assumptions, the differential equation governing the behaviour of thin plates is formulated as the St Venant's Equation:

$$D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = q(x, y) \tag{9.1}$$

Where D is the flexural rigidity, q is the transverse (z-direction) load and w is the z-direction displacement. This can be represented in terms of the Biharmonic operator ∇^4:

$$D\nabla^4 w = q(x, y) \tag{9.2}$$

Timoshenko (1959)[8] showed that when in-plane loads are also present in the middle plane of the plate, Equation 9.2 becomes:

$$D\nabla^4 w = q(x, y) + N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + 2N_{x,y}\frac{\partial^2 w}{\partial x \partial y} \tag{9.3}$$

Where N_x, N_y and N_{xy} are normal forces per unit length in the x, y and xy (shear) directions respectively. This chapter will examine a plate subject only to N_x; no other in-plane or out-of-plane forces are present. Equation 9.3 becomes a linear eigenvalue problem of the form:

$$D\nabla^4 w - N_x\frac{\partial^2 w}{\partial x^2} = 0, \tag{9.4a}$$

$$\mathbf{K}w = N_x\mathbf{G}w \tag{9.4b}$$

Whereby $\mathbf{K}$ is the elastic (material) stiffness matrix, and $\mathbf{G}$ is the geometric (initial-stress) stiffness matrix. Thus, (in addition to the trivial solution of zero deflection), there are an infinite number of nontrivial solutions (i.e., plate buckling modes). this

chapter will examine a simply-supported rectangular plate, whereby the boundary conditions for each edge are:

$$(x = 0; x = a) \begin{cases} w = 0, \\ \frac{\partial^2 w}{\partial x^2} = 0, \end{cases} \tag{9.5a}$$

$$(y = 0; y = b) \begin{cases} w = 0, \\ \frac{\partial^2 w}{\partial y^2} = 0 \end{cases} \tag{9.5b}$$

The deflection surface satisfying all boundary conditions and Equation 9.4 is therefore:

$$w = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \tag{9.6}$$

Where A_{mn} is an amplitude factor. Therefore Equation 9.4 can be mathematically formulated as:

$$\sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \left(\frac{m^4\pi^4}{a^4} + \frac{2m^2n^2\pi^4}{a^2b^2} + \frac{n^4\pi^4}{b^4} - \frac{N_x}{D}\frac{m^2\pi^2}{a^2}\right) \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) = 0 \tag{9.7}$$

The infinite sum of independent functions in Equation 9.7 can only be equal to zero when the coefficient of each term except one is equal to zero. Therefore:

$$A_{mn} \left\{ \pi^4 \left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)^2 - \frac{N_x}{D}\frac{m^2\pi^2}{a^2} \right\} = 0 \tag{9.8}$$

This is satisfied if either $A_{mn} = 0$, or the bracket term is 0. The former case defines the trivial solution, where the plate does not deform for any loading. The nontrivial solution is therefore:

$$N_x = \frac{Da^2\pi^2}{m^2} \left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)^2 \tag{9.9}$$

The critical load is when Equation 9.9 is a minimum, found when $n = 1$ and m is the number of buckling halfwaves in the x-direction, dependant on the ratio a/b. It follows that Equation 9.6 can be re-written as:

$$w = A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \tag{9.10}$$

For the examination of a square plate, we have $a = 1$, $b = 1$, $n = 1$ and $m = 1$. Substituting these into Equation 9.10, Equation 9.4 becomes:

$$\frac{N_x}{D} = 4\pi^2 \tag{9.11}$$

9.4.2 Weak-form

An alternative to solving the differential plate equation is to evaluate the Rayleigh quotient, which provides an upper bound to the critical buckling load. Trial functions must satisfy the Dirichlet boundary conditions, while the Neumann conditions arise naturally from minimising the function, and therefore do not need to be imposed explicitly. It can be shown that [45]:

$$N_x = \min \frac{\Delta U}{\Delta V} \tag{9.12}$$

Where ΔU and ΔV denote the internal strain and external work done on the plate respectively. These are given as [9]:

$$\Delta U = \frac{D}{2} \int_0^a \int_0^b \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right)^2 - 2(1-\nu) \left(\frac{\partial^2 w}{\partial x^2} \cdot \frac{\partial^2 w}{\partial y^2} - \left(\frac{\partial^2 w}{\partial x \partial y} \right)^2 \right) dy\, dx, \tag{9.13a}$$

$$\Delta V = \frac{N_x}{2} \int_0^a \int_0^b \left(\frac{\partial w}{\partial x} \right)^2 dy\, dx \tag{9.13b}$$

9.5 Surrogate Solution to the Problem

The surrogate solution to the problem of an elastically buckling plate is a physics-informed neural network (PINN). As a 'benchmark', the elastic buckling load predicted with the PINN will be compared to that predicted with [26], [23], and a simple feed-forward neural network (FNN).

9.5.1 Feed-Forward Neural Network (FNN)

The FNN receives the 870 cases used to develop the empirical equations in Chapter 4 [23], with a train-test-split of 80%-20%. The validation loss is computed on the test set at the end of every epoch to determine the generalisation error, and used to conduct early stopping with a patience of 10 (i.e., the number of epochs to wait after the last improvement in the validation loss before terminating training); therefore, the FNN should not over generalise on the test set. Early stopping is implemented with the Adam optimiser (Section 9.3.2) which receives training data in batches of 32, and the optimal number of hidden layers and neurons-per-layer is found using grid search. Using 'ReLU' activation in each hidden layer, this investigation finds that a neural network with 4 hidden layers of 128 neurons in the first and last layer, and 64 neurons in the middle two layers results in the lowest validation error.

9.5.2 Physics-Informed Neural Network (PINN)

9.5.2.1 PINN Architecture

As the loss terms may differ greatly in magnitude, the LBFGS optimiser is preferred to Adam, as it tends to converge faster and more precisely for such problems with ill-conditioned loss landscapes (Section 9.3.2). Additionally, LBFGS adapts its own step sizes through line search and curvature approximations thus mitigating learning rate tuning associated with Adam. The scaling factor on the line search step is provided as 0.1, with the number of optimisation steps capped at 1 per epoch.

The PINN weights are initialised with Xavier, and tanh activation is considered for all layers (Section 9.3.1). To begin, a PINN with 4 hidden layers and 35 neurons in each layer is considered, and grid search will be performed later.

9.5.2.2 PINN Losses

The strong-form and weak-form discussed in Section 9.4 can be separately considered to formulate alternative PINNs, assuming either a 'forward' (subscript 'F') or 'inverse' (subscript 'I') total loss function (Table 9.1). This may be composed of the physics loss ($\mathcal{L}_{Phys}$), boundary loss ($\mathcal{L}_{BD}$), and data loss ($\mathcal{L}_{Data}$).

Table 9.1: Four possible PINN variants

Variant	$\mathcal{L}_{Phys}$	$\mathcal{L}_{BD}$	$\mathcal{L}_{D}$
$\mathcal{L}_{Total,F,Weak}$	■		
$\mathcal{L}_{Total,F,Strong}$	■	■	
$\mathcal{L}_{Total,I,Weak}$	■		■
$\mathcal{L}_{Total,I,Strong}$	■	■	■

Where λ are weighting terms (taken to be 1 unless otherwise stated), the total forward loss for the strong-form PINN is given as:

$$\mathcal{L}_{Total,F,Strong} = \lambda_{Phys,Strong}\mathcal{L}_{Phys,Strong} + \lambda_{BD,x=0}\mathcal{L}_{BD,x=0} + \lambda_{BD,x=a}\mathcal{L}_{BD,x=a} + \lambda_{BD,y=0}\mathcal{L}_{BD,y=0} + \lambda_{BD,y=b}\mathcal{L}_{BD,y=b} \quad (9.14)$$

The total forward loss for the weak-form PINN only considers one loss term, as the Neumann boundary conditions are satisfied by default, provided that the predicted out-of-plane displacement $\widehat{w}$ satisfies the Dirichlet conditions (in this case, $\widehat{w} = 0$ is enforced on all boundaries):

$$\mathcal{L}_{Total,F,Weak} = \lambda_{Phys,Weak}\mathcal{L}_{Phys,Weak} \quad (9.15)$$

The total inverse loss includes knowledge of a data loss:

$$\mathcal{L}_{Total,I} = \mathcal{L}_{Total,F} + \lambda_D\mathcal{L}_D \quad (9.16)$$

Where the data loss is given in terms of either the true displacement w, or the true capacity $N_{cr,h}$:

$$\mathcal{L}_D = \frac{1}{T_D}\sum_{k=1}^{T_D}|\widehat{w} - w|^2, \tag{9.17a}$$

$$\mathcal{L}_D = |\widehat{N}_{cr,h} - N_{cr,h}|^2 \tag{9.17b}$$

Unlike the weak-form loss (where the Dirichlet conditions are hard-coded in the forward function of the weak-form neural network definition), the strong-form boundary loss must be softly enforced on the strong-form PINN (Equation 9.14). For each the left ($x = 0$), right ($x = a$), bottom ($y = 0$) and top ($y = b$) edges, the strong-form boundary losses are given as:

$$\mathcal{L}_{BD,x=0} = \frac{1}{T_{x=0}}\sum_{i=1}^{T_{x=0}}|\widehat{w}|^2 + \left|\frac{\partial^2\widehat{w}}{\partial x_i^2}\right|^2, \tag{9.18a}$$

$$\mathcal{L}_{BD,x=a} = \frac{1}{T_{x=a}}\sum_{i=1}^{T_{x=a}}|\widehat{w}|^2 + \left|\frac{\partial^2\widehat{w}}{\partial x_i^2}\right|^2, \tag{9.18b}$$

$$\mathcal{L}_{BD,y=0} = \frac{1}{T_{y=0}}\sum_{i=1}^{T_{y=0}}|\widehat{w}|^2 + \left|\frac{\partial^2\widehat{w}}{\partial y_i^2}\right|^2, \tag{9.18c}$$

$$\mathcal{L}_{BD,y=b} = \frac{1}{T_{y=b}}\sum_{i=1}^{T_{y=b}}|\widehat{w}|^2 + \left|\frac{\partial^2\widehat{w}}{\partial y_i^2}\right|^2 \tag{9.18d}$$

$$\tag{9.18e}$$

Finally, the physics loss is determined with a separate expression for each the strong-form ($\mathcal{L}_{Phys,Strong}$) and weak-form ($\mathcal{L}_{Phys,Weak}$) PINN. $\mathcal{L}_{Phys,Strong}$ is found as a mean least squares error loss of the PDE residual for each (x_i, y_i) pair over the total number of the interior collocation points (T_i):

$$\mathcal{L}_{Phys,Strong} = \frac{1}{T_i}\sum_{i=1}^{T_i}\left|\frac{\partial^4\widehat{w}}{\partial x_i^4} + 2\frac{\partial^4\widehat{w}}{\partial x_i^2\partial y_i^2} + \frac{\partial^4\widehat{w}}{\partial y_i^4} - 4\pi^2\frac{\partial^2\widehat{w}}{\partial x_i^2}\right|^2 \tag{9.19}$$

Note that Equation 9.19 considers a square plate; it embeds the solution given in Equation 9.11. Conversely, there is no PDE residual for the weak-form Rayleigh quotient loss, which behaves like an eigenvalue function:

$$\mathcal{L}_{Phys,Weak} = \frac{\Delta U}{\Delta V} = \frac{\sum_{i=1}^{T_i}\left(\frac{\partial^2\widehat{w}}{\partial x_i^2} + \frac{\partial^2\widehat{w}}{\partial y_i^2}\right)^2 - 2(1-\nu)\left(\frac{\partial^2\widehat{w}}{\partial x_i^2}\cdot\frac{\partial^2\widehat{w}}{\partial y_i^2} - \left(\frac{\partial^2\widehat{w}}{\partial x_i\partial y_i}\right)^2\right)}{\sum_{i=1}^{T_i}\left(\frac{\partial\widehat{w}}{\partial x_i}\right)^2} \tag{9.20}$$

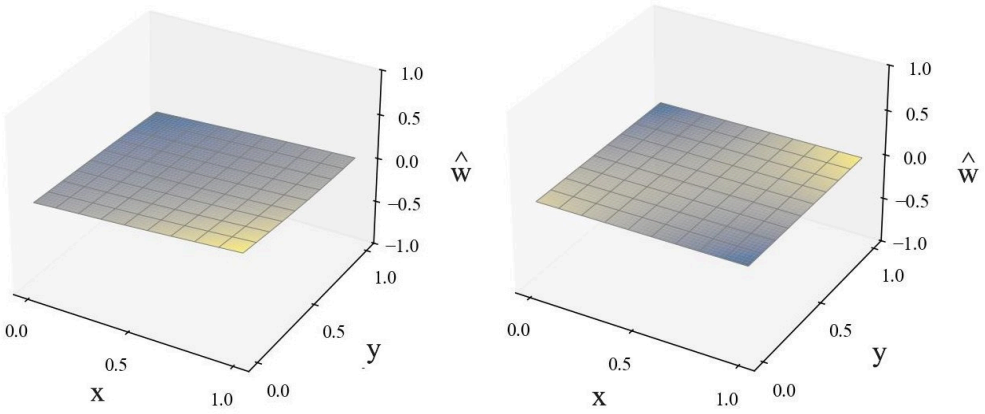
9.6 Unperforated Plate

The first problem is to determine whether a PINN can accurately find the buckling modeshape for a simply-supported square plate with no perforation. The problem is posed for a square plate, where $a = 1$, $b = 1$.

9.6.1 Strong-Form

With the loss terms formulated, a forward PINN is first computed (Equation 9.14, i.e., no data-loss term). $\mathcal{L}_{Phys,Strong}$ (Equation 9.19) is found for $T_i = 8000$ points, and L_{BD} is found for $T_{BD} = 2000$ points on each plate edge. The results are shown for (a) 250 epochs and (b) 2500 epochs, which are computed in 4 minutes and 60 minutes respectively.

The network has no incentive to find a nontrivial shape when informed only by the residual and boundary losses (i.e., the forward-case). Thus the problem is ill-posed for minimisation, and gradient descent collapses to the simplest (trivial) solution for both cases (Figure 9.1).

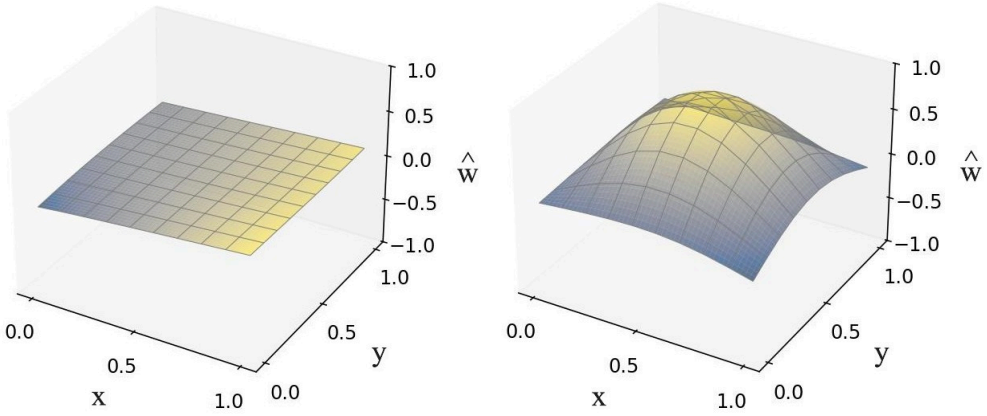


(a) 250 epochs of training **(b)** 2500 epochs of training

Figure 9.1: Strong-form results for forward PINN

To make the problem well-posed, an inverse PINN is now considered, by adding a data loss term ($\mathcal{L}_D$, Equation 9.17(a)) to give Equation 9.16. In this case, the term $\mathcal{L}_D$ is simply taken as unity displacement at the centre of the plate. When there is a great difference between $\widehat{w}$ and w at $x_i = a/2, y_i = b/2$, $\mathcal{L}_D$ is large, thus $\widehat{w}$ is forced away from the trivial case..

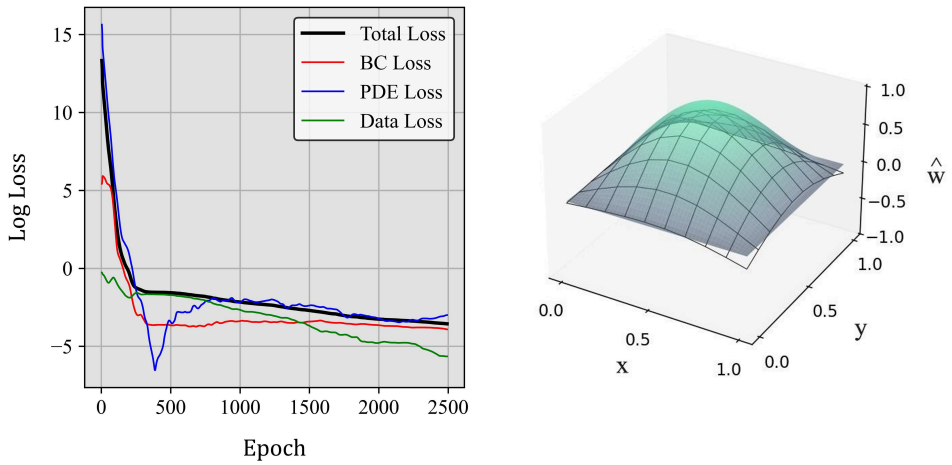
This exact solution is computed at all 8000 interior points and compared to the predicted value after each step. The results are again shown for (a) 250 epochs and (b) 2500 epochs, taking a similar amount of time to compute as before (Figure 9.2).



(a) 250 epochs of training **(b)** 2500 epochs of training

Figure 9.2: Strong-form results for inverse PINN

This time, the modeshape is correctly determined, with a difference in accuracy of only 1.42% to the true modeshape. The log loss is calculated at every epoch (Figure 9.3(a)) and the true modeshape is shown in blue-green in comparison to the predicted shape, shown as the black wireframe (Figure 9.3(b)).



(a) Total loss as the number of training epochs increases

(b) Modeshape predicted by the PINN (wireframe) shown in comparison to the true modeshape (surface)

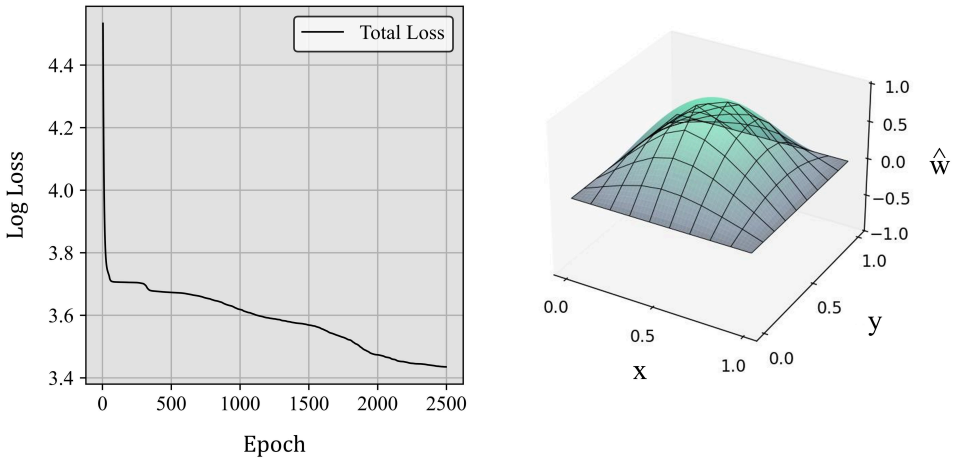
Figure 9.3: Strong-form inverse PINN

Figure 9.3(a) suggests that the total loss would continue to decline if the number of epochs were increased. It is hypothesised that the large drop in $\mathcal{L}_{PDE}$ at around 400 epochs is when the trivial solution is most satisfied (notice that $\mathcal{L}_D$ is quite constant between 250 and 500 epochs). However, after 400 epochs, the PINN learns a non-trivial solution (the first buckling mode); the data loss begins to decline.

9.6.2 Weak-Form

The strong-form PINN is slow because fourth-order differentials must be determined at each of the 8000 collocation points, and $\mathcal{L}_{BD}$ must be computed at each of the 8000 boundary points. Comparatively, the weak-form PINN is significantly faster as it requires nothing beyond the second-order differentials, and the boundary loss is hard-coded on the output. Additionally, the weak-form PINN loss (Equation 9.20) is not at risk of collapsing to the trivial solution, for two reasons. Firstly, the function does not decrease if $\widehat{w}$ is reduced; for $\widehat{w} = c\widehat{w}$, both the numerator and denominator scale by c^2. Secondly, if $\widehat{w} = 0$, the quotient is undefined, thus the zero function is an inadmissible candidate. Essentially the weak-form PINN removes amplitude freedom whereas the strong-form PINN does not.

After 2500 epochs, the forward-case Rayleigh quotient PINN has a difference in accuracy to the true modeshape of 3.01%, after only 5 minutes and 10 seconds (Figure 9.4). The weak-form PINN thus outperforms the strong-form PINN as it is faster, and highly accurate without any data-loss term.



(a) Total loss as the number of training epochs increases

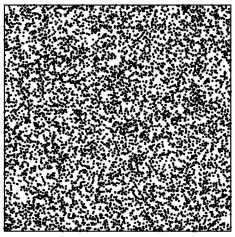
(b) Modeshape predicted by the PINN (wireframe) shown in comparison to the true modeshape (surface)

Figure 9.4: Weak-form forward PINN

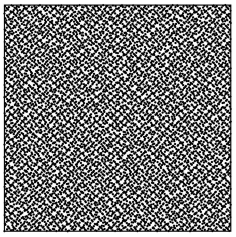
9.6.3 Oversampling

Authors have found that oversampling or importance sampling with different methods can improve the accuracy of a PINN (Section 9.3.3). Sobol sampling can enable a more uniform coverage of the domain, requiring fewer points for a similar accuracy than 'uniform' (Monte-Carlo) sampling considered thus far [184]. This makes Sobol sampling better for capturing more localised physics, such as behaviour near boundaries and high-curvature regions. Furthermore, oversampling (with points concentrated at high-curvature regions where the physics error is large) can outperform the accuracy of uniform coverage [193].

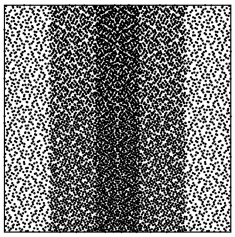
From the findings of [193] and [184], the 8000 collocation points are oversampled into five intervals using Sobol sampling (Figure 9.5(c)). This finds only a 0.132% difference to the true mdoeshape after 250 epochs, outperforming even the strong-form inverse PINN.



(a) Uniform sampling



(b) Sobol sampling



(c) Sobol oversampling into five intervals

Figure 9.5: Sampling collocation points in a square plate with loading on the left and right edges

9.7 Perforated Plate

The weak-form (Rayleigh quotient) formulation will henceforth be considered, and $T_i = 8000$ collocation points will be oversampled into $I_n = 5$ intervals using Sobol sampling.

9.7.1 Mathematical Background

The boundary condition for a free-edge is given by [8];

$$\frac{\partial^3 \widehat{w}}{\partial \iota_i^3} + (2-\nu)\frac{\partial^3 \widehat{w}}{\partial \iota_i \partial \zeta_i^2} = 0, \tag{9.21a}$$

$$\frac{\partial^2 \widehat{w}}{\partial \iota_i^2} + \nu \frac{\partial^2 \widehat{w}}{\partial \zeta_i^2} = 0 \tag{9.21b}$$

Where ι is the direction of an edge, and ζ is the direction of the perpendicular edge. This investigation considers holes with edges parallel to the plate edges, and the global $x-y$ co-ordinate system is therefore used. Using Equation 9.21, the Rayleigh quotient PINN developed in Section 9.6.2 is modified to include $\mathcal{L}_{HB}$ at the hole border. Therefore, the

total loss of the perforated plate is given independently of any data-loss term as:

$$\mathcal{L}_{Total,F,Weak} = \lambda_{Phys,Weak}\mathcal{L}_{Phys,Weak} + \lambda_{HB,x}\mathcal{L}_{HB,x} + \lambda_{HB,y}\mathcal{L}_{HB,y} \tag{9.22}$$

Where the first term is given in Equation 9.20, and the hole border loss is given in Equation 9.21. The location of the hole is excluded from the interior points, and points which lie approximately along the border are identified (Figure 9.7.1).

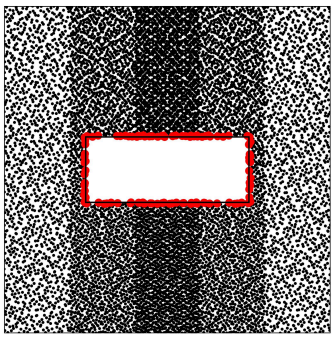


Figure 9.6: Example of Sobol sampling into 5 intervals ($I_n = 5$), with the points along the hole border identified in red

The number of points on each of the plate edges is 2000, and the number of points on each hole border is given as:

$$\left\lfloor \frac{T_i(l_x + l_y)}{54} \right\rfloor \tag{9.23}$$

As the hole diameter increases beyond 40% of the plate width, the stress-redistribution effect dominates [35, 73] (Section 2.2.1). This redistribution behaviour is physically represented by the Föppl-von Kármán equations for a variable stress field, written in terms of the Airy stress function Φ. The nonlinear coupling of membrane stress redistribution and plate bending, is significantly influenced by the presence of a large perforation. Classical analyses by [196] and [8] demonstrate that membrane stresses grow exponentially towards the hole boundary, thereby significantly enhancing bending-membrane interaction and influencing the plate's buckling response. Additionally, [197] found that as the hole diameter increases, there are localised curvature concentrations, and strong rotation gradients around the hole boundary (i.e., transverse shear and rotation-derivative contributions to

the nonlinear terms become non-negligible). In such cases [197] recommend using shear-deformable theories (such as the Reissner-Mindlin plate theory, rather than the Kirchhoff-Love theory) and that the in-plane strain should be given in terms of a geometrically nonlinear rotation term (θ_ζ). For strain in the $\zeta\zeta$ direction, this is expressed as:

$$\epsilon_{\zeta\zeta} = \frac{1}{2}\left(\frac{\partial w}{\partial \zeta}\right)^2 + \frac{1}{2}\left(\frac{\partial \theta_\zeta}{\partial \zeta}\right)^2 \tag{9.24}$$

However, for simplicity, the perforated plate PINN considered herein is formulated with the original Rayleigh quotient loss (Equation 9.20).

9.7.2 PINN with One Input

To examine the performance of the perforated plate PINN, 10 examples are evaluated for 250 epochs. Each case has the same number of uniformly Sobol-sampled interior points ($I_n = 5$), and the same number of boundary points. There are a total of 50 points situated on the hole border, and $\lambda_{Phys,Weak}$ and λ_{HB} are arbitrarily taken as 10 and 0.1 respectively.

Table 9.2 summarises the dimensions assessed, in addition to the difference in accuracy compared to the ABAQUS finite element model ($N_{cr,FE}$), when $\widehat{N}_{cr,h} = N_M$ (the capacity predicted with the MEP in Chapter 3, [26]); when $\widehat{N}_{cr,h} = N_E$ (the capacity predicted with [23]); when $\widehat{N}_{cr,h} = N_F$ (the capacity predicted with the FNN developed in Section 9.5.1); and when $\widehat{N}_{cr,h} = N_P$ (the capacity predicted with the PINN developed in Section 9.6.2).

Two dimensions are examined; a plate of aspect ratio 2 and a plate with aspect ratio 3, both with an unperforated elastic buckling load $N_0 = 7353$ N. The FE model is created with ABAQUS, and exactly follows the linear perturbation model in Section 7.3. Each time the PINN runs, it takes 30 seconds to compute, and it takes ABAQUS (with S4 elements 1mm square) 40 seconds (although this excludes the time taken to mesh and apply material, loading and boundary conditions). In Table 9.2, the plate length (a), plate width (b), central x co-ordinate (c_x), central y co-ordinate (c_y), hole length (l_x) and hole width (l_y) are all given in mm. The accuracy is given by δ; the absolute percentage error of the predicted capacity $\widehat{N}_{cr,h} = N_P, N_E, N_F$ and N_M.

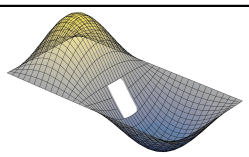
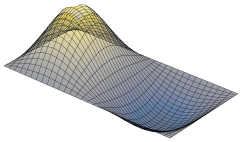
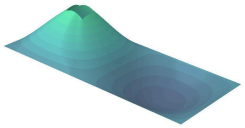
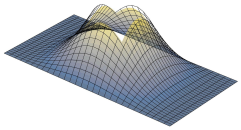
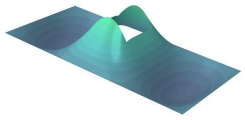
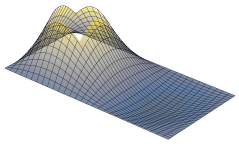
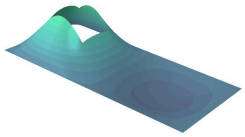
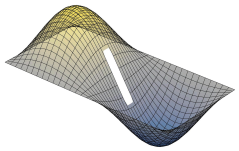
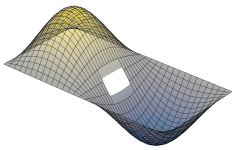
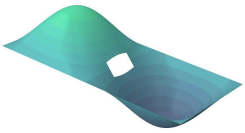
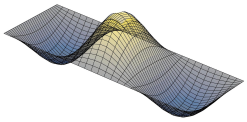
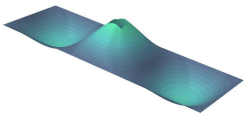
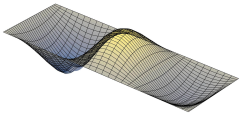
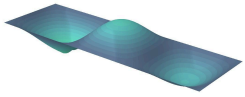
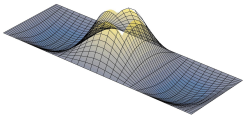
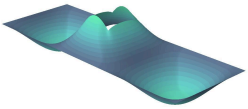
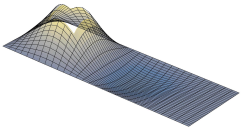
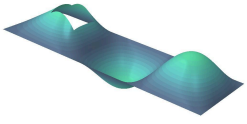
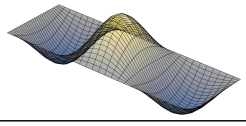
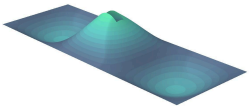
$$\delta = \left|\frac{100(N_{FE} - \widehat{N}_{cr,h})}{N_{FE}}\right| \tag{9.25}$$

Table 9.2: Geometries considered in a study on the PINN with one input, and variables are summarised in Figure 1.4. All dimensions in mm; the measurement origin is at the bottom left corner of the plate

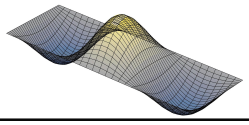
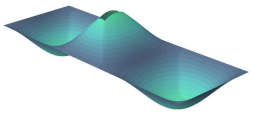
Example	a	b	c_x	c_y	l_x	l_y	δ_P	δ_E	δ_F	δ_M
1	200	100	100	50	20	20	2.69	11.8	4.99	4.53
2	200	100	50	50	20	20	0.09	0.16	11.4	0.55
3	200	100	100	50	40	40	28.0	29.3	13.9	17.21
4	200	100	50	50	40	40	21.7	5.20	0.20	9.15
5	200	100	100	50	30	10	3.76	14.4	3.54	2.03
6	200	100	100	50	10	30	0.60	9.04	8.58	6.06
7	300	100	100	50	20	20	0.56	0.45	12.6	0.76
8	300	100	50	50	20	20	0.72	2.87	7.84	0.70
9	300	100	100	50	40	40	17.7	9.07	0.85	12.6
10	300	100	50	50	40	40	22.1	12.0	12.8	9.61
11	300	100	100	50	30	0.1	5.75	0.52	12.56	3.79
12	300	100	100	50	10	30	2.28	1.70	15.4	3.29

The PINN is more accurate than both the prediction developed by [23] (δ_E) and the benchmark vanilla FNN (Section 9.5.1, δ_F). The median (η) and interquartile range IQR are $\eta_P = 3.23\%$, $IQR_P = 18.0\%$, $\eta_E = 7.12\%$, $IQR_E = 10.5\%$, and $\eta_F = 9.99\%$, $IQR_F = 8.27\%$. However, the MEP developed in [26] (Chapter 3) is most accurate with $\eta_M = 4.16$, $IQR_M = 7.55$. The classification algorithm developed in [23] is quite good at describing the buckling modeshape (Table 9.3); plates with a much larger deformation towards one end of the plate (Test 2, 4, 10) are identified as an 'INT'eraction of buckling modes. However, where this classification can only describe the buckling modeshape, the PINN can predict what this modeshape will look like, which is found very accurately in all examples after only 250 epochs (with the slight exception of Test 10, Table 9.3).

Table 9.3: A comparison of the buckling modeshape found using FE and PINN with one input

Example	Mode Prediction [23]	PINN Result	FE Result
1	WPB		
2	INT		
3	WPB		
4	INT		
5	INT		
6	WPB		
7	WPB		
8	WPB		
9	WPB		
10	INT		
11	WPB		

Continued on next page

Example	Mode Prediction [23]	FE Result	PINN Result
12	WPB		

These results show that a Rayleigh quotient forward-PINN can approximate both the buckling modeshape, and the elastic buckling load with only knowledge of the plate and hole dimensions, with no knowledge of the true result. Not only can the result be found in an equivalent computational time to ABAQUS, but the additional time spent meshing and selecting boundary conditions is avoided with the Rayleigh quotient PINN. Yet although the predicted elastic buckling load is particularly accurate for plates with 'small' holes, the accuracy decreases as the area of the hole increases; the prediction for plates of both dimensions with a 40 x 40 mm hole are very conservative.

Test 3 is the only plate of aspect ratio 2 to deform into 3 halfwaves, thus exhibiting wavelength stiffening (Section 2.2.2), possessing a 16% increase in elastic buckling load when compared to the unperforated plate, attributed to the stress-redistribution effect.

Section 9.7.1 discussed that this behaviour is represented by the Föppl-von Kármán equations, which should be considered in addition to shear-deformable effects. As the energy associated with these phenomena is not included in Equation 9.20, the PINN is unable to accurately predict the elastic buckling load of a test with 'large' holes (Tests 3, 4, 9 and 10).

When these four results are excluded, the PINN performs with the greatest accuracy of the four methods compared, with an average % error (δ) of $\eta_P = 1.50, IQR_P = 2.36$ in comparison to $\eta_E = 2.28, IQR_E = 9.24$; $\eta_F = 9.99, IQR_F = 5.44$ and $\eta_M = 2.66, IQR_M = 3.23$. Therefore, the remainder of this chapter will focus on developing the PINN for perforations less than 40% of the plate width, and a separate study should modify the loss term for plates with larger perforations.

9.7.3 PINN with Four Inputs

In addition to poorly predicting the capacity of plates with 'large' holes, these Rayleigh quotient PINNs are not generalisable and therefore not sufficiently robust (i.e., the PINN trained to find the capacity of Test 1 cannot be used to find the capacity and modeshape of any other test, and vice-versa). This section will examine whether a multi-purpose inverse-PINN can be used to find the elastic buckling load of various perforated plates. This PINN will include a data loss term $\mathcal{L}_D$; the difference between the predicted capacity and the true capacity, known from an FE analysis (i.e., Equation 9.17(b) rather than Equation 9.17(a)).

In Section 9.3.1 different types of Adaptive weight schemes were discussed, and Relative Loss Balancing with Random Lookback (ReLoBRaLo) is chosen for this analysis ReLoBRaLo [62]. To be generalisable, the PINN receives multiple inputs and the loss is found for each input in parallel. An algorithm for how the PINN receives inputs and back-propogates the loss is given as follows:

Algorithm 1 Training procedure of parallel PINN

Generate T_i collocation points in the plate domain; select I_n groups of points with oversampling at the plate centre.
Generate T_ζ points on each of the four boundaries.
Generate a $T_i \times 6$ feature matrix containing the collocation point coordinates (x, y), the hole centre coordinates (c_x, c_y), and the hole dimensions (l_x, l_y).
Repeat for all cases to produce an input tensor of shape (T_d, $T_i^{(d)}$, 6), where T_d is the total number of training cases and $T_i^{(d)}$ denotes the number of collocation points for case d.
Set the model architecture (number of layers and number of neurons per layer).
Initialise the model parameters using Xavier initialisation.
for epoch $i = 1, \ldots, T_{\text{epoch}}$
    Use L-BFGS as the optimiser with learning rate of 0.1 and maximum number of evaluations $T_{\text{eval}} = 50{,}000$.
    for $j = 1, \ldots, T_d$:
        Apply ReLoBRaLo to determine the loss weights λ_{RQ}, λ_{D}, and λ_{HB}.
        Compute the total loss $\mathcal{L}_j$ for case j.
    end for
    Compute the mean total loss:

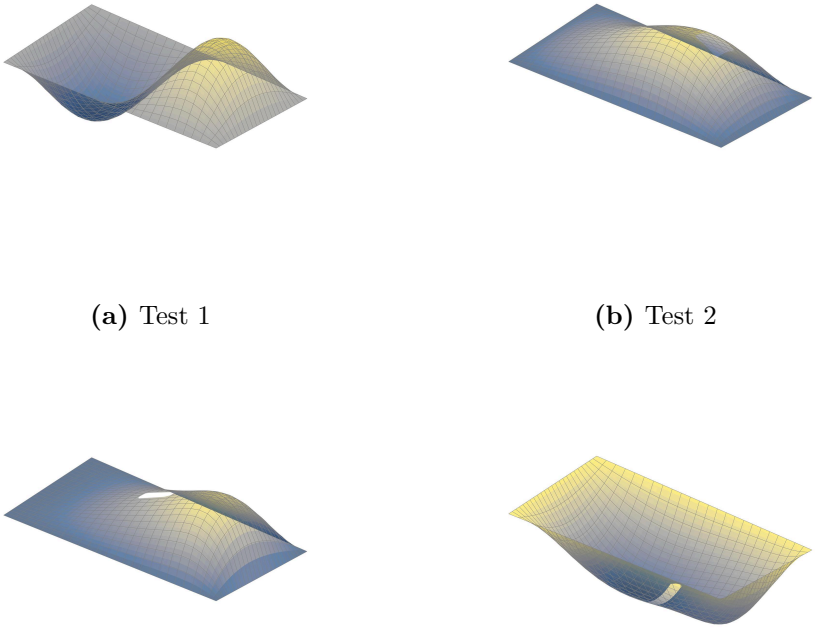
$$\mathcal{L}_{\text{mean}} = \frac{\sum_{j=1}^{T_d} \mathcal{L}_j}{\sum_{j=1}^{T_d} T_i^{(j)}},$$

    Backpropagate mean total loss to update the model parameters.
end for

Algorithm 1 is used to create a generalisable PINN, receiving four inputs (Test 1, 2, 5 and 6) as training data T_d. After 250 epochs, the capacity and modeshape predicted for Test 1, 2, 5 and 6 is given for the PINN trained with only Test 1 as an input, and the PINN with four inputs (Algorithm 1), both of which include $\mathcal{L}_D$ in the total loss for comparability. The results are shown in Table 9.4 and Table 9.5 respectively; as hypothesised, the PINN with four inputs can generalise better over the perforation geometries on which it was trained.

Table 9.4: PINN trained with one input

Test	Predicted capacity (N)	True capacity (N)	Absolute percentage error
Test 1	7740	7680	0.853
Test 2	29400	6760	336
Test 5	23800	7470	205
Test 6	15000	7810	100



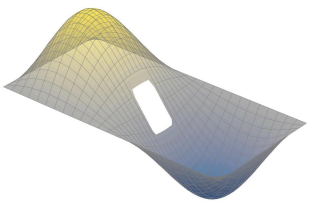
(a) Test 1 **(b)** Test 2

(c) Test 5 **(d)** Test 6

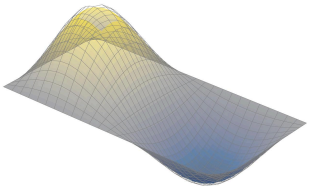
Figure 9.7: Modeshape predicted by PINN trained on one input

Table 9.5: PINN trained with four inputs

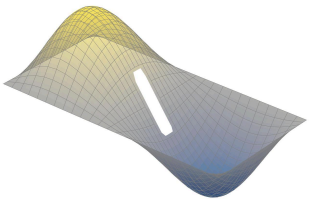
Test	Predicted capacity (N)	True capacity (N)	Absolute percentage error
Test 1	7750	7680	0.906
Test 2	6600	6760	2.30
Test 5	7570	7470	1.29
Test 6	7570	7810	3.09



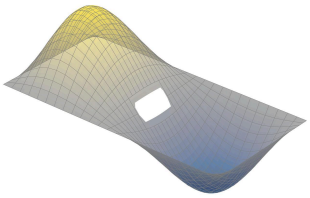
(a) Test 1



(b) Test 2



(c) Test 5



(d) Test 6

Figure 9.8: Modeshape predicted by PINN trained on four inputs

9.7.4 PINN with Many Inputs

The generalisable PINN developed using Algorithm 1 in the previous section is now used to train a parametric PINN for a new plate geometry. This PINN will be trained on many inputs, so that it can accurately predict the modeshape and elastic buckling load for a plate with an aspect ratio of $a/b = 2.2$ and a slenderness of ($b/t = 100$, with a square perforation located anywhere in the plate, even those cases on which it was not trained.

The plate aspect ratio is chosen as it lies closer to the 'transition' aspect ratio for unperforated plates (i.e., the point at which the deformation modeshape will gain a halfwave). Therefore, it is hypothesised that this geometry will be more prone to wavelength stiffening when perforations are located at the central node.

The height of the square perforation is restricted to $l_y \leq 0.3$, as it is known that the capacity of plates with perforations where $l_y \geq 0.4$ cannot be predicted with the simple weak-form loss function considered here (Equation 9.20).

The total number of interior collocation points $T_i = 8000$ and oversampling is performed with $I_n = 5$, with the greatest number of points in the centre of the plate (see Figure 9.7.1). The number of points on the plate and hole edges are defined in Section 9.7.1.

9.7.4.1 Grid Search

Using 24 inputs, grid search is first performed to optimise the number of hidden layers and number of neurons in each layer, and then the optimal number of epochs is selected. For simplicity, each layer has the same number of neurons.

The data used to train the PINN is given in Table 9.6, and data used to validate the PINN (i.e., data it has not seen before) is given in Table 9.7. Note that the PINN is trained for a hole located in the first quadrant of the plate (i.e., $c_x \leq a/2$ and $c_y \leq b/2$). A perforation located anywhere in the plate will always fall within this domain, as a plate with a single perforation has a mirror symmetry of order 4. Bear this in mind when comparing images of the deflected shape, shown later.

Table 9.6: Data used to train the many-input PINN (variables summarised in Figure 1.4)

Training Data Name	c_x/a	c_y/b	l_x/b	l_y/b	$N_{cr,FE}$
1	0.5	0.50	0.05	0.05	7390
2	0.25	0.50	0.05	0.05	7330
3	0.25	0.25	0.05	0.05	7350
4	0.5	0.25	0.05	0.05	7370
5	0.5	0.50	0.10	0.10	7470
6	0.25	0.50	0.10	0.10	7340
7	0.25	0.25	0.10	0.10	7250
8	0.5	0.25	0.10	0.10	7220
9	0.5	0.50	0.15	0.15	7550
10	0.25	0.50	0.15	0.15	7030
11	0.25	0.25	0.15	0.15	7070
12	0.5	0.25	0.15	0.15	7280
13	0.5	0.50	0.20	0.20	7650
14	0.25	0.50	0.20	0.20	6880
15	0.25	0.25	0.20	0.20	6820
16	0.5	0.25	0.20	0.20	7190
17	0.5	0.50	0.25	0.25	7450
18	0.25	0.25	0.25	0.25	7320
19	0.5	0.50	0.25	0.25	7490
20	0.25	0.25	0.25	0.25	7290
21	0.5	0.50	0.30	0.30	7410
22	0.25	0.25	0.30	0.30	7240
23	0.5	0.50	0.30	0.30	7300
24	0.25	0.25	0.30	0.30	7310

Table 9.7: Data used to validate the many-input PINN (variables summarised in Figure 1.4)

Validation Data Name	c_x/a	c_y/b	l_x/b	l_y/b	$N_{cr,FE}$
1	0.20	0.40	0.05	0.05	7340
2	0.35	0.20	0.05	0.05	7360
3	0.25	0.60	0.05	0.05	7340
4	0.30	0.30	0.05	0.05	7350
5	0.20	0.40	0.1	0.1	7220
6	0.35	0.20	0.1	0.1	7270
7	0.45	0.20	0.1	0.1	7300
8	0.25	0.60	0.1	0.1	7220
9	0.30	0.30	0.1	0.1	7260
10	0.20	0.40	0.15	0.15	6990
11	0.35	0.20	0.15	0.15	7120
12	0.45	0.20	0.15	0.15	7180
13	0.25	0.60	0.15	0.15	7050
14	0.30	0.30	0.15	0.15	7120
15	0.20	0.40	0.25	0.25	6480
16	0.35	0.20	0.25	0.25	6450
17	0.45	0.20	0.25	0.25	6680
18	0.25	0.60	0.25	0.25	6710
19	0.30	0.30	0.25	0.25	6740
20	0.20	0.40	0.30	0.30	6246
21	0.35	0.20	0.30	0.30	5830
22	0.45	0.20	0.30	0.30	6110
23	0.25	0.60	0.30	0.30	6570
24	0.30	0.30	0.30	0.30	6481

Grid search was performed by training a PINN with $T_{neurons}$ per hidden layer, and T_{layer} hidden layers for 500 epochs. The results are summarised in Table 9.8, with the median absolute percentage error given for each the training set (Table 9.6) and the validation set (Table 9.7). The Normalised Accuracy Score (V) is given in terms of the

median η and inter-quartile range IQR:

$$V = \sqrt{\eta_{train}^2 + \eta_{val}^2 + IQR_{train}^2 + IQR_{val}^2} \tag{9.26}$$

Table 9.8: Grid search results for PINNs with T_{layers} and $T_{neurons}$

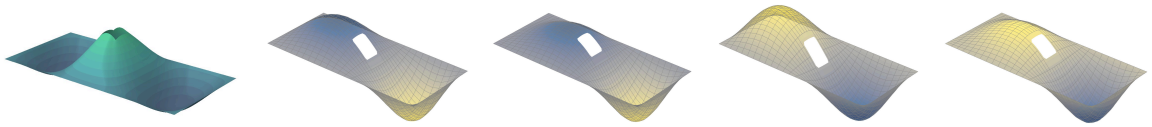
T_{layers}	$T_{neurons}$	Median (η) abs percentage error		Inter-Quartile Range (IQR)		Normalised Accuracy Score (V)
		Training Set	Validation Set	Training Set	Validation Set	
2	64	2.59	1.78	2.86	5.31	6.57
2	70	2.42	2.27	2.87	9.27	9.71
2	80	1.84	1.50	3.46	5.68	6.22
2	90	2.39	2.73	2.61	6.33	7.39
2	100	2.10	3.94	2.69	16.0	17.76
2	128	2.49	2.30	2.78	7.02	7.23
2	150	2.40	1.94	2.69	5.00	6.39
3	64	2.54	2.39	2.91	6.64	6.61
3	70	2.70	7.21	3.31	12.2	13.5
3	80	2.43	2.14	3.19	12.4	11.7
3	90	2.74	3.91	3.17	7.44	9.14
3	100	2.49	2.20	3.09	4.94	6.09
3	128	2.85	2.44	3.18	4.85	8.18
3	150	2.35	7.70	2.55	17.8	17.1
4	64	2.50	1.81	2.84	11.5	9.92
4	70	2.64	2.20	2.80	9.11	10.2
4	80	2.27	2.10	2.89	9.64	8.83
4	90	3.13	2.40	3.17	10.4	10.9
4	100	150	150	31.8	25.1	220
4	128	2.64	6.99	3.12	26.4	24.1
4	150	2.62	2.50	2.25	3.53	5.10

9.7.4.1.1 Quantitative Analysis There is no particular pattern to how T_{layers} and $T_{neurons}$ influence the Normalised Accuracy Score V; for example $T_{neurons} = 150$ is optimal when $T_{layers} = 4$, but least optimal when $T_{layers} = 3$. The inverse is true when $T_{neurons} = 100$; the losses explode when $T_{layers} = 4$.

The four top performing PINNs after 500 epochs are (1) $T_{layers} = 4$ and $T_{neurons} = 150$ with $V = 5.10$ (the largest network investigated), (2) $T_{layers} = 3$ and $T_{neurons} = 100$ with $V = 6.09$, (3) $T_{layers} = 2$ and $T_{neurons} = 80$ with $V = 6.22$, and (4) $T_{layers} = 2$ and $T_{neurons} = 150$ with $V = 6.39$. For plates with $l_x = l_y \geq 0.25b$, the largest network investigated also has the least absolute percentage error between $N_{cr,FE}$ and $N_{cr,PINN}$, where the median values are $\delta_1 = 4.90$, $\delta_2 = 5.42$, $\delta_3 = 7.88$ and $\delta_4 = 5.15$.

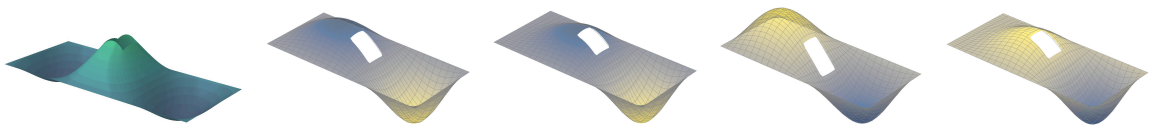
9.7.4.1.2 Qualitative Analysis To ensure that the modeshape is accurately predicted, the output of each PINN must also be qualitatively compared to the FE result. PINNs (1) and (3) do not identify the modeshape with three buckling halfwaves for the training set tests 13, 17, 21 and 22 after 500 epochs, and therefore always predict a modeshape with two buckling halfwaves for all of the validation data.

On the other hand, PINN (2) and PINN (4) can somewhat identify this modeshape change, resulting in a more accurate prediction for the elastic buckling load for tests 13, 17, 21 and 22 (Figure 9.9 - Figure 9.12).



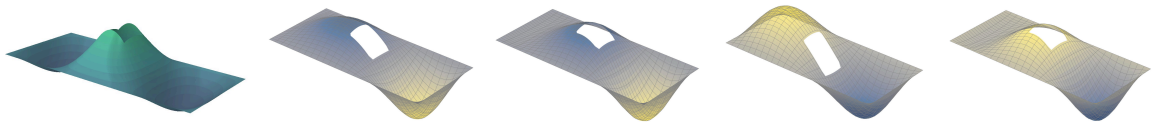
(a) ABAQUS **(b)** $\delta_1 = 1.34\%$ **(c)** $\delta_2 = 0.605\%$ **(d)** $\delta_3 = 3.14\%$ **(e)** $\delta_4 = 0.987\%$

Figure 9.9: Training after 500 epochs - Test 13 (Table 9.6)



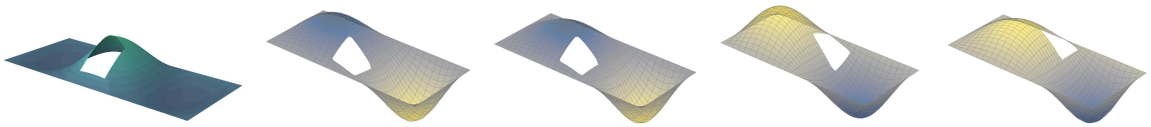
(a) ABAQUS **(b)** $\delta_1 = 3.83\%$ **(c)** $\delta_2 = 1.34\%$ **(d)** $\delta_3 = 8.74\%$ **(e)** $\delta_4 = 2.00\%$

Figure 9.10: Training after 500 epochs - Test 17 (Table 9.6)



(a) ABAQUS **(b)** $\delta_1 = 3.90\%$ **(c)** $\delta_2 = 0.882\%$ **(d)** $\delta_3 = 14.1\%$ **(e)** $\delta_4 = 0.365\%$

Figure 9.11: Training after 500 epochs - Test 21 (Table 9.6)



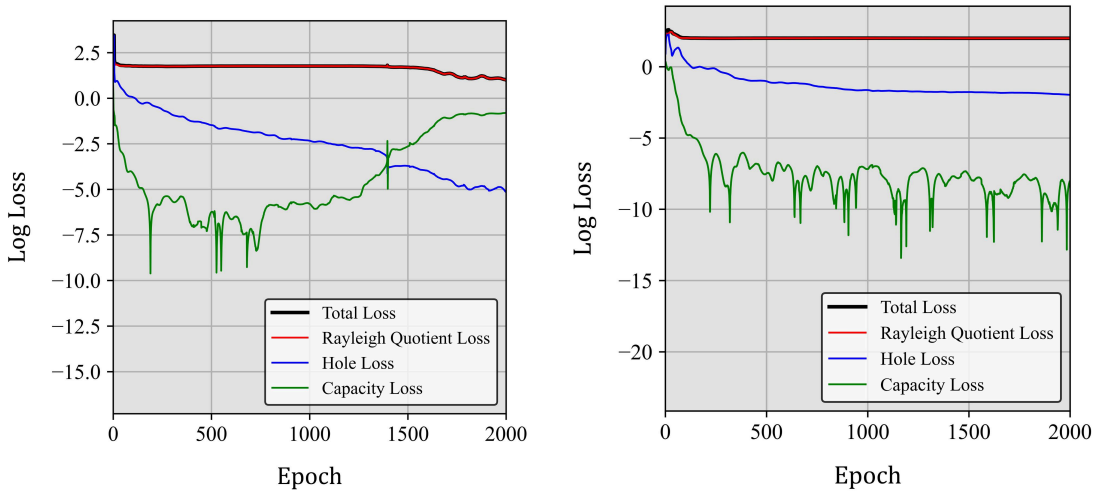
(a) ABAQUS **(b)** $\delta_1 = 12.0\%$ **(c)** $\delta_2 = 11.9\%$ **(d)** $\delta_3 = 13.5\%$ **(e)** $\delta_4 = 11.7\%$

Figure 9.12: Training after 500 epochs - Test 22 (Table 9.6)

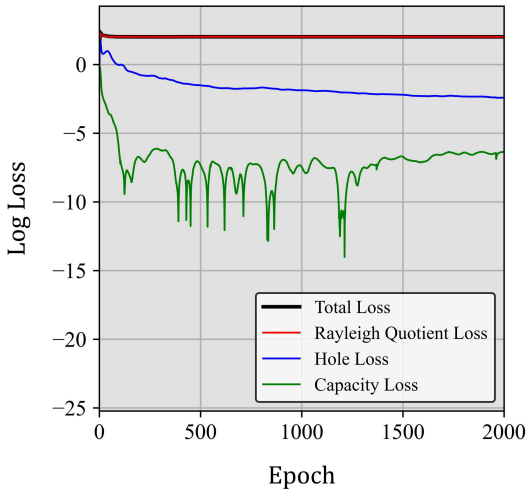
As the elastic buckling load is poorly predicted for $l_x = l_y \geq 0.25b$, and always predicts exactly the same modeshape, PINN(3) is dismissed and only PINN(1),(2) and (4) shall be henceforth compared.

9.7.4.1.3 Extended Training To investigate whether the correct modeshape can be identified after prolonged training, (1), (2) and (4) are trained for a total of 2000 epochs. The log loss - epoch relationship is shown in Figure 9.13. After 2000 epochs, the Normalised Accuracy Score V is 30.6, 19.2 and 30.5 for PINN(1) ($T_{layers} = 4$, $T_{neurons} = 150$), PINN(2) ($T_{layers} = 3$, $T_{neurons} = 100$), and PINN(4) ($T_{layers} = 2$, $T_{neurons} = 150$), respectively.

The modeshape predicted with PINN(1) is significantly less smooth, even for small holes which were previously well predicted; consequently $IQR_{1,train} = 28.6$ compared to $IQR_{2,train} = 3.12$ and $IQR_{4,train} = 2.78$. This can be attributed to the sudden rise in $\mathcal{L}_D$ at the compromise of a reduced $\mathcal{L}_{Phys,Weak}$, not seen in the training results of the other two PINNs (Figure 9.13). Additionally, although PINN(4) can accurately identify the modeshape of the training data, it has the least accuracy on the validation set, suggesting overfitting ($\eta_{1,train} = 2.450$, $\eta_{2,train} = 3.30$, and $\eta_{4,train} = 3.05$ compared to $\eta_{1,val} = 5.998$, $\eta_{2,val} = 7.36$, and $\eta_{4,val} = 9.72$).



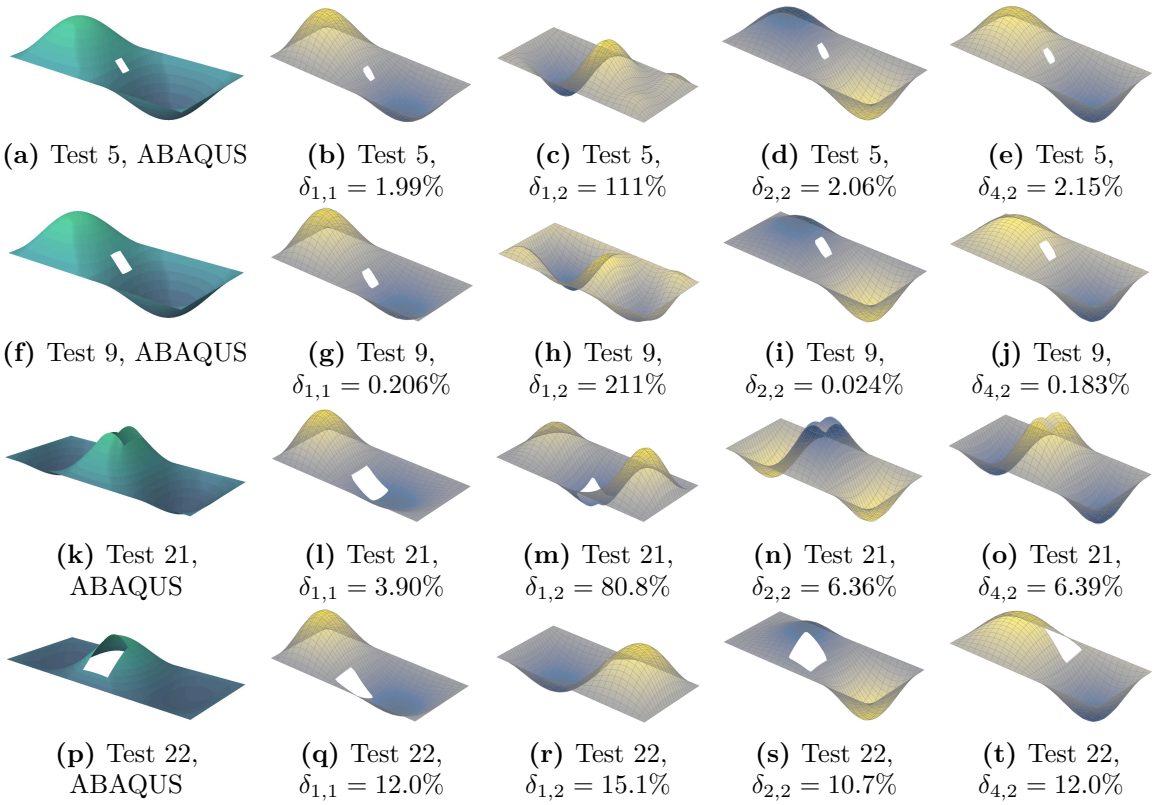
(a) PINN(1) with $T_{layers} = 4$, $T_{neurons} = 150$ **(b)** PINN(2) with $T_{layers} = 3$, $T_{neurons} = 100$



(c) PINN(4) with $T_{layers} = 2$, $T_{neurons} = 150$

Figure 9.13: Log loss as the number of training epochs increases

For selected examples, the modeshape predicted by each of the three PINNs trained for 2000 epochs are shown for the training set (Figure 9.14), and the validation set (Figure 9.15, Figure 9.16). The modeshape predicted with PINN(1) trained for 500 epochs is also shown. The percentage error with the FE-predicted capacity is given by $\delta_{i,j}$ where i denotes the PINN ($i = 1, 2, 4$) and j denotes the number of training epochs ($j = 1$ for 500 and $j = 2$ for 2000).



(a) Test 5, ABAQUS **(b)** Test 5, $\delta_{1,1} = 1.99\%$ **(c)** Test 5, $\delta_{1,2} = 111\%$ **(d)** Test 5, $\delta_{2,2} = 2.06\%$ **(e)** Test 5, $\delta_{4,2} = 2.15\%$

(f) Test 9, ABAQUS **(g)** Test 9, $\delta_{1,1} = 0.206\%$ **(h)** Test 9, $\delta_{1,2} = 211\%$ **(i)** Test 9, $\delta_{2,2} = 0.024\%$ **(j)** Test 9, $\delta_{4,2} = 0.183\%$

(k) Test 21, ABAQUS **(l)** Test 21, $\delta_{1,1} = 3.90\%$ **(m)** Test 21, $\delta_{1,2} = 80.8\%$ **(n)** Test 21, $\delta_{2,2} = 6.36\%$ **(o)** Test 21, $\delta_{4,2} = 6.39\%$

(p) Test 22, ABAQUS **(q)** Test 22, $\delta_{1,1} = 12.0\%$ **(r)** Test 22, $\delta_{1,2} = 15.1\%$ **(s)** Test 22, $\delta_{2,2} = 10.7\%$ **(t)** Test 22, $\delta_{4,2} = 12.0\%$

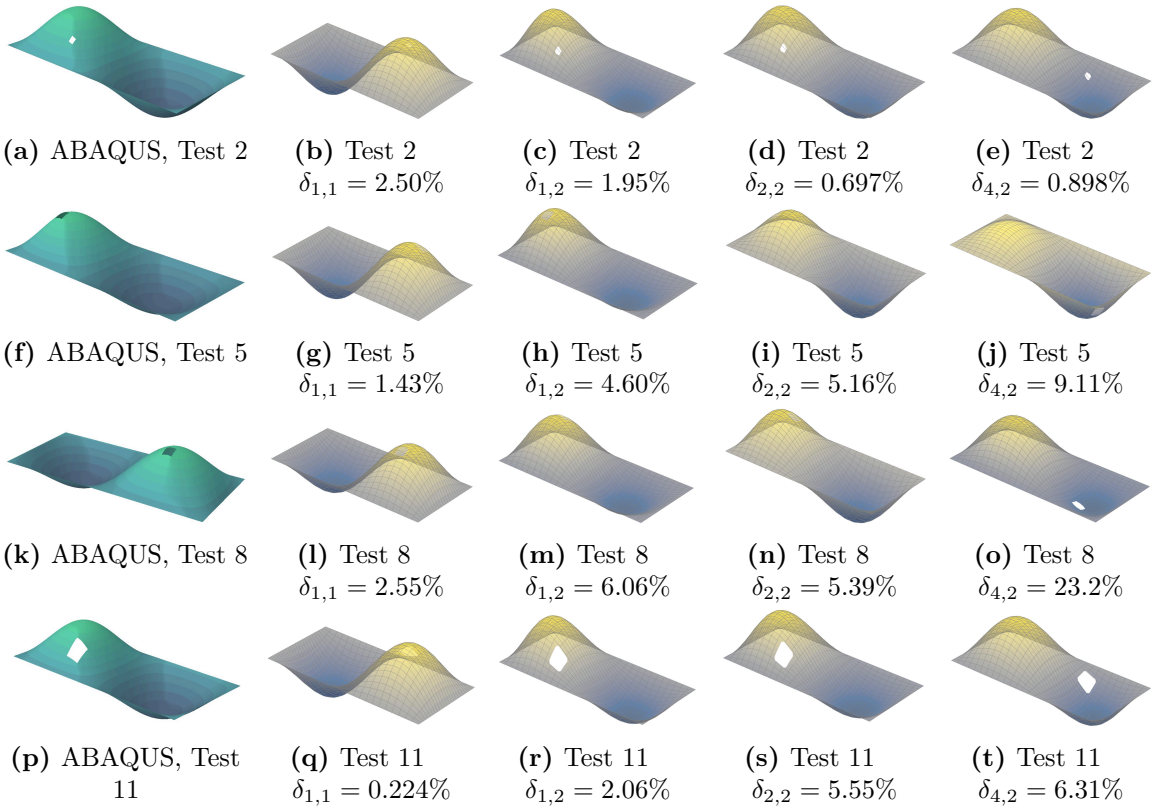
Figure 9.14: Predictions for the modeshape on examples from the training dataset (Table 9.6). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$)

Despite having a large error on the training set (Figure 9.14), PINN(1) trained for 2000 epochs generalises very well, with the best modeshape prediction of all extended training PINNs when used on for all cases in the validation set, even those with larger holes. However, the modeshape predicted using PINN(1) trained for 2000 epochs is not superior to when it is trained for 500 epochs, and it is less accurate ($\eta_{1,1} = 2.47$ and $\eta_{1,2} = 6.00$ after 500 and 2000 epochs respectively). For this reason, PINN(1) trained for 500 epochs is chosen as the final architecture. As this PINN cannot identify a modeshape change, the perforation size and location should comply with the training domain, and exclude the domain where wavelength stiffening is believed to occur (i.e., at the plate centre for a plate with the geometry examined here):

$$c_x = \begin{cases} x \in (\frac{a}{4}, \frac{a}{2}), & \text{if } l_y < 0.2, \\ x \in (\frac{a}{4}, \frac{a}{2}) \setminus \left[\frac{a}{2} - l_x,\ \frac{a}{2} + l_x\right], & \text{if } 0.2 \leq l_y \leq 0.3. \end{cases} \tag{9.27}$$

When comparing the modeshapes in Figures 9.15 and 9.16, please acknowledge that

they are rotationally symmetric and thus equivalent (each PINN received the same data, despite the results being displayed differently).



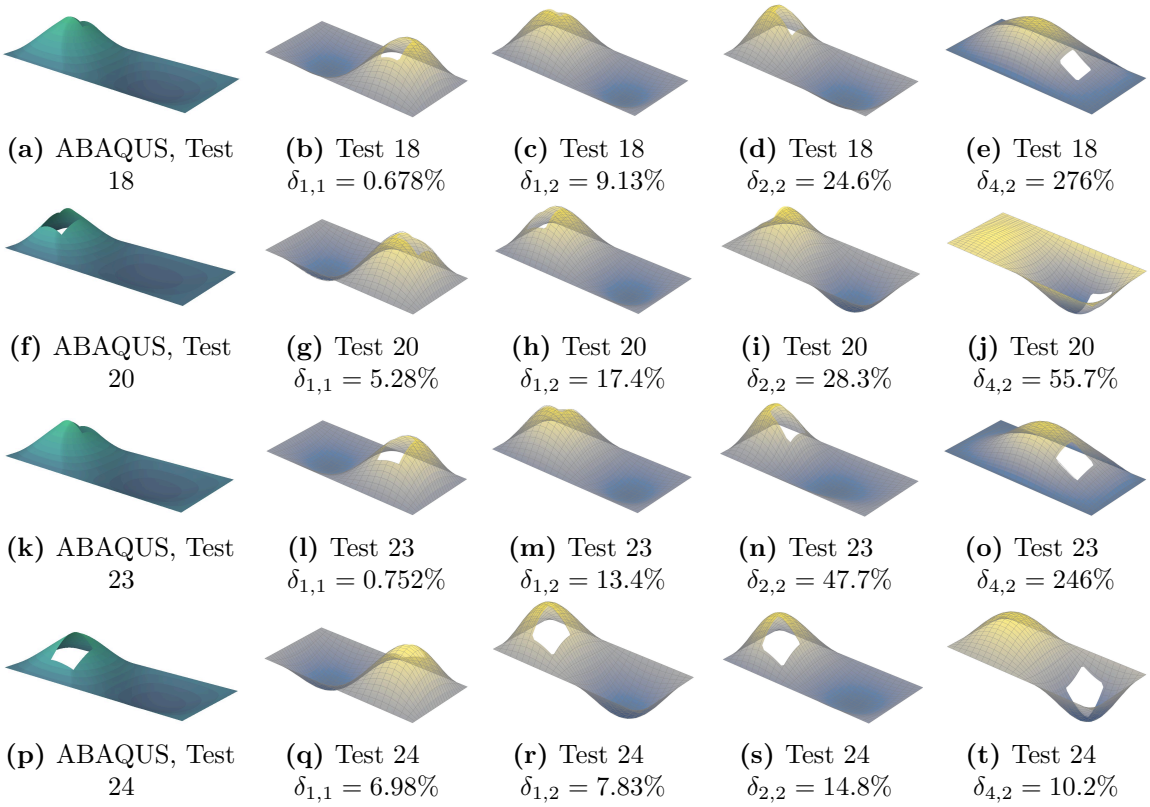
(a) ABAQUS, Test 2 **(b)** Test 2 $\delta_{1,1} = 2.50\%$ **(c)** Test 2 $\delta_{1,2} = 1.95\%$ **(d)** Test 2 $\delta_{2,2} = 0.697\%$ **(e)** Test 2 $\delta_{4,2} = 0.898\%$

(f) ABAQUS, Test 5 **(g)** Test 5 $\delta_{1,1} = 1.43\%$ **(h)** Test 5 $\delta_{1,2} = 4.60\%$ **(i)** Test 5 $\delta_{2,2} = 5.16\%$ **(j)** Test 5 $\delta_{4,2} = 9.11\%$

(k) ABAQUS, Test 8 **(l)** Test 8 $\delta_{1,1} = 2.55\%$ **(m)** Test 8 $\delta_{1,2} = 6.06\%$ **(n)** Test 8 $\delta_{2,2} = 5.39\%$ **(o)** Test 8 $\delta_{4,2} = 23.2\%$

(p) ABAQUS, Test 11 **(q)** Test 11 $\delta_{1,1} = 0.224\%$ **(r)** Test 11 $\delta_{1,2} = 2.06\%$ **(s)** Test 11 $\delta_{2,2} = 5.55\%$ **(t)** Test 11 $\delta_{4,2} = 6.31\%$

Figure 9.15: Predictions for the modeshape on examples from the validation dataset with smaller holes (Table 9.7). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$)



(a) ABAQUS, Test 18 | **(b)** Test 18 $\delta_{1,1} = 0.678\%$ | **(c)** Test 18 $\delta_{1,2} = 9.13\%$ | **(d)** Test 18 $\delta_{2,2} = 24.6\%$ | **(e)** Test 18 $\delta_{4,2} = 276\%$

(f) ABAQUS, Test 20 | **(g)** Test 20 $\delta_{1,1} = 5.28\%$ | **(h)** Test 20 $\delta_{1,2} = 17.4\%$ | **(i)** Test 20 $\delta_{2,2} = 28.3\%$ | **(j)** Test 20 $\delta_{4,2} = 55.7\%$

(k) ABAQUS, Test 23 | **(l)** Test 23 $\delta_{1,1} = 0.752\%$ | **(m)** Test 23 $\delta_{1,2} = 13.4\%$ | **(n)** Test 23 $\delta_{2,2} = 47.7\%$ | **(o)** Test 23 $\delta_{4,2} = 246\%$

(p) ABAQUS, Test 24 | **(q)** Test 24 $\delta_{1,1} = 6.98\%$ | **(r)** Test 24 $\delta_{1,2} = 7.83\%$ | **(s)** Test 24 $\delta_{2,2} = 14.8\%$ | **(t)** Test 24 $\delta_{4,2} = 10.2\%$

Figure 9.16: Predictions for the modeshape on examples from the validation dataset with larger holes (Table 9.7). Training after 500 epochs ($\delta_{i,1}$) and 2000 epochs ($\delta_{i,2}$)

9.7.4.2 Testing

Finally, PINN(1) with $T_{layers} = 4$, $T_{neurons} = 150$ and trained for 150 epochs is applied to two new test sets summarised in Tables 9.9 (Test Set 1) and 9.10 (Test Set 2), shown in increasing percentage error (δ). It is hypothesised that Test Set 1 will be well predicted, since the size of perforations is within the training domain, although the location of perforations may be located slightly outside (closer to the hole edge in some cases).

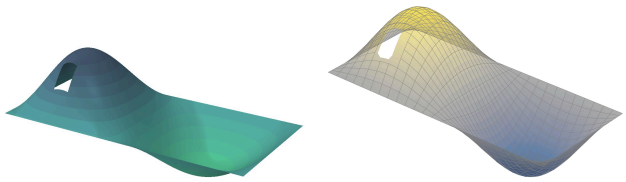
The data in Test Set 2 is hypothesised to be predicted poorly, as perforations lie far outside either or both of the training domain and the wavelength stiffening limit imposed by Equation 9.27. The percentage error between the elastic buckling load found with FE, and the capacity found with the PINN (δ_P) and the prediction developed by [23] (δ_E) is given with each set of results.

Of the models with perforation sizes within the domain on which the PINN was trained (Table 9.9), two of the least accurate predictions are plates with perforations located outside of the domain in both the $x-$ and $y-$ directions (Test 18 and 19, Table 9.9). Although also located outside, the perforations in Tests 6 and 17 are closer to the training domain, and as the perforation in Test 6 is so small, its accuracy is not so greatly affected. Test 20 has the largest error, as the location of the large perforation is very close to the restricted domain (Equation 9.27). Comparatively, the location of the large perforation in Test 16 is well within the training domain, and although the perforation in Test 15 is not, it is much smaller. This is why these tests perform more accurately than Tests 17-20.

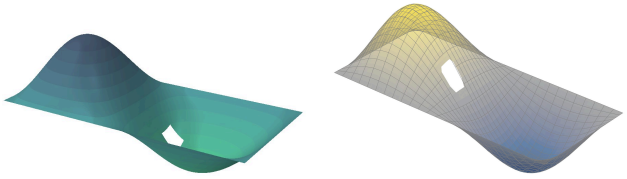
All of these cases exhibit two buckling halfwaves in both the FE and the PINN prediction, and the modeshape of even the least accurate predictions in this subset is well-predicted (Figure 9.18). Overall, the elastic buckling load is predicted very accurately accross all 20 cases, with a median absolute percentage error δ_P of $\eta_{P,test,1} = 1.70$ and $IQR_{P,test,1} = 2.76$, superior to the median absolute percentage error δ_M of $\eta_{M,test,1} = 3.73$ and $IQR_{M,test,1} = 5.36$ and δ_E of $\eta_{E,test,1} = 3.23$ and $IQR_{E,test,1} = 8.09$.

Table 9.9: Test set of models more similar to the training domain (Test Set 1) (variables summarised in Figure 1.4)

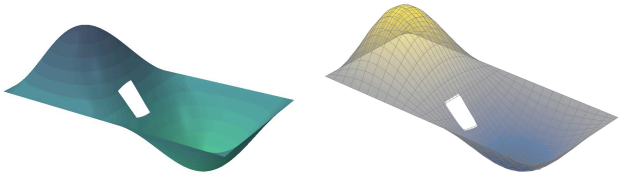
Test Data Name	c_x/a	c_y/b	l_x/b	l_y/b	δ_P	δ_M	δ_E	Mode Predicted by [23]
1	0.22	0.22	0.14	0.14	0.069	4.40	3.54	WPB
2	0.43	0.39	0.13	0.13	0.233	0.485	0.496	WPB
3	0.50	0.40	0.17	0.17	0.309	1.61	2.44	WPB
4	0.26	0.33	0.27	0.27	0.490	12.8	9.71	INT
5	0.14	0.35	0.05	0.05	0.636	0.853	0.872	WPB
6	0.24	0.10	0.05	0.05	0.783	0.877	15.3	WPB
7	0.34	0.43	0.18	0.18	0.932	3.67	3.22	INT
8	0.24	0.35	0.30	0.30	0.965	16.2	13.7	INT
9	0.37	0.41	0.17	0.17	1.41	2.60	1.90	WPB
10	0.24	0.35	0.15	0.15	1.57	5.31	3.27	WPB
11	0.26	0.30	0.13	0.13	1.83	3.79	0.078	WPB
12	0.34	0.45	0.11	0.11	1.92	2.09	1.88	WPB
13	0.24	0.35	0.05	0.05	2.41	1.06	0.675	WPB
14	0.26	0.49	0.06	0.06	3.11	1.31	1.30	WPB
15	0.14	0.35	0.15	0.15	3.24	5.09	3.06	WPB
16	0.26	0.50	0.28	0.28	4.32	9.61	9.58	INT
17	0.24	0.10	0.15	0.15	5.59	7.87	10.2	WPB
18	0.14	0.10	0.05	0.05	6.96	0.795	15.4	WPB
19	0.14	0.10	0.15	0.15	9.15	6.19	11.6	WPB
20	0.37	0.25	0.26	0.26	9.47	10.7	3.77	WPB



(a) ABAQUS, Test 1 **(b)** PINN, Test 1

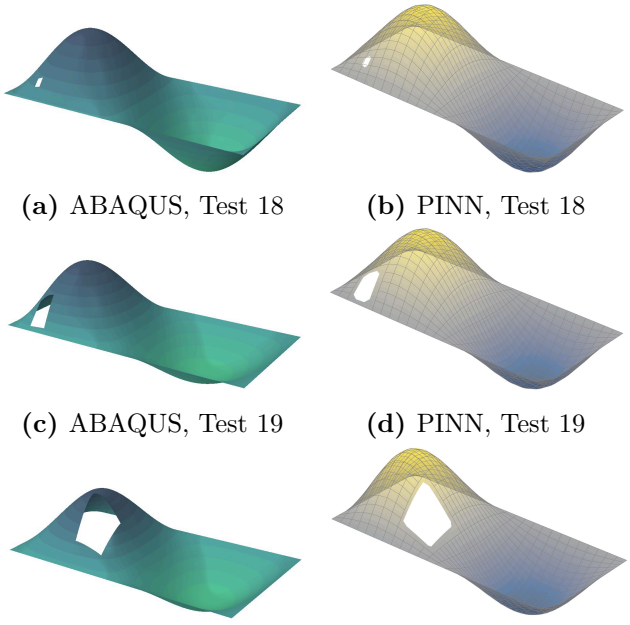


(c) ABAQUS, Test 2 **(d)** PINN, Test 2



(e) ABAQUS, Test 3 **(f)** PINN, Test 3

Figure 9.17: The modeshape of the three best predictions in Test Set 1 (Table 9.9)



(a) ABAQUS, Test 18 **(b)** PINN, Test 18

(c) ABAQUS, Test 19 **(d)** PINN, Test 19

(e) ABAQUS, Test 20 **(f)** PINN, Test 20

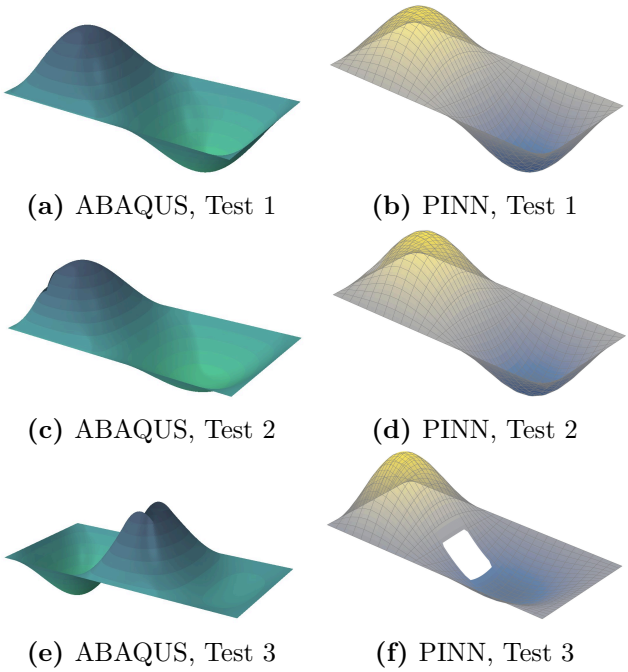
Figure 9.18: The modeshape of the three worst predictions in Test Set 1 (Table 9.9)

9. A Physics Informed Neural Network for the elastic buckling load of a Perforated Plate

For the set of models whose perforations lie outside of the reliable domain, the elastic buckling load of plates with the smallest holes are typically predicted most accurately (Tests 1, 2, 4). However, if the perforation encroaches too close to the plate edge, the accuracy drops greatly, regardless of hole size (Test 7, 8, 17 and 18). Holes larger than those on which the PINN was trained increase the percentage error, particularly if the location exceeds the training domain in both dimensions, or extremely in one dimension (Tests 10, 13, 15 and 20). A significant modeshape change is seen for Test 19 (Figure 9.20), which is not predicted by the PINN, and incidentally results in the second greatest prediction error. However, although the modeshape change for Test 3 is not identified by the PINN, it has the third smallest percentage error. As expected, Test Set 2 (Table 9.10) performs much more poorly than Test Set 1 (Table 9.9), with $\eta_{P,test,2} = 11.7$ and $IQR_{P,test,2} = 13.2$, slightly worse than the median absolute percentage error δ_N of $\eta_{E,test,2} = 8.23$ and $IQR_{E,test,2} = 13.0$. However, the MEP developed in [26] performs much better with $\eta_{M,test,2} = 4.40$ and $IQR_{M,test,2} = 12.6$.

Table 9.10: Test set of models less similar to the training domain (Test Set 2) (variables summarised in Figure 1.4)

Test Data Name	c_x/a	c_y/b	l_x/b	l_y/b	δ_P	δ_M	δ_E	Mode Predicted by [23]
1	0.1	0.3	0.05	0.05	4.21	0.706	2.63	WPB
2	0.1	0.3	0.15	0.15	5.27	3.27	0.475	WPB
3	0.48	0.50	0.28	0.28	7.93	2.39	2.42	WPB
4	0.16	0.1	0.15	0.15	7.96	6.76	11.1	WPB
5	0.40	0.50	0.21	0.21	8.48	1.80	1.78	WPB
6	0.45	0.41	0.23	0.23	8.77	0.137	0.61	WPB
7	0.06	0.35	0.15	0.15	10.0	1.06	0.898	WPB
8	0.06	0.35	0.05	0.05	11.0	0.579	1.14	WPB
9	0.45	0.20	0.26	0.26	11.5	12.5	2.05	WPB
10	0.25	0.30	0.37	0.37	11.6	30.1	24.8	INT
11	0.1	0.1	0.1	0.1	11.9	1.96	14.8	WPB
12	0.3	0.3	0.35	0.35	13.0	20.7	15.8	INT
13	0.25	0.25	0.32	0.32	13.4	27.6	19.4	INT
14	0.80	0.40	0.13	0.13	19.1	4.68	3.84	WPB
15	0.16	0.35	0.30	0.30	21.8	23.7	21.0	INT
16	0.10	0.30	0.25	0.25	22.2	12.3	7.98	INT
17	0.06	0.10	0.15	0.15	26.3	4.12	13.3	WPB
18	0.06	0.10	0.05	0.05	28.0	0.692	15.5	WPB
19	0.45	0.32	0.40	0.40	32.7	12.2	8.47	WPB
20	0.10	0.30	0.30	0.30	40.6	19.3	14.6	INT

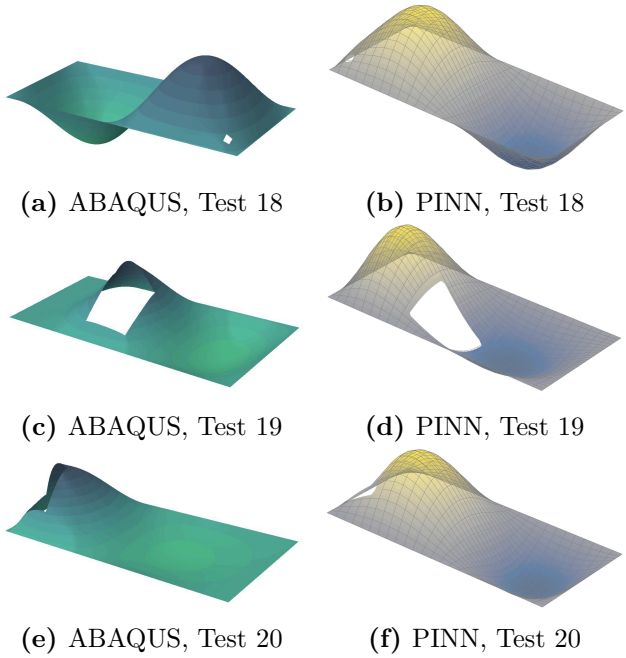


(a) ABAQUS, Test 1 **(b)** PINN, Test 1

(c) ABAQUS, Test 2 **(d)** PINN, Test 2

(e) ABAQUS, Test 3 **(f)** PINN, Test 3

Figure 9.19: Three best predictions in Test Set 2 (Table 9.10)



(a) ABAQUS, Test 18 **(b)** PINN, Test 18

(c) ABAQUS, Test 19 **(d)** PINN, Test 19

(e) ABAQUS, Test 20 **(f)** PINN, Test 20

Figure 9.20: Three worst predictions in Test Set 2 (Table 9.10)

9.8 Conclusion

this chapter compares methods used to determine the elastic buckling load of a perforated plate; FE analysis, an empirical prediction by [23], a semi-analytical prediction by [26], a simple FNN and a PINN. First, the strong-form and weak-form PINNs are compared; the weak-form PINN can accurately compute the elastic buckling load in under a tenth of the time it takes the strong-form PINN to compute. Additionally, the strong-form PINN collapses to the trivial solution when there is no data loss term, which is not required by the weak-form PINN. Furthermore, Sobol oversampling into five intervals reduces the weak-form PINN percentage error to the FE-found modeshape from 3.01% to 0.132% after only 250 epochs.

When a hole loss term is included, a forward weak-form PINN with one input can determine the elastic buckling load more accurately than the empirical prediction by [23] and a vanilla FNN (Section 9.5.1). However, as l_y becomes large, membrane stress redistribution and pronounced hole-local curvature effects become non-negligible. As the loss associated with these phenomena is not accounted for by this Rayleigh quotient PINN, it cannot accurately predict the elastic buckling load and modeshape for plates with large perforations. Future research should develop a PINN for such cases, which requires the physics loss to be formulated with Föppl-von Kármán and Reissner-Mindlin plate theories (Section 9.7.1). When the largest perforations are excluded from consideration, the PINN with one input computes the elastic buckling load and modeshape of the remaining 8 cases with a median percentage difference in accuracy to the FE solution of 1.50% after only 30 seconds (250 epochs). Thus, when used in cases where the physics is valid, a non-generalisable forward-PINN with one input can determine the elastic buckling load accurately (shown for this small set of cases). If similar accuracy is found for a greater domain of cases (such as multiple holes of various shapes and sizes), this work demonstrates that the computational speed, accuracy and ease of use associated with such a simple PINN can rival FE analysis.

A generalisable inverse-PINN was trained on many inputs according to Algorithm 1, and grid search was performed to determine the optimum number of T_{layers} and $T_{neurons}$

(where $T_{neurons}$ is the same in each layer). The data loss term was formulated with knowledge of the FE-found elastic buckling load (Equation 9.17(b)), and future work should explore how a PINN performs when formulated with the FE-found displacement at salient points (Equation 9.17(a)). A PINN with $T_{layers} = 4$ and $T_{neurons} = 150$ predicts the elastic buckling load most accurately, although it cannot identify the modeshape for plates exhibiting wavelength stiffening. Therefore, the PINN should be used only for a limited domain, and thus Equation 9.27 is imposed.

Since the PINN was trained for $0.25 \leq c_x/a \leq 0.5$ and $0.25 \leq c_y/b \leq 0.5$, it performs exceptionally poorly for $c_x > a/2$ or $c_y > b/2$. If the mirror image of the perforations in Test Set 1 is considered (Section 9.7.4.2), the median percentage difference with the $N_{cr,FE}$ capacity increases significantly to $\eta_P = 33.1\%$ and $IQR_P = 37.0\%$, despite the geometry being rotationally symmetric about the plate centroid (and thus equivalent to the original subset of models 'more similar' to the training domain). The user must ensure inputs are such that $c_x \leq a/2$ and $c_y \leq b/2$.

When applied to unseen data, the PINN determines the elastic buckling load accurately with a median absolute error of only 1.70% in comparison to the FE result (Test Set 1, Table 9.9). In addition, it can predict the elastic buckling modeshape with good accuracy, unlike the prediction by [23], which can only inform the user of the modeshape qualitatively (Table 9.9 and Table 9.10, final column). Neither the PINN nor the prediction by [23] predict the elastic buckling load particularly well when the perforation geometry does not comply with Equation 9.27 (Test Set 2, Table 9.10). However, this set of data was selected with the hypothesis that the PINN would perform poorly; thus if the PINN is trained on a wider domain of perforation sizes and locations, an improvement in accuracy can be expected.

Overall, this research demonstrates that deep learning has the potential to surpass the computational speed of current numerical methods, without compromising on accuracy and interpretability. In the near future, a PINN trained on a broader domain of cases with a more appropriate physics-found loss could greatly benefit the optimisation of perforated members thanks to its capacity for rapid results, and mesh-free modelling.

10 Conclusion

The research presented in this thesis addressed the three challenges outlined in Section 1.2. Firstly (i), existing design methods for perforated plates are often limited in either accuracy, safety, or applicability; this thesis therefore developed and verified empirical and semi-empirical models suitable for design-oriented prediction in Chapters 4 to 7. Secondly (ii), the buckling behaviour of perforated plates is generally too complex to describe analytically and often expensive to evaluate numerically; consequently, reliable semi-analytical and surrogate modelling alternatives were developed in Chapters 3 and 9 to improve computational efficiency while retaining physical insight and accuracy. Thirdly (iii), the optimisation of perforated plates typically relies on repetitive parametric FE studies; therefore, a heuristic optimisation approach was investigated in Chapter 8 to enable the efficient and intuitive exploration of large design spaces for the identification of favourable perforation arrangements.

10.1 Solution 1: Empirical, Standard-Ready Predictions

Although authors have successfully identified four types of perforated plate buckling [2, 3, 42], little attention had been given to the categorisation and modelling of their behaviour, particularly in a design standard-appealing format. This thesis resolved these gaps, by establishing standard-ready semi-empirical design equations for $N_{cr,h}$ and $N_{u,h}$ that are accurate across a wide range of cases. This accuracy is ensured by validating the expressions I developed against numerical FE models.

Importantly, these models were calibrated against both numerical results [2, 35], and an experimental campaign I undertook independently. This experimental study was the first to measure the buckling of simply supported perforated plates subject to uniform compression using DIC.

The plates examined in this study did not buckle into the number of halfwaves expected for an equivalent unperforated plate. This could suggest that the point at which plates gain a buckling halfwave depends on the local buckling mode, subject to the experimental and numerical tolerances discussed. This local buckling mode was reliably identified using a rule-based classification framework I developed, which can determine the local buckling mode, subject to geometric variables. In addition to a fifth 'interaction' mode, these are 'whole plate buckling' (WPB, typical where the perforated area, eccentricity and hole height is small); 'unstiffened strip buckling' (USB, typical where $n_r = 1$ and holes are tall with a large aspect raito); 'Euler strip buckling' (ESB, typical in plates with $n_r > 1$ and holes of a large aspect ratio); and 'lateral perforation buckling' (LPB, typical in plates with $n_r = 1$ and very tall holes $l_y \geq 0.6$). The numerical limits imposed by the categorisation algorithm were iteratively fine-tuned to maximise the percentage of accurately classified modes.

A parametric study found that only two expressions for $N_{cr,h}$ are necessary to model four of the five local buckling modes, where elastic buckling behaviour is predominantly controlled by n_r. An empirical expression proposed by [4] can be used to find $\widehat{N}_{cr,h}$ with moderate accuracy for $n_r > 1$, but not for $n_r = 1$. Conversely, the two design

equations I proposed (Equation 4.6 and Equation 4.8, Table 10.1) can determine $\widehat{N}_{cr,h}$ more accurately than [2–4], with an R^2 coefficient of 0.840.

Governing the response of plates with perforations where $l_y/b \geq 0.6$, the LPB mode was shown to exhibit fundamentally separate buckling behaviour, and cannot be found with Equation 4.6 and Equation 4.8 (Table 10.1). Across all studies in this thesis, plates exhibiting LPB are the most efficient. The elastic buckling load of these plates can be up to three times greater than an unperforated plate, as the tall holes serve to dampen deformation and redistribute force towards the unloaded edges. According to the size and location of the buckling halfwaves, the distinct LPB behaviour was classified into different sub-modes using a feed-forward neural network, which requires only four geometric inputs. Three corresponding expressions for $\widehat{N}_{cr,h}$ perform with a combined R^2 score of 0.976 against FE results (Equation 7.3, Equation 7.6 and Equation 7.8 in Table 10.1). Yet despite their high elastic buckling load, slender plates with tall holes typically fail like stocky plates, with $N_{u,h} << N_{cr,h}$.

Calibrated on GMNIA FE models, this thesis established a DSM-style prediction for the ultimate capacity $\widehat{N}_{u,h}$ (Equation 6.15, Table 10.1). Experimental and numerical results demonstrated that $N_{u,h}$ is not directly governed by the elastic buckling mode as hypothesised; instead, this study found it to be dictated by the modified slenderness $\lambda_{l,m}$ and geometric variable Ω. This variable can discern where $\widehat{N}_{u,h}$ is accurate ($\Omega \leq 0.265$ with an R^2 score of 0.935) and less accurate ($\Omega > 0.265$ with an R^2 score of 0.624), in comparison to $N_{u,FE}$. Most crucially, Ω can determine whether failure will be localised and column-like, with progressive stiffness degradation ($\Omega \geq 0.45$); or whether failure will be globalised and plate-like with a strong and distinct ultimate buckling capacity, followed by an unstable post-buckling response ($\Omega < 0.45$). Thus Ω brings significant insight into ultimate behaviour, and should be imposed by design standards on the manufacturing of perforated plates.

Importantly, the DSM expression presented in Equation 6.15 (Table 10.1) can be applied to buckling plates with either a regular or staggered array of perforations, whereas previous modifications were either only developed for plates with $n_r = 1$, or they falsely considered that $N_{cr,h} \approx N_0$. However, an important caveat must be made on these DSM curves. The

study in this thesis calibrated the GMNIA models on experimental data, and followed [18] to prescribe the imperfection size and shape. Accordingly, the proposed DSM expressions are implicitly conditioned on this assumed imperfection model, consistent with current design practice. Yet, the experimental investigation in Chapter 5 demonstrated that local buckling is highly sensitive to the imperfection shape, which could also be a twist, dent, or bow shape. Plates are particularly imperfection-sensitive, as their post-buckling behaviour strongly depends on membrane redistribution (Section 1.3.3), which imperfections disrupt. Additionally, these experiments determined that the imperfection size exceeds the value recommended in [18]. Therefore, variations in imperfection shape and amplitude may shift these DSM curves, rather than invalidate their functional form. It is hypothesised that different local buckling modes will respond differently to the same imperfection; for example, LPB behaviour is particularly governed by force redistribution, and so may be more imperfection-sensitive. Accordingly, future work must explore how the DSM expressions proposed in Table 10.1 are affected by a greater range of imperfection shapes and sizes. This could result in DSM curves with explicit mode-dependent imperfection parameters (as considered in column buckling design using the Perry-Robertson formula [1]), or with revised calibration factors to maintain target reliability levels.

Yet despite this sensitivity, the DSM expressions summarised in Table 10.1 remain valid within the bounds of the adopted imperfection model and calibration dataset, and are therefore directly comparable to existing DSM formulations in design standards. In its current form, the proposed expressions are readily applicable to AISI design standards [7], without requiring alternative LRFD resistance factors. The current resistance factor accommodates models with $\forall\Omega \in \mathbb{R}$, and it is recommended to design with $\Omega \leq 0.265$, for high confidence.

The expressions developed for $N_{cr,h}$ (Table 10.1) are also directly applicable to design standards. On the Equations 4.6 and 4.8 for $N_{cr,h}$, a statistical model factor γ is determined for implementation in EN 1993-1-5 [18]. This factor depends on the local buckling mode; for example $\gamma = 1.83$ is imposed on ESB with the worst R^2 score, whereas $\gamma = 1.06$ is imposed on WPB with the best R^2 score, at the 5% significance level.

The expressions for LPB buckling (Equation 7.3, Equation 7.6 and Equation 7.8 in Table 10.1) are intended for EN 1993-1-13 [6], which does not currently provide buckling design equations for members subject to uniform compression. The three expressions for each LPB designation can find $\widehat{N}_{cr,h}$ with $R^2 > 0.800$. It is recommended that further work should verify their accuracy over a broader domain, and determine suitable partial safety factors for complete implementation into EN 1993-1-13 [6].

In conclusion, the semi-empirical equations developed in this thesis (summarised in Table 10.1) address the first problem; they are safe, accurate, and mostly standard-ready.

Table 10.1: Proposed semi-empirical standard-ready design equations (refer to referenced equation for undefined variables)

Chapter	Eq.	Expression	Failure Type	Domain of Confidence
4	4.6	$\widehat{N}_{cr,h} = N_0 \exp\left\{ -0.714 r_p \cdot \left(\left(\frac{S_{yo}}{b}\right)^{-0.407} + \left(\frac{l_y}{b}\right)^{-0.274} + \left(\frac{l_x}{a}\right)^{0.452} \right) \right\}$	Elastic	Regular arrangement of perforations $n_r > 1$ WPB (360 models): $R^2 = 0.843$ ESB (48 models): $R^2 = 0.193$ INT (135 models): $R^2 = 0.852$
4	4.8	$\widehat{N}_{cr,h} = N_0 \exp\left\{ -1.58 \left(\left(e_x + \frac{l_y}{b}\right)^{35.3} + \left(e_y^{1.08} + (\frac{l_x}{a})^{0.561}\right)^{2.28} \right) \right\}$	Elastic	Regular arrangement of perforations $n_r = 1$ WPB (91 models): $R^2 = 0.515$ USB (46 models): $R^2 = 0.514$ INT (95 models): $R^2 = 0.592$
6	6.15	$\widehat{N}_{u,h} = \begin{cases} N_{u,e} & \text{if } \lambda_{l,m} \le \Lambda_{l,m} \\ N_{y,net} \cdot \left(1 + \kappa \left(\frac{1}{\lambda_{l,m}+1}\right)^{7.95}\right) \cdot \left(\frac{1}{\lambda_{l,m}+1}\right)^{7.95} & \text{if } \lambda_{l,m} > \Lambda_{l,m} \end{cases}$	Plastic	Regular and staggered arrangement of perforations $\Omega \le 0.265$ Regular (109 models): $R^2 = 0.927$ Staggered (54 models): $R^2 = 0.959$ $\Omega > 0.265$ Regular (91 models): $R^2 = 0.625$ Staggered (46 models): $R^2 = 0.619$

Continued on next page

Chapter	Eq.	Expression	Failure Type	Domain of Confidence
7	7.3	$\widehat{N}_{cr,h} = N_0 \Bigg[\psi_1 \Bigg\{ \Bigg(-0.13 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right)^2 + 1.94 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right) + 0.78 \Bigg) - \ln \left(6.90 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right)^2 - 4.57 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right) + 1.95 \right) \Bigg\} + \psi_2 \Bigg\{ \Bigg(-0.68 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right)^2 + 3.60 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right) + 0.43 \Bigg) - \ln \left(17.6 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right)^2 - 10 \cdot \left(\psi_3 \cdot \frac{S_{xo}}{b} \right) + 1.51 \right) \Bigg\} \Bigg]$	Elastic	Regular arrangement of perforations LPB-1 (200 models): $R^2 = 0.971$
7	7.6	$\widehat{N}_{cr,h} = N_0 \left[\left(\frac{m}{\frac{s_x}{b} - 0.45} + \frac{\frac{s_x}{b} - 0.45}{m} \right)^{0.9} - 0.8 \right]$	Elastic	Regular arrangement of perforations LPB-2 (40 models): $R^2 = 0.867$
7	7.8	$\widehat{N}_{cr,h} = \left(\frac{\alpha^2}{\beta (\ln s_x)^\gamma + \alpha^2} \right) \cdot \frac{\pi^2 E I_s}{0.274 \cdot S_{yo} L_{x,o}^2} \cdot b \left\{ 1 - \frac{l_y}{b} \right\}$	Elastic	Regular arrangement of perforations LPB-3 (40 models): $R^2 = 0.800$

10.2 Solution 2: Reliable Alternatives to Numerical Analysis

The semi-empirical predictions proposed in this thesis (Section 10.1) are safe and accurate within the domain of confidence (Table 10.1). This domain of confidence reflects what is typically designed in practice; members with a regular or staggered array of holes in their web. However, structural optimisation is concerned with the creation of new geometries, where the arrangement of perforations could be much less regular than standard design. For this domain, numerical analysis (such as the finite element method) must be used, to ensure that buckling behaviour is determined reliably. However, numerical analysis can be slow; setting the appropriate mesh, loading, and boundary conditions takes time, particularly for complex geometries.

This thesis proposed two reliable alternatives to numerical analysis; a semi-analytical prediction in Chapter 3, and a deep learning surrogate in Chapter 9. Each of these alternative methods was formulated on Rayleigh's energy method; the Rayleigh quotient was 'modified' to account for the energy lost at perforations within the 'Modified Energy Method' (MEP), and it was used as a physics loss for the neural network developed later.

Intended to accommodate perforations in any regular or irregular arrangement, the MEP assumes that the displacement function is similar to that of an unperforated plate, and consequently it performs most accurately for plates exhibiting WPB. However, the error is greater for other modes of buckling; for the models examined in [23], the MEP-found $\widehat{N}_{cr,h}$ for plates exhibiting an interaction of buckling modes is predicted half as accurately as that found for plates exhibiting WPB.

The MEP assumes that force is uniformly redistributed around the perforations, whereas [196] showed that an infinitely large plate with a circular hole under uniform remote tension exhibits an exponential redistribution of stress away from the perforation. It is recommended that future work should investigate how both the modeshape, and a modification to the additional internal strain energy term, affect the MEP accuracy. Yet despite scope for improvement, there is a clear domain where the MEP can reliably determine $N_{cr,h}$; plates with any arrangement of perforations exhibiting WPB with both

$l_y/b \leq 0.3$ and $l_x/l_y \leq 2$. For the 363 parametric models from Chapter 4 in this domain, the MEP predicts $N_{cr,h}$ with only a 3.05% difference in accuracy to $N_{cr,FE}$, superior to that found with Equations 4.6 and 4.8 (Table 10.1).

Later, I proposed a physics-informed neural network (PINN) as an alternative semi-analytical method that could be used to replace numerical analysis, within a particular domain of confidence. Unlike any of the other methods proposed in this thesis, the PINN includes a visual output. This is significant, as FE methods remain the most reliable tool in practice, partly due to their visual representation of structural response.

The domain of confidence given by Equation 9.27 was chosen to firstly restrict $l_y/b \leq 0.3$ (so that the physics loss term is valid); and secondly isolate the location where wavelength stiffening will occur, as the PINN cannot identify the subsequent modeshape change. Within this domain, the PINN predicts $N_{cr,h}$ with only 1.70% difference in accuracy to $N_{cr,FE}$ for a test subset. Yet although this method is reliable for a particular plate with small holes, it does not present an immediate rival to the versatility of numerical anlaysis.

However, the work on PINNs presented in this thesis serves as a substantial proof of concept in a previously underexplored field, and a 'stepping-stone' to more widely applicable models. Combining operator learning and importance sampling [71] with the equations proposed in Section 9.7.1, the PINN could be improved to accommodate plates of any dimension, with multiple holes of any shape or size. If the accuracy of the PINN developed in Chapter 9 can be replicated in this broader field, AI models will soon replace numerical analysis of perforated plates, mitigating computing time by providing instantaneous results, and at no sacrifice to accuracy or interpretability. Therefore, this work helps to shift the structural engineering industry from a discipline dominated by case-by-case numerical analyses to one increasingly characterised by powerful, accurate and fast semi-analytical generalisable deep-learning models.

10.3 Solution 3: Intuitive Optimisation

With further development, the work presented in the previous section may replace the current FE backbone of structural optimisation, which is stunting its use. However, it remained possible to develop a fast and intuitive optimisation framework for the arrangement of perforations in a plate without it. This was demonstrated in Chapter 8, which explored how the empirical predictions for $N_{cr,h}$ developed in the previous chapters (Table 10.1) could be used within a shape annealing optimisation framework. Those chapters found that USB and ESB modes are particularly weak, and that WPB, LPB, or an interaction of these modes should be targeted for optimal performance. Additionally, observations on the experimental tests conducted in Chapter 5 determined that holes placed along the plate midline stabilise the elastic response, and that a regular arrangement of perforations is more structurally efficient. As a consequence, the optimisation framework strictly outputs plates with a rotationally symmetric distribution of perforations.

The stochastic property of shape annealing enables the discovery of geometries more optimal than those obtained through slow and deterministic FE optimisation. Shape annealing optimisation can correctly find minima in the loss landscape, and found that plates with multiple rows of uniformly sized and spaced perforations (i.e., rack sections) are less efficient than plates with one row of tall perforations, or plates with one column of very small perforations at the plate centre. As expected, none of the optimised geometries exhibit USB or ESB, and even subtle changes in a single geometric variable can greatly affect the elastic buckling load, complementing the observations made in the experimental tests of Chapter 5. As a general guideline for the design of rack section webs, the study presented in Chapter 8 established that efficiency is maximised by orienting perforations with their major dimension perpendicular to the load-axis, minimising perforation area, and confining perforations towards the centre of the plate. Rack sections can be made even more efficient if perforations can be non-uniform in size and shape.

Thus, this shape annealing framework can be used to inform novel perforation patterns which improve strength, while adhering to practical design constraints. This framework was developed into Abalone [29, 30], a plug-in for Grasshopper [31] for easy and intuitive

real-world application. Future modifications should accommodate the variable Ω, and equations developed for plastic capacity (Equation 6.15, Table 10.1), to ensure that the optimised geometries adhere to the stronger globalised plastic failure mode observed for plates with tall perforations.

10.4 Final remarks

This thesis has established (i) reliable empirical expressions that describe the behaviour of local buckling modes, categorised by a simple classification algorithm; (ii) fast semi-analytical methods that could soon replace expensive FE simulations in optimisation routines; and (iii) an intuitive optimisation framework, which can determine the most efficient distribution of perforations in a buckling plate. Importantly, these methods operate accurately within clearly defined domains, and are supported by a rigorous and novel experimental campaign. This work is directly applicable to real-world contexts; the semi-empirical expressions are aligned to Eurocode and AISI standards, and the optimisation framework has been transformed into an intuitive working tool, which can find perforation designs that are both efficient, and practical for use in construction. The PINN is a significant first step towards a new paradigm of structural optimisation, in which AI-based models may be used as surrogates for FE analysis, while retaining the visual interpretability that is often lost in other techniques. In conclusion, the methods and tools developed in this thesis are highly exploitable, and can be reliably used to design and assess thin perforated plates subject to buckling.

Appendices

A

Mesh Convergence Study

In Chapter 3 and Chapter 4, a mesh convergence study for S8R5 elements was undertaken. These elements are well suited to linear eigenvalue buckling analyses of thin perforated plates as they can accurately capture bending-dominated instability modes without excessive mesh sensitivity or locking effects, particularly in combination with a structured mesh (as was chosen). Compared with 4-node elements, fewer of these quadratic elements are required as they allow for an improved representation of curvature and local deformation. Additionally, reduced integration lowers computational cost and avoids excessive artificial stiffness. Finally, only 5 degrees of freedom per node are necessary as for a bending-dominated linear elastic thin-shell buckling perturbation analysis, the θ_3 rotation is unimportant, permitting cleaner buckling modes with less numerical overhead. The relevant mesh convergence study is shown in Figure A.1, where the 'accuracy' is the ratio of the ABAQUS-found capacity to that given by Timoshenko's plate buckling equation.

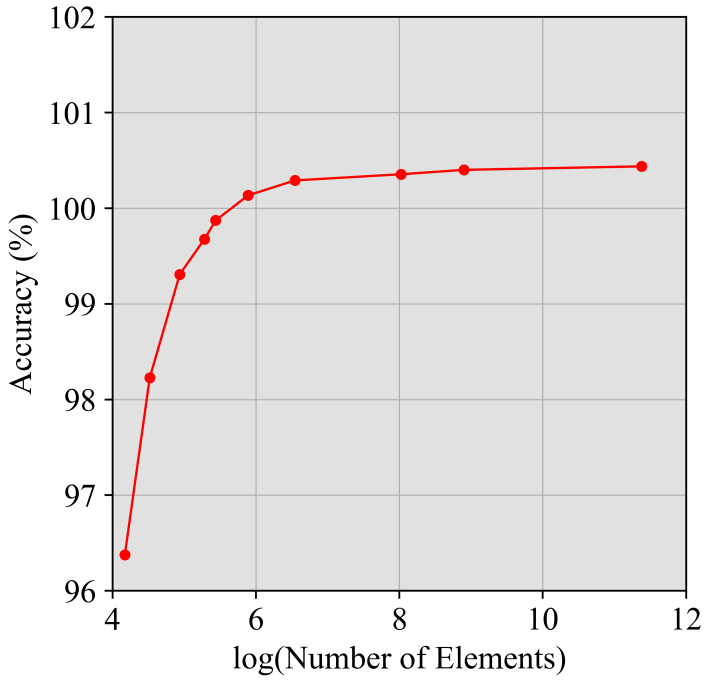


Figure A.1: Mesh convergence study for S8R5 elements (Chapter 3 and Chapter 4)

B

Linear And Nonlinear ABAQUS Input File Creator

```
# -*- coding: utf-8 -*-

# . . . . . .
    . . . . . #

                    #       S Y N O P S I S       #

# . . . . . .
    . . . . . #

# This code creates .inp files for ABAQUS

# It reads a list of input Excel files listed as rows in the '
   bank' file

# The user can select either Steel or Aluminium as the material

# The user can choose either S4 or S4R elements

# A linear (_L) and a nonlinear (_NL) input file is created for
   ABAQUS

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

# 1. Linear .inp file
```

```
# The code reads an input file, which is an Excel file with two
   worksheets:

    # PC: Co-ordinates of the plate i.e. the length (XMAX - XMIN)
       and the width (YMAX - YMIN), and the plate thickness

    # HC: Co-ordinates of the holes i.e. the length (x2 - x1) and
       the width (y2 - y1)

# The user specifies the minimum and maximum permissible element
   size

# Two extra worksheets are saved to the Excel file:

    # NODES: with the node number, x, y and z co-ordinates (4
       columns)

    # ELEMENTS: with the element number, n1, n2, n3 and n4 (5
       columns) for an S4 or S4R element. Quadratic elements are
       not possible with this code

# The linear .inp file is saved and ready to be submitted in
   ABAQUS command.

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - - - - - - #

# 2. Nonlinear .inp file

# This requires all four Excel worksheets to exist in the Excel
   file

# Different nonlinear controls and parameters can be chosen:

    # Newton Raphson Method with no damping (NO)

    # Newton Raphson Method with damping (ND)

    # Newton Raphson Method with time incrementation controls (NT
       )

    # Riks Arc Length Method with load control (L) or
       displacement control (D)

        # Note that the Newton Raphson Method parameters
           considered for each of these nonlinear controls is
           found in:
            #Z. Yao., K. J. Rasmussen., 2014, Design of
               Perforated Thin-Walled Steel Columns (No. R949),
```

```
                SeS faculties schools: Faculty of Engineering:
                School of Civil Engineering, https://hdl.handle.net
                /2123/24066

# The user may choose the number eigenmode considered as the
   geometric imperfection. If unspecified, the default of 1 is
   chosen. This can be determined by the user after running the
   linear input file in ABAQUS.

# The nonlinear .inp file is saved and ready to be submitted in
   ABAQUS command.

# . . . . . .
    . . . . . #

            #       I M P O R T    L I B R A R I E S        #

# . . . . . .
    . . . . . #

import itertools
import numpy as np
import pandas as pd
from os import listdir
from os.path import isfile, join

# . . . . . .
    . . . . . #

            #       H O M E - M A D E    F U N C T I O N S
                      #

# . . . . . .
    . . . . . #

# THIS FUNCTION TAKES A LIST OF ITEMS AND TURNS IT INTO A LIST OF
    ONE ITEM, A STRING OF ALL THE ITEMS.
def HAPPY_STRING(happy_list):
    combo = ''.join(map(str,happy_list))
    happy_list = [combo]

# THIS FUNCTION TAKES A LIST OF LISTS AND PUTS IT INTO A LIST OF
   SINGLE ITEMS.
def SMUSH(s):
    return [item for sublist in s for item in sublist]

# Nodes must lie outside of a hole region
# The hole region is defined if the central co-ordinate in the
   list is at the centre of a hole
```

```
# This is identified with the hole_region_identifier variable (
   True for no hole, False for hole)
def Nodes_In_Region(idr):

    hole_region_identifier = True

    XCORDS = []
    YCORDS = []

    for x in X_NODES:
        if D_R_DF.iloc[idr,0]<=x<=D_R_DF.iloc[idr,1]:
            XCORDS.append(x)

        XCORDS = list(dict.fromkeys(XCORDS))
        LENXM = int(len(XCORDS)/2)

    for y in Y_NODES:
        if D_R_DF.iloc[idr,2]<=y<=D_R_DF.iloc[idr,3]:
            YCORDS.append(y)

        YCORDS = list(dict.fromkeys(YCORDS))
        LENYM = int(len(YCORDS)/2)

    for ih in list(range(N_HOLES)):

        if HOLECOORDS.iloc[ih,0]<= XCORDS[LENXM] <= HOLECOORDS.
           iloc[ih,1] and HOLECOORDS.iloc[ih,2]<= YCORDS[LENYM] <=
            HOLECOORDS.iloc[ih,3]:
            hole_region_identifier = False

    return hole_region_identifier, XCORDS, YCORDS

# Fast vectorised search for adjacent nodes to define elements
def Define_Element(nodes_lookup, data_frame, x, y, new_id):

    tmp = data_frame.merge(nodes_lookup, how = 'left', left_on=[x
       , y], right_on=['x', 'y']).rename(columns={'nid': new_id}).
       drop(columns=['x', 'y'])

    return tmp

# Functions used for writing the .inp files
def EXTRACTOR(NAME):
    data_store = ""
    with open(NAME) as fp:
        data = ""
        data = fp.read()
        data_store+=data
    return(data_store)
```

```
def LIST_WRITER(NAME):
    with open("C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
       COMPUTATIONAL/RIKS_PACKAGE/TEMPORARY_FILE.txt","w+") as f:
        count = 1
        print(len(NAME))
        for i in NAME:
            if NAME.index(i)==0 and count==1:
                f.write(str(i))
            elif NAME.index(i)!=0 and (count-1)%16!=0 :
                f.write(","+str(i))
            elif NAME.index(i)!=0 and (count-1)%16 ==0:
                f.write(str(i))
            if count%16==0 and i!=NAME[-1]:
                f.write("\n")
            count+=1
            #print(i)
    return("C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
       COMPUTATIONAL/RIKS_PACKAGE/TEMPORARY_FILE.txt")
# . . . . . .
    . . . . . #

                    #       S E T - U P       #

# . . . . . .
    . . . . . #
# Set to True if want to create the following .inp files (both
   can be true)
Linear_Input_File = True
Nonlinear_Input_File = True
# Set to false and default of the first eigenmode is selected.
   Otherwise, user can specify the first positive eigenmode if
   Eigenmode_Choice is True.
Eigenmode_Choice = False

MATERIAL = "STEEL_SD"
# STEEL_RO  >> RAMBERG OSGOOD MODEL WITH ABAQUS FROM https://
   classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/
   v6.6/books/stm/default.htm?startat=ch04s03ath111.html
# STEEL_SD  >> STEEL DESIGN MANUAL EQUATIONS FROM P.G. 155 (STEEL
    DESIGN 4TH EDITION - SEE RAMBERG OSGOOD FOLDER ON NEXUS, SEE
   1. New material Model - Ellie Excel)
# ALUMINIUM >> MEASURED FROM TENSILE TEST DATA (see 01
   _11_MATERIAL_DATA in C:\Users\pemb6626\OneDrive - Nexus365\
   Documents\YEAR2\WP1_PLATES\EXPERIMENTS_PROCESSING\EXCEL\
   MATERIAL)

#(SEE Yao, Z., & Rasmussen, K. (2014). Design of Perforated Thin-
   Walled Steel Columns. Sydney: The University of Sydney School
```

```
   of Civil Engineering.  p. 21)
if Nonlinear_Input_File:
    inc = int(input("N U M B E R   O F   I N C R E M E N T S = ")
       )
    CONTROL = str(input("""S E L E C T   N O N - L I N E A R   S
       O L U T I O N   C O N T R O L
RIKS METHODS (RECOMMENDED WHEN EXPECT SNAP-THROUGH)
L >> RIKS ARC LENGTH LOAD CONTROL
D >> RIKS ARC LENGTH DISP CONTROL

NEWTON RAPHSON METHOD (RECOMMENDED BY YAO & RASMUSSEN, 2014)
NO >> NEWTON RAPHSON WITH NO DAMPING - START
ND >> NEWTON RAPHSON WITH DAMPING - ONLY SELECT IF NO DID NOT
   CONVERGE
NT >> NEWTON RAPHSON WITH TIME INCREMENTATION CONTROL

>> """))

    assert CONTROL != ("L", "D", "NR", "NO", "ND", "NT"), "I N V
       A L I D   S O L U T I O N   C O N T R O L   C H O S E N"

    if CONTROL == "L" or CONTROL == "D":
        print("RIKS METHOD SELECTED.")

        Default = str(input("""D E F A U L T   V A L U E S   F O
            R   R I K S ? (Y/N)
    Initial Step Size = 1
    Arc Length = 5
    Min Step Size = 0.0001
    Max Step Size = 5
    Load Proportionality Factor = 2
    Max Displacement = 5"""))
        if Default == "N":
            start_ss = input("I N I T I A L   S T E P   S I Z E =
                ")
            arc = input("A R C   L E N G T H  = ")
            min_ss = input("M I N   S T E P   S I Z E = ")
            max_ss = input("M A X   S T E P   S I Z E = ")
            lpf = input("L O A D   P R O P O R T I O N A L I T Y
                 F A C T O R = ")
            disp_max = input("M A X   D I S P L A C E M E N T =
               ")
        else:
            start_ss = 1                    # DEFAULT >> 1
            arc = 5                         # DEFAULT >> 5
            min_ss = 0.0001                 # DEFAULT >> 0.0001
            max_ss = 5                      # DEFAULT >> 5
            lpf = 2.0                       # DEFAULT >> 2
            disp_max = 5                    # DEFAULT >> 5
```

```
    elif CONTROL == "NO" or CONTROL == "ND" or CONTROL == "NT":
        print("""NEWTON RAPHSON METHOD WITH DAMPING SELECTED.
    ITERATIVELY INCREASE THE ACCURACY_TOLERANCE BY THE DEFAULT
       VALUE""")

        Default = str(input("""D E F A U L T   V A L U E S   F O
            R   N R   W I T H   D A M P I N G ? (Y/N)
    Accuracy Tolerance = 0.005
    Initial Step Size = 0.001
    Time Increment = 1
    Min Step Size = 0.000001
    Max Step Size = 0.015
    Max Displacement = 1""" ))
        if Default == "N":
            accuracy_tolerance = input("A C C U R A C Y   T O L E
                R A N C E = ")
            start_ss_nr = input("I N I T I A L   S T E P   S I Z
               E = ")
            time_nr = input("T I M E   I N C R E M E N T = ")
            min_ss_nr = input("M I N   S T E P   S I Z E = ")
            max_ss_nr = input("M A X   S T E P   S I Z E = ")
            nr_disp = input("M A X   D I S P L A C E M E N T = ")
        else:
            accuracy_tolerance = 0.005      # DEFAULT >> 0.005,
               INCREASE BY 0.005 FOR EVERY UNCONVERGED SIMULATION
            start_ss_nr = 0.001             # DEFAULT >> 0.001
            time_nr = 1                     # DEFAULT >> 1
            min_ss_nr = 0.000001            # DEFAULT >> 0.000001
            max_ss_nr = 0.015               # DEFAULT >> 0.015
            nr_disp = 1                     # DEFAULT >> 1

element_type = "S4"
# S4  >> 4-NODE GENERAL PURPOSE FULLY INTEGRATED FINITE MEMBRANE
   STRAIN QUAD SHELL ELEMENT
# S4R >> ABAOVE, WITH REDUCED INTEGRATION

# MINIMUM ELEMENT SIZE PERMITTED
min_ele_size = 0.5
# MAXIMUM ELEMENT SIZE PERMITTED
max_ele_size = 1
# AIM TO SPECIFY NO MORE THAN MAX_ELE_SIZE = 2 * MIN_ELE_SIZE TO
   TARGET ELEMENTS WITH A SMALL ASPECT RATIO, BETTER FOR MESHING,
   SEE:
    # C. Moen., 2008. Direct Strength Design of Cold-Formed Steel
        Members with Perforations. Dissertation (Ph. D), Vols.
       Johns Hopkins University, Baltimore, Maryland.

DIRI_EXT = "29_07_Staggered_Perforations - Copy/"
```

```
# FOLDER WHERE THE BANK FILE IS STORED
DIRECTORYJ = "E:/DOCUMENTS/YEAR3/COMPUTATIONAL/EXCEL_LIBRARY/"
# FOLDER WHERE THE ABAQUS INPUT FILES WILL BE SAVED
DIRECTORYO = "E:/DOCUMENTS/YEAR3/COMPUTATIONAL/ABAQUS_INPUTS/"
# FOLDER PROVIDING TEMPORARY STORAGE (LIKE A RECYCLE BIN)
DIRECTORYX = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/"
# BANK FILE NAME
EXCEL_NAME = "29_07_Staggered_Perforations_Bank_Copy.xlsx"
SHEET_NAME = "Sheet1"

mypath = r"C:\Users"
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f)
   )]

SKIPROWS = 0 #list(range(1,17))
# IF NOT 0, SKIPROWS MUST START WITH 1 SO THAT HEADERS ARE
   INCLUDED, AND ADD THE skiprows = SKIPROWS in below line

DATA = pd.read_excel(DIRECTORYJ + DIRI_EXT + EXCEL_NAME,
   index_col = "NAME", sheet_name = SHEET_NAME, skiprows=SKIPROWS)

# LIST OF EXCEL FILES TO CREATE INPUT FILES FOR ABAQUS FROM
INP_FIL_NAME_LIST = DATA.index.tolist()
print(INP_FIL_NAME_LIST)

# . . . . . .
    . . . . . #

            #       C R E A T I N G    T H E    M E S H        #

# . . . . . .
    . . . . . #

for FIL_NAME in INP_FIL_NAME_LIST:

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    # Read the PC and HC worksheets
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #

    TITLE = FIL_NAME

    INP_FIL_NAME = DIRECTORYJ + DIRI_EXT + FIL_NAME
    OUT_FIL_NAME_L = DIRECTORYO + DIRI_EXT+ FIL_NAME +"_L.inp"
    OUT_FIL_NAME_NL = DIRECTORYO + DIRI_EXT+ FIL_NAME +"_NL" +
       str(CONTROL) + ".inp"
```

```
n_sheets = len((pd.ExcelFile(INP_FIL_NAME +".xlsx")).
   sheet_names)

PLATECOORDS = pd.read_excel(INP_FIL_NAME +".xlsx",sheet_name
   ='PC')

XMIN = PLATECOORDS.iloc[0,0]
XMAX = PLATECOORDS.iloc[0,1]
YMIN = (PLATECOORDS.iloc[0,2])
YMAX = (PLATECOORDS.iloc[0,3])
thickness = PLATECOORDS.iloc[0,4]

MIDX = (XMAX-XMIN)/2
MIDY = (YMAX-YMIN)/2

HOLECOORDS = pd.read_excel(INP_FIL_NAME +".xlsx",sheet_name='
   HC', index_col="number")

# The total number of holes in the plate
N_HOLES = len(HOLECOORDS.iloc[:,0])

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Make a list of the x and y co-ordinates marking important
   locations, i.e.
    # The x and y co-ordinate of the holes
    # The x and y co-ordinate mid-way along each edge (MIDX,
       MIDY)
    # The maximum x and y co-ordinae (XMAX, YMAX)
# Co-ordinates are rounded to 5 decimal places and any
   repeats are removed.
# The list is ordered from smallest to greatest.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

# x co-ordinates
X_HC_DF = HOLECOORDS.iloc[:,0:2]
X_HC_LIST = list(dict.fromkeys(SMUSH(X_HC_DF.values.tolist())
   ))

X_HC_LIST.append(MIDX)
X_HC_LIST.append(XMAX)
X_HC_LIST.sort()
X_HC_LIST = [round(elem,5) for elem in X_HC_LIST]
X_HC_LIST = list(dict.fromkeys(X_HC_LIST))

# y co-ordinates
Y_HC_DF = HOLECOORDS.iloc[:,2:4]
```

```
Y_HC_LIST = list(dict.fromkeys(SMUSH(Y_HC_DF.values.tolist())
   ))

Y_HC_LIST.append(MIDY)
Y_HC_LIST.append(YMAX)
Y_HC_LIST.sort()
Y_HC_LIST = [round(elem,5) for elem in Y_HC_LIST]
Y_HC_LIST = list(dict.fromkeys(Y_HC_LIST))

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# The nodes between the important locations found previously
   are now determined.
# The region between two important locations is of length l
# l is split into an integer number n such that d = l/n
# Where d must satisfy min_ele_size <= d <= max_ele_size
# The nodes are appended to the appropriate list
# If min_ele_size is too large, the while loop exits and the
   user must reduce min_ele_size
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

# x co-ordinates
X_NODES = []
for i in list(range(len(X_HC_LIST))):
    if i != 0:
        xi = X_HC_LIST[i]
        xo = X_HC_LIST[i-1]
    else:
        xi = X_HC_LIST[i]
        xo = XMIN

    l = xi-xo
    n = 1
    d = l/n
    s = True

    while s:
        if min_ele_size<=d<=max_ele_size:
            s = False
        else:
            n+=1
            d = l/n
            assert n < 1000, "Minimum element size too large.
                Try reducing min_ele_size"
    x = xo
    X_NODES.append(x)
    while x<xi:
        x+=d
```

```
            X_NODES.append(x)
    X_NODES.append(XMAX)

    # y co-ordinates
    Y_NODES = []
    for i in list(range(len(Y_HC_LIST))):

        if i != 0:
            yi = Y_HC_LIST[i]
            yo = Y_HC_LIST[i-1]
        else:
            yi = Y_HC_LIST[i]
            yo = YMIN

        l = yi-yo
        n = 1
        d = l/n
        s = True

        while s:
            if min_ele_size<=d<=max_ele_size:
                s = False
            else:
                n+=1
                d = l/n
                assert n < 1000, "Minimum element size too large.
                    Try reducing min_ele_size"
        y = yo
        Y_NODES.append(y)
        while y<yi:
            y+=d
            Y_NODES.append(y)
    Y_NODES.append(YMAX)

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    # The previous list comprehension stops appending when the
       element has exceeded the boundary limit.
    # e.g. for a region between 20 and d=2.25, we have in the
       list:
        # 20, 22.25, 24.5, 26.75
    # The next region could be between 25 and 30, therefore the
       final list is:
        # 20, 22.25, 24.5, 26.75, 25, 27.25, ...
    # But 25 < 26.75 and so this would make very erroneous
       elements
    # Therefore if the element at 'val - 1' > element at 'val',
       it is deleted from the list.
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

lengthx = len(X_NODES)
val=1
while val<lengthx-1:
    if X_NODES[val+1]<X_NODES[val] or X_NODES[val-1]>X_NODES[
       val]:
        X_NODES.pop(val)
        lengthx= len(X_NODES)
    val+=1

lengthy = len(Y_NODES)
val=1
while val<lengthy-1:
    if Y_NODES[val+1]<Y_NODES[val] or Y_NODES[val-1]>Y_NODES[
       val]:
        print(Y_NODES[val])
        Y_NODES.pop(val)

        lengthy= len(Y_NODES)
    val+=1

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# We also want to mitigate really small elements like the
   following:
    # 20, 24.4449, 25.5, ...
# Therefore such elements are deleted from the list.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

lengthx = len(X_NODES)
val=1
while val<lengthx-1:
    if float(round(X_NODES[val+1],1))==float(round(X_NODES[
       val],1)) or float(round(X_NODES[val+1],2))==float(round
       (X_NODES[val],2)):
        X_NODES.pop(val)
        lengthx= len(X_NODES)
    val+=1

lengthy = len(Y_NODES)
val=1
while val<lengthy-1:
    if float(round(Y_NODES[val+1],1))==float(round(Y_NODES[
       val],1)) or float(round(Y_NODES[val+1],2))==float(round
       (Y_NODES[val],2)):
        print(Y_NODES[val])
```

```
            Y_NODES.pop(val)

            lengthy= len(Y_NODES)
        val+=1

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    # Append the minimum co-ordinate to the relevant list
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    X_HC_LIST.append(XMIN)
    X_HC_LIST.sort()

    Y_HC_LIST.append(YMIN)
    Y_HC_LIST.sort()

    print("SORTED")

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    # Creating the nodes

    # First, a dataframe informs what the regions are in terms of
        x1r, x2r, y1r, y2r
    # i.e. left-most co-ordinate, right-most co-ordinate, bottom-
       most co-ordinate, top-most co-ordinate
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #

    x1r = []
    x2r = []
    y1r = []
    y2r = []

    yi = 0
    xi = 0

    while yi +1<len(Y_HC_LIST):
        xi = 0
        while xi+1< len(X_HC_LIST):
            x1r.append(X_HC_LIST[xi])
            x2r.append(X_HC_LIST[xi+1])
            y1r.append(Y_HC_LIST[yi])
            y2r.append(Y_HC_LIST[yi+1])
            xi+=1
        yi+=1

    DATA_REGION = {"x1r":x1r,"x2r":x2r,"y1r":y1r,"y2r":y2r}
    D_R_DF = pd.DataFrame(DATA_REGION)
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# For each row in the region dataframe, I want a list of x
   and y co-ordinates at locations from the list of nodes
# Use the Nodes_In_Region function to identify the eligible
   regions
# Nodes associated with regions outside of holes are appended
    to the final node list.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

nrdr, ncdr = D_R_DF.shape
idr = 0

XFINALS = []
YFINALS = []
FINALS = []
NONFINALS = []

while idr < nrdr:

    REGION = []

    hole_region_identifier, XCORDS, YCORDS = Nodes_In_Region(
       idr)

    if hole_region_identifier:
        ly = 0
        while ly<len(YCORDS):
            lx = 0
            while lx<len(XCORDS):

                REGION.append([XCORDS[lx],YCORDS[ly]])

                lx+=1
            ly+=1

        FINALS.append(REGION)
    idr+=1

# NEXT LINE CREATES A LIST OF VALUES WITH COORDINATES X,Y,X,Y
   ...
FINALS = SMUSH(FINALS)
FINALS.sort()
FINALS = list(FINALS for FINALS,_ in itertools.groupby(FINALS
   ))
FINALS = SMUSH(FINALS)
```

```
for i in list(range(len(FINALS))):
    if i%2 == 0:
        XFINALS.append(FINALS[i])
    else:
        YFINALS.append(FINALS[i])

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Fine-comb for more repeats. Check the y-values and delete
   accordingly.
# Must delete the x-value with the same index.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

lengthy = len(YFINALS)
val=1
while val<lengthy:
    if float(round(YFINALS[val-1],1))==float(round(YFINALS[
       val],1)):
        YFINALS.pop(val-1)
        XFINALS.pop(val-1)
        lengthy= len(YFINALS)
    val+=1

newindex = list(range(1,len(XFINALS)+1))
DATA_NODES = {"x":XFINALS,"y":YFINALS,"z":[0]*len(XFINALS)}
D_N_DF = pd.DataFrame(DATA_NODES,index=newindex)

nr, nc = D_N_DF.shape
in_final=nr
D_N_DF_ROUNDED = D_N_DF.round(4)

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Creating the elements

# Elements must not be accross regions as this would be
   erroneous.
# Therefore, elements are sorted according to D_R_DF
# All of the x- and y- nodes associated with each region are
   again identified,
# and nodes within a 'hole region' are skipped according to
   the hole_region_identifier variable
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

EL1 = []
EL2 = []
EL3 = []
```

```
    EL4 = []

    idr = 0

    while idr < nrdr:

        X1 = []
        X2 = []
        Y1 = []
        Y2 = []

        hole_region_identifier, XCORDS, YCORDS = Nodes_In_Region(
           idr)

        if hole_region_identifier:
            ly = 0
            while ly+1<len(YCORDS):#max(L_R_Y):
                lx = 0
                while lx+1<len(XCORDS):

                    X1.append(XCORDS[lx])
                    Y1.append(YCORDS[ly])
                    X2.append(XCORDS[lx+1])
                    Y2.append(YCORDS[ly+1])

                    #N_VAL += 1
                    lx+=1
                ly+=1

            DATA_EL_NODES = {"X1":X1,"X2":X2,"Y1":Y1,"Y2":Y2}
            D_EN_DF = pd.DataFrame(DATA_EL_NODES).round(4)

            # - - - - - - - - - - - - - - - - - - - - - - - - - - -
                - - - - - - #
            # Nodes are ordered anticlockwise from the bottom
               left corner in ABAQUS S4 elements
            # The appropriate neighboring nodes for each element
               is search for using fast vectorised mapping.
            # The nodes are appended to the master list defined
               outside of the while loop
            # - - - - - - - - - - - - - - - - - - - - - - - - - - -
                - - - - - - #
            nodes_lookup = D_N_DF_ROUNDED.reset_index().rename(
               columns={'index': 'nid'})[['nid', 'x', 'y']]

            tmp = Define_Element(nodes_lookup, D_EN_DF, 'X1', 'Y1
               ', 'N1')
            tmp = Define_Element(nodes_lookup, tmp, 'X2', 'Y1', '
               N2')
```

```
            tmp = Define_Element(nodes_lookup, tmp, 'X2', 'Y2', '
               N3')
            tmp = Define_Element(nodes_lookup, tmp, 'X1', 'Y2', '
               N4')

            D_E_DF_part = tmp[['N1', 'N2', 'N3', 'N4']]

            # Drop degenerate or missing elements
            D_E_DF_part = D_E_DF_part.dropna()
            D_E_DF_part = D_E_DF_part.astype(int)
            D_E_DF_part = D_E_DF_part[
                (D_E_DF_part['N1'] != D_E_DF_part['N4'])
                & (D_E_DF_part['N1'] != D_E_DF_part['N2'])
            ]

            # Append to master lists
            EL1.extend(D_E_DF_part['N1'].tolist())
            EL2.extend(D_E_DF_part['N2'].tolist())
            EL3.extend(D_E_DF_part['N3'].tolist())
            EL4.extend(D_E_DF_part['N4'].tolist())

        idr+=1

    # Element dataframe
    DATA_ELES = {"N1":EL1,"N2":EL2,"N3":EL3,"N4":EL4}
    D_E_DF = pd.DataFrame(DATA_ELES)

    # Get rid of illegal groupings
    D_NEW = (D_E_DF.index[(D_E_DF["N1"] == D_E_DF["N4"])]).tolist
       ()
    D_E_DF.drop(D_NEW, axis=0, inplace=True)
    D_NEW = (D_E_DF.index[(D_E_DF["N1"] == D_E_DF["N2"])]).tolist
       ()
    D_E_DF.drop(D_NEW, axis=0, inplace=True)

    D_E_2_r1 = np.array((D_E_DF).iloc[:,0])
    D_E_2_r2 = np.array((D_E_DF).iloc[:,1])
    D_E_2_r3 = np.array((D_E_DF).iloc[:,2])
    D_E_2_r4 = np.array((D_E_DF).iloc[:,3])

    # Re-index element dataframe into final state (omitting
       illegal groupings)
    newindex = list(range(1,len(D_E_2_r1)+1))
    DATA_ELES = {"N1":D_E_2_r1,"N2":D_E_2_r2,"N3":D_E_2_r3,"N4":
       D_E_2_r4}
    D_E_DF = pd.DataFrame(DATA_ELES,index=newindex)
    er, ec = D_E_DF.shape
    ie_final=er
```

```
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - #
    # Save element dataframe to ELEMENTS worksheet in Excel input
        file (if they don't already exist)
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - #
    if n_sheets !=4:
        with pd.ExcelWriter(INP_FIL_NAME +".xlsx",mode='a') as
           writer:
            D_N_DF.to_excel(writer,"NODES")
            D_E_DF.to_excel(writer,"ELEMENTS")

# . . . . . .
    . . . . . #

#       C R E A T I N G    T H E    L I N E A R    I N P U T    F
    I L E       #

# . . . . . .
    . . . . . #

    D_N_DF = pd.read_excel(INP_FIL_NAME+".xlsx",sheet_name="NODES
       ")
    D_N_DF_inds = pd.read_excel(INP_FIL_NAME+".xlsx",sheet_name="
       NODES",index_col=0)
    nr, nc = D_N_DF.shape
    number_of_nodes=nr

    D_E_DF = pd.read_excel(INP_FIL_NAME+".xlsx",sheet_name='
       ELEMENTS')
    D_E_DF_inds = pd.read_excel(INP_FIL_NAME+".xlsx",sheet_name='
       ELEMENTS',index_col=0)
    nr, nc = D_E_DF.shape
    number_of_elements=nr

    X_L_NSET=(D_N_DF_inds.index[(D_N_DF_inds["x"]==XMIN)]).tolist
       ()
    X_R_NSET=(D_N_DF_inds.index[(D_N_DF_inds["x"]==XMAX)]).tolist
       ()

    Y_T_NSET=(D_N_DF_inds.index[(D_N_DF_inds["y"]==YMAX)]).tolist
       ()
    Y_B_NSET=(D_N_DF_inds.index[(D_N_DF_inds["y"]==YMIN)]).tolist
       ()

    X_L_MID = X_L_NSET[int(len(X_L_NSET)/2)]
    X_R_MID = X_R_NSET[int(len(X_R_NSET)/2)]

    Y_B_MID = Y_B_NSET[int(len(Y_B_NSET)/2)]
```

```
Y_T_MID = Y_T_NSET[int(len(Y_T_NSET)/2)]

# SETS
# LOADED POINT (MAKE IN LINE WITH MID-POINT)
co_ord_x_load_left = XMIN - 10
co_ord_x_load_right = XMAX + 10

co_ord_y_load_left = MIDY
co_ord_y_load_right = MIDY

co_ord_z_load_left = 0
co_ord_z_load_right = 0

# FIRST LINES FOR THE HEADING AND JOB NAMES ETC
D_N_DF.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES.
   txt",header=False,index=False)
D_E_DF.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS.txt",header=False,index=False)

D_N_DF_inds.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/
   WP1_PLATES/COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_NODES_INDS.txt",header=False,index=False)
D_E_DF_inds.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/
   WP1_PLATES/COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS_INDS.txt",header=False,index=False)

NODES = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES.
   txt"
ELEMENTS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS.txt"

NODES_INDS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_NODES_INDS.txt"
ELEMENTS_INDS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES
   /COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS_INDS.txt"

WRITE_FILE = "10_06_PRACTICE"
READ_FILE = [NODES, ELEMENTS]

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Create Part
```

```
# - - -
```

```
    P5 += "\n*Nset, nset=Y_T_EDGE, instance=PART-1-1\n"+EXTRACTOR
       (LIST_WRITER(Y_T_NSET))
    P5 += "\n*Nset, nset=loaded_left\n1,"
    P5 += "\n*Nset, nset=loaded_right\n2,"
    P5 += "\n*Surface, type=NODE, name=X_L_EDGE_CNS_, internal\
       nX_L_EDGE, 1."
    P5 += "\n*Surface, type=NODE, name=X_R_EDGE_CNS_, internal\
       nX_R_EDGE, 1."
    P5 +="\n** Constraint: Constraint-1\n*Coupling, constraint
       name=Constraint-1, ref node=loaded_left, surface=
       X_L_EDGE_CNS_\n*Distributing, weighting method=UNIFORM"
    P5 +="\n** Constraint: Constraint-2\n*Coupling, constraint
       name=Constraint-2, ref node=loaded_right, surface=
       X_R_EDGE_CNS_\n*Distributing, weighting method=UNIFORM\n*
       End Assembly\n**\n"

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    # Define material properties
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - #
    if MATERIAL == "STEEL_RO":
        Default = str(input("""D E F A U L T   V A L U E S   F O
            R   S T E E L ? (Y/N)
Young's Modulus = 203400 MPa
Poisson's Ratio = 0.3
Yield Strength = 450 MPa
Hardening Exponent = 8
Yield Offset = 0.904"""))
        if Default == "N":
            E = str(input("Y O U N G S   M O D U L U S   (M P a)
               = "))
            v = str(input("P O I S S O N ' S   R A T I O = "))
            ys = str(input("Y I E L D   S T R E N G T H   (M P a)
                = "))
            nexp = str(input("H A R D E N I N G   E X P O N E N T
                = "))
            alpha = str(input("Y I E L D   O F F S E T = "))
        else:
            E = "203400"
            v = "0.3"
            ys = "450"
            nexp = "8"
            alpha = "0.904"
        P6 = f"** MATERIALS\n**\n*Material, name={MATERIAL}\n*
           Deformation Plasticity\n"+E+","+v+","+ys+","+nexp+","+
           alpha+"\n**"
    elif MATERIAL == "STEEL_SD":
         P6 = f"""** MATERIALS
```

```
**
*Material, name={MATERIAL}
*Elastic
 203400, 0.3
*Plastic
 450,          0.
 455.09, 0.00021
 460.23, 0.00051
 465.52, 0.00111
 471.12, 0.00239
 477.29, 0.00483
 485.35, 0.00907
 492.69, 0.01585
 502.75, 0.02602
 515.04, 0.04048
 530.09, 0.06018
**"""

    elif MATERIAL == "ALUMINIUM":
         P6 = f"""** MATERIALS
**
*Material, name={MATERIAL}
*Elastic
 60073.5, 0.33
*Plastic
 78.0955,          0.
 78.6679, 4.87013e-06
 83.0176, 3.47809e-05
   83.82, 4.02475e-05
  84.956, 6.56712e-05
  87.421, 0.000127628
 89.8845, 0.000221616
 92.0401, 0.000375636
 94.2774, 0.000633782
 96.5057,  0.00100217
 98.7358,  0.00155987
 100.966,  0.00227367
 103.195,  0.00309864
 105.425,   0.0039664
**"""
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define boundary conditions
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
    P7 = """\n**BOUNDARY CONDITIONS
**
** Name: Disp-BC-1 Type: Displacement/Rotation
*Boundary
```

```
XL, 2, 2
** Name: Disp-BC-2 Type: Displacement/Rotation
*Boundary
XR, 2, 2
** Name: Disp-BC-3 Type: Displacement/Rotation
*Boundary
YB, 1, 1
** Name: Disp-BC-4 Type: Displacement/Rotation
*Boundary
YT, 1, 1
** Name: Disp-BC-5 Type: Displacement/Rotation
*Boundary
X_L_EDGE, 3, 3
** Name: Disp-BC-6 Type: Displacement/Rotation
*Boundary
X_R_EDGE, 3, 3
** Name: Disp-BC-7 Type: Displacement/Rotation
*Boundary
Y_T_EDGE, 3, 3
** Name: Disp-BC-8 Type: Displacement/Rotation
*Boundary
Y_B_EDGE, 3, 3
**
   ----------------------------------------------------------------

"""

    if Linear_Input_File:

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Define linear set-up
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        P1_L ="*Heading\n** Job name: "+str(FIL_NAME)+"_L Model
           name: "+str(FIL_NAME)+"_L"
        P1_L +="\n** Generated by: Abaqus/CAE 2022\n*Preprint,
           echo=NO, model=NO, history=NO, contact=NO\n**"
        P2 = "\n** PARTS\n**\n*Part, name=PART-1\n*Node\n"

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Define linear step
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        P8_L = """**
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=NO, perturbation
```

```
*Buckle
10, , 18, 300
**
"""
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Define linear loading
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        P9_L = "**LOADS\n**\n**Name: LEFTLOAD Type: Concentrated
           force\n*Cload\n"
        P9_L += "loaded_left, 1, "+str(1)
        P9_L +="\n"
        P9_L += "**Name: RIGHTLOAD Type: Concentrated force\n*
           Cload\n"
        P9_L += "loaded_right, 1, "+str(-1)
        P9_L +="\n**"

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Linear output
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        P10_L = """\n** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
*
*Output, field, variable=PRESELECT
*NODE FILE, GLOBAL=NO,
MODE = 1, LAST MODE=5
U
*End Step
"""
        with open(OUT_FIL_NAME_L,"w+") as f:
            f.writelines(P1_L)
            f.writelines(P2)
            f.write(EXTRACTOR(NODES))
            f.write(P3)
            f.write(EXTRACTOR(ELEMENTS))
            f.write(P4)
            f.write(P5)
            f.write(P6)
            f.write(P7)
            f.write(P8_L)
            f.write(P9_L)
            f.write(P10_L)
```

```
    if Nonlinear_Input_File:

        if Eigenmode_Choice:
            EIG = int(input("F I R S T   P O S I T I V E   E I G
               E N M O D E  (G E O M E T R I C   I M P E R F E C T
                I O N) = "))
        else:
            EIG = 1

        EIGENSCALE = min((XMAX-XMIN)/200,(YMAX-YMIN)/200) # SCALE
            IS ARBITRARY, CHANGE TO BE SOME FACTOR OF LENGTH OR
           WIDTH ACCORDING TO GUIDANCE

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        # Define nonlinear set-up
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        P1_NL ="*Heading\n** Job name: "+str(FIL_NAME)+"_NL Model
            name: "+str(FIL_NAME)+"_NL"
        P1_NL +="\n** Generated by: Abaqus/CAE 2022\n*Preprint,
           echo=NO, model=NO, history=NO, contact=NO\n**"

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        # Define nonlinear imperfection
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        PI = "\n*IMPERFECTION, FILE= "+FIL_NAME+"_L, STEP=1, nset
           =WHOLE_GEOMETRY\n"+str(EIG)+","+str(EIGENSCALE)

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        # Define nonlinear solution control
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - #
        if CONTROL == "L":
            P8_NL = "**\n** STEP: Step-1\n**\n*Step, name=Step-1,
                nlgeom=YES, inc="+str(inc)+"\n*Static, riks\n"+str
               (start_ss)+","+str(arc)+","+str(min_ss)+","+str(
               max_ss)+","+str(lpf)+","+"\n**\n"

        elif CONTROL == "D":
            P8_NL = "**\n** STEP: Step-1\n**\n*Step, name=Step-1,
                nlgeom=YES, inc="+str(inc)+"\n*Static, riks\n"+str
               (start_ss)+","+str(arc)+","+str(min_ss)+","+str(
               max_ss)+","+str(lpf)+", loaded_left, 1, "+str(
               disp_max)+"\n**\n"
```

```
        elif CONTROL == "NO":
            P8_NL = "**\n** STEP: Step-1\n**\n*Step, name=Step-1,
                nlgeom=YES, inc="+str(inc)+"\n*Static\n"+str(
               start_ss_nr)+","+str(time_nr)+","+str(min_ss_nr)
               +","+str(max_ss_nr)+"\n**\n"

        elif CONTROL == "ND":
            P8_NL = "**\n** STEP: Step-1\n**\n*Step, name=Step-1,
                nlgeom=YES, inc="+str(inc)+"\n*Static, stabilize,
               allsdtol="+str(accuracy_tolerance)+"\n"+str(
               start_ss_nr)+","+str(time_nr)+","+str(min_ss_nr)
               +","+str(max_ss_nr)+"\n**\n"

        elif CONTROL == "NT":

            P8_NL = """**\n** STEP: Step-1
**
*Step, name=Step-1, nlgeom=YES, inc=175
*Static, stabilize, allsdtol=0.01
*Controls, parameters=Time Incrementation
** Relax checks on rate of convergence:
** IO   IR   IP   IC   IL   IG
   16 , 18 , 20 , 40 , 30 ,  6
0.001,1,1e-06,0.01
**\n"""

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Define nonlinear loading
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        ELASTIC_CAPACITY = float(input("E L A S T I C   C A P A C
            I T Y   F R O M   L I N E A R   P E R T U R B A T I O
           N   A S S E S S M E N T   (N) = "))
        if CONTROL == "L":

            P9_NL = "**LOADS\n**\n**Name: LEFTLOAD Type:
               Concentrated force\n*Cload\n"
            P9_NL += "loaded_left, 1, "+str(ELASTIC_CAPACITY)
            #P9 += "\nloaded_left, 2,"+str(ELASTIC_CAPACITY*
               y_load_mag)
            #P9+="\nloaded_left,3,"+str(ELASTIC_CAPACITY*
               z_load_mag)
            P9_NL +="\n"
            P9_NL += "**Name: RIGHTLOAD Type: Concentrated force\
               n*Cload\n"
            P9_NL += "loaded_right, 1, "+str(-1*ELASTIC_CAPACITY)
            #P9 += "\nloaded_right, 2,"+str(-1*ELASTIC_CAPACITY*
               y_load_mag)
```

```
            #P9+="\nloaded_right,3,"+str(-1*ELASTIC_CAPACITY*
               z_load_mag)+"\n**"
            P9_NL +="\n**"

        elif CONTROL == "D":

            P9_NL = "**LOADS\n**\n**Name: BC-9 Type: Displacement
               /Rotation\n*Boundary\n"
            P9_NL += "loaded_left, 1, 1, "+str(ELASTIC_CAPACITY
               /100)
            P9_NL +="\n"
            P9_NL += "**Name: BC-10 Type: Displacement/Rotation\n
               *Boundary\n"
            P9_NL += "loaded_right, 1, 1, "+str(-1*
               ELASTIC_CAPACITY/100)
            P9_NL +="\n**"

        elif CONTROL == "NO" or CONTROL == "ND" :

            P9_NL = "**LOADS\n**\n**Name: BC-9 Type: Displacement
               /Rotation\n*Boundary\n"
            P9_NL += "loaded_left, 1, 1, "+str(nr_disp)
            P9_NL +="\n"
            P9_NL += "**Name: BC-10 Type: Displacement/Rotation\n
               *Boundary\n"
            P9_NL += "loaded_right, 1, 1, "+str(-1*nr_disp)
            P9_NL +="\n**"

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # Nonlinear Output
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        P10_NL = """\n** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
CF, RF, U
*Element Output, directions=YES
LE, PE, PEEQ, PEMAG, S, SF
*Contact Output
CDISP, CSTRESS
**
** HISTORY OUTPUT: H-Output-1
**
```

```
*Output, history
*Node Output, nset=loaded_left
CF1, CF2, CF3, CM1, CM2, CM3, RF1, RF2
RF3, RM1, RM2, RM3, U1, U2, U3, UR1
UR2, UR3
*End Step
"""
        with open(OUT_FIL_NAME_NL,"w+") as f:
            f.writelines(P1_NL)
            f.writelines(P2)
            f.write(EXTRACTOR(NODES))
            f.write(P3)
            f.write(EXTRACTOR(ELEMENTS))
            f.write(P4)
            f.write(P5)
            f.write(P6)
            f.write(PI)
            f.write(P7)
            f.write(P8_NL)
            f.write(P9_NL)
            f.write(P10_NL)
```

C

DIC Data ABAQUS Input File Creator

```
# -*- coding: utf-8 -*-

# . . . . . .
    . . . . . #

                    #       S Y N O P S I S       #

# . . . . . .
    . . . . . #

# This code references the same EXCEL workbook considered by
   A2_Geometric_Imperfection

# Ensure that the NODES sheet is ordered  in terms of increasing
   x co-ordinate

# Cull points with particularly high errors (noise)

# Open the Excel and create two new sheets:

    # PLANE >> copy the index, x, y, z_outs columns (rename
       latter as 'z') from NODES sheet

        # IMPERFECTION >> copy the index, x, y, z columns from
           the NODES sheet

# The co-ordinates corresponding to each edge are extracted. This
    is important as the edges are typically 'jagged' as they are
   from the DIC data (and do not lie along a strict line)
```

```
# The list of DIC nodes isn't in a straightforward arrangement,
   and the spacing and angle between nodes may vary

# So the mesh is built up from the DIC data by assessing a
   vertical 'strip' of points at a time

# The mesh is built up from left to right and each strip of co-
   ordinates is appended into a line from L0 to Ln, the right-most
    strip (list_dict_A)

# Want to create a mesh of elements that aren't greatly distorted
   . Therefore sort through this list of strips to determine valid
    co-ordinates relative to a 'target' list of co-ordinates

# The elements are saved to the input Excel file, to be used
   later.

# Must ensure that the force is in the direction of the plane of
   the plate, therefore find the vector from the centre of the
   right edge to the centre of the left edge.

# The imperfection data is saved onto the IMPERFECTION sheet.
   This is used to inform .inp model using the command '
   IMPERFECTION, SYSTEM=R' (see PI)

# More info on this command can be found here: https://classes.
   engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/
   books/key/default.htm?startat=ch09abk03.html

# The .inp file is created similarly to
   A1_Linear_And_Nonlinear_Mesher. This reads the ELEMENT and
   PLANE sheets from the excel file (S4 elements are used)

# See the 'user defined values' section - note that the user must
    change the 'thickness' value on line 115 which is what is
   measured from the buckling tests.

# Similarly, the material properties are measured from the
   tensile test, and converted into true strain-stress (see P6)

# To mimic the true test set-up, loading is only applied on the
   right-hand edge, and the left-hand edge is fixed in-plane (see
   P7)

# Here, the Newton-Raphson method is used, with no damping. The
   solution controls are hard-coded (see P8)

# Displacement control is used. A maximum displacement of 20mm is
```

```
    considered, broken into components as determined by the '
   direction' of the force (see P9)

# As seen in A1_Linear_And_Nonlinear_Mesher, this will output a .
   inp file that can be sumbitted using the ABAQUS command window

# . . . . . .
    . . . . . . #

            #       I M P O R T    L I B R A R I E S        #

# . . . . . .
    . . . . . . #
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from math import sqrt

# . . . . . .
    . . . . . . #

            #       H O M E - M A D E    F U N C T I O N S
                      #

# . . . . . .
    . . . . . . #

def DIRECTION(p1,p2):
    A = np.array(p1).reshape(3,1)
    B = np.array(p2).reshape(3,1)
    C = B-A
    mag = sqrt((C[0])**2+(C[1])**2+(C[2])**2)
    D = np.array([C[0]/mag,C[1]/mag,C[2]/mag])
    return (D)

# Functions used for writing the .inp files
def EXTRACTOR(NAME):
    data_store = ""
    with open(NAME) as fp:
        data = ""
        data = fp.read()
        data_store+=data
    return(data_store)

def LIST_WRITER(NAME):
    with open("C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
       COMPUTATIONAL/RIKS_PACKAGE/TEMPORARY_FILE.txt","w+") as f:
        count = 1
```

```
        print(len(NAME))
        for i in NAME:
            if NAME.index(i)==0 and count==1:
                f.write(str(i))
            elif NAME.index(i)!=0 and (count-1)%16!=0 :
                f.write(","+str(i))
            elif NAME.index(i)!=0 and (count-1)%16 ==0:
                f.write(str(i))
            if count%16==0 and i!=NAME[-1]:
                f.write("\n")
            count+=1
            #print(i)
    return("C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
       COMPUTATIONAL/RIKS_PACKAGE/TEMPORARY_FILE.txt")

# . . . . . .
    . . . . . #

            #       U S E R   D E F I N E D   V A L U E S       #

# . . . . . .
    . . . . . #
# Manually change the thickness according to value measured
thickness = 0.896667

DIR = "C:/Users/pemb6626/OneDrive - Nexus365/Documents/YEAR2/
   WP1_PLATES/EXPERIMENTS_IMPERFECTION/MESH_DATA/"
EXT = "A3_Mesh_From_DIC_Example_Data"
FIL = DIR+EXT+".xlsx"
OUT_FIL_NAME = DIR+EXT+".inp"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Modify these if error before 'B U I L D    M E S H' section
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
node_spacing = 0.55
node_spacing2 = 0.4

# . . . . . .
    . . . . . #

                        #       D A T A       #

# . . . . . .
    . . . . . #

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
```

```
# Read the plane data
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
sheetname = "PLANE"
DATA = pd.read_excel(FIL,sheet_name=sheetname,index_col="index")

xs_out = DATA.loc[:,'x'].tolist()
ys_out = DATA.loc[:,'y'].tolist()
zs_out = DATA.loc[:,'z'].tolist()

fig, ax = plt.subplots()
ax.scatter(xs_out,ys_out)

# . . . . . .
    . . . . . #

            #       O R G A N I S E    T H E    D A T A       #

# . . . . . .
    . . . . . #

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Extract co-ordinated corresponding to each edge
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# LEFT HAND EDGE
edge_data = DATA.sort_values(by=["x"],ignore_index=False)[:500]
edge_data_sorted = edge_data.sort_values(by=["y"],axis=0,
   ascending=True)
nr, nc = edge_data_sorted.shape

i = 0

start_frame = pd.DataFrame({"x":[0],"y":[0]})
while i <= nr:
    indis = []
    for j in edge_data_sorted.loc[:,'y'].tolist():
        if int(j) == i:
            indis.append(edge_data_sorted.index[edge_data_sorted
               ["y"]==j][0])

    print(i,indis)
    if len(indis)>0:
        temp_data_frame = edge_data_sorted.loc[indis].copy()
        drops = []
        nrt, nct = temp_data_frame.shape
        minx = min(temp_data_frame.loc[:,'x'])
        for n in temp_data_frame.loc[:,'x'].tolist():
```

```
            if n > minx +0.1:
                drops.append(temp_data_frame.index[
                   temp_data_frame["x"]==n][0])

        temp_data_frame_out = temp_data_frame.drop(drops)

        start_frame = pd.concat([start_frame, temp_data_frame_out
           ])

    i+=1

# GETS RID OF THE INITIAL CONDITION HAD TO DO TO CREATE THE
   DATAFRAME
L0 = start_frame.loc[~((start_frame["x"]==0)&(start_frame["y
   "]==0))]
ax.scatter(L0.loc[:,'x'],L0.loc[:,'y'],color="r")
##########################################################################

# RIGHT HAND EDGE
edge_data = DATA.sort_values(by=["x"],ascending = False,
   ignore_index=False)[:500]
edge_data_sorted = edge_data.sort_values(by=["y"],axis=0,
   ascending=True)
nr, nc = edge_data_sorted.shape

i = 0

start_frame = pd.DataFrame({"x":[0],"y":[0]})
while i <= nr:
    indis = []
    for j in edge_data_sorted.loc[:,'y'].tolist():
        if int(j) == i:
            # Need to know the indices
            indis.append(edge_data_sorted.index[edge_data_sorted
               ["y"]==j][0])

    print(i,indis)
    if len(indis)>0:
        temp_data_frame = edge_data_sorted.loc[indis].copy()
        drops = []
        nrt, nct = temp_data_frame.shape
        maxx = max(temp_data_frame.loc[:,'x'])
        for n in temp_data_frame.loc[:,'x'].tolist():
            if n < maxx - 0.1:
                drops.append(temp_data_frame.index[
                   temp_data_frame["x"]==n][0])

        temp_data_frame_out = temp_data_frame.drop(drops)
```

```
        start_frame = pd.concat([start_frame, temp_data_frame_out
            ])

    i+=1

# GETS RID OF THE INITIAL CONDITION HAD TO DO TO CREATE THE
    DATAFRAME
Ln = start_frame.loc[~((start_frame["x"]==0)&(start_frame["y
    "]==0))]
ax.scatter(Ln.loc[:,'x'],Ln.loc[:,'y'],color="r")
nlr, nlc = Ln.shape
################################################################################

# BOTTOM EDGE

edge_data = DATA.sort_values(by=["y"],ignore_index=False)[:3000]
edge_data_sorted = edge_data.sort_values(by=["x"],axis=0,
    ascending=True)
nr, nc = edge_data_sorted.shape

i = 0

start_frame = pd.DataFrame({"x":[1000],"y":[1000]})
while i <= nr:
    indis = []
    for j in edge_data_sorted.loc[:,'x'].tolist():
        if int(j) == i:
            # Need to know the indices
            indis.append(edge_data_sorted.index[edge_data_sorted
                ["x"]==j][0])

    print(i,indis)
    if len(indis)>0:
        temp_data_frame = edge_data_sorted.loc[indis].copy()
        drops = []
        nrt, nct = temp_data_frame.shape
        miny = min(temp_data_frame.loc[:,'y'])
        for n in temp_data_frame.loc[:,'y'].tolist():
            if n > miny +0.1:
                drops.append(temp_data_frame.index[
                    temp_data_frame["y"]==n][0])

        temp_data_frame_out = temp_data_frame.drop(drops)

        start_frame = pd.concat([start_frame, temp_data_frame_out
            ])

    i+=1
```

```
# GETS RID OF THE INITIAL CONDITION HAD TO DO TO CREATE THE
   DATAFRAME
LB = start_frame.loc[~((start_frame["x"]==1000)&(start_frame["y
   "]==1000))]
ax.scatter(LB.loc[:,'x'],LB.loc[:,'y'],color="b")
nr, nc = LB.shape
number_in_length = nr
##########################################################################

# TOP EDGE

edge_data = DATA.sort_values(by=["y"],ignore_index=False)[-2000:]
edge_data_sorted = edge_data.sort_values(by=["x"],axis=0,
   ascending=True)
nr, nc = edge_data_sorted.shape

i = 0

start_frame = pd.DataFrame({"x":[1000],"y":[1000]})
while i <= nr:
    indis = []
    for j in edge_data_sorted.loc[:,'x'].tolist():
        if int(j) == i:
            # Need to know the indices
            indis.append(edge_data_sorted.index[edge_data_sorted
               ["x"]==j][0])

    print(i,indis)
    if len(indis)>0:
        temp_data_frame = edge_data_sorted.loc[indis].copy()
        drops = []
        nrt, nct = temp_data_frame.shape
        maxy = max(temp_data_frame.loc[:,'y'])
        for n in temp_data_frame.loc[:,'y'].tolist():
            if n < maxy - 0.1:
                drops.append(temp_data_frame.index[
                   temp_data_frame["y"]==n][0])

        temp_data_frame_out = temp_data_frame.drop(drops)

        start_frame = pd.concat([start_frame, temp_data_frame_out
           ])

    i+=1

# GETS RID OF THE INITIAL CONDITION HAD TO DO TO CREATE THE
   DATAFRAME
LT = start_frame.loc[~((start_frame["x"]==1000)&(start_frame["y
   "]==1000))]
```

```
ax.scatter(LT.loc[:,'x'],LT.loc[:,'y'],color="b")
##########################################################################

# SORTING INTO COLUMNS

list_dict_A = {'L0':L0}
list_names = ['L0']

ts = list(range(number_in_length))
final_new_frame = L0

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Create a working copy for interior mesh generation
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
DATA_COPY = DATA.sort_values(by=["x"],ignore_index=False)
nr, nc = DATA_COPY.shape
t = 0

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Build vertical strips from left to right until no points remain
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
while nr>0:
    t+=1
    lista = final_new_frame.loc[:,"x"].tolist()
    listb = final_new_frame.loc[:,"y"].tolist()

    xa = min(final_new_frame.loc[:,'x'].tolist())
    ya = min(final_new_frame.loc[:,'y'].tolist())

    xb = max(final_new_frame.loc[:,'x'].tolist())
    yb = max(final_new_frame.loc[:,'y'].tolist())

    drops = final_new_frame.index.values.tolist()# list of
       final_new_frame indicies
    DATA_COPY = DATA_COPY.drop(drops) # delete the origonal list
       from the list of points, so search from a reduced list
    nr, nc = DATA_COPY.shape
    print("Number of rows in copied frame = " + str(nr)) # to
       ensure dropping

    new_data = DATA_COPY[:250]
    mindistsx = []
    mindistsy = []

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
    - - - - - - #
# Determine distance from candidate points to current column
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
for i,j in zip(new_data.loc[:,"x"].tolist(),new_data.loc[:,"y
   "].tolist()):
    distsx = []
    distsy = []
    for x,y in zip(lista,listb):
        distsx.append(abs(x-i))
        distsy.append(abs(y-j))

    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - - - #
    # Identify the closest y co-ordinate
    # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        - - - - - - - - #
    mindistsy.append(min(distsy))
    mindistsx.append(distsx[distsy.index(min(distsy))])

DIST = pd.DataFrame({"x":new_data.loc[:,"x"].tolist(),"y":
   new_data.loc[:,"y"].tolist(),"dx":mindistsx,"dy":mindistsy
   })

if nr != 0:
    new_frame = pd.DataFrame({"x":[1000],"y":[1000]})

    x0 = min(DATA_COPY.loc[:,'x'].tolist())
    ix0 = DATA_COPY.index[(DATA_COPY["x"]== x0)]
    y0 = DATA_COPY.loc[ix0,'y'].tolist()[0]

    for (x,y) in zip(DATA_COPY.loc[:,'x'].tolist(),DATA_COPY.
       loc[:,'y'].tolist()):

        if x in new_data.loc[:,"x"].tolist():
            ind = DIST.index[DIST["x"]==x][0]

        xlim = (xb - xa)*(y - y0)/(yb - ya) + xa +
           node_spacing + 0.1

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - - #
        # Select next strip of nodes for analysis within
           spacing tolerance
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - - #
        if x<xlim or DIST.loc[ind,"dx"]<= node_spacing +
           node_spacing2:# and DIST.loc[ind,"dy"]<=
           node_spacing+ node_spacing_2):
```

```
                indi = DATA.index[(DATA["y"]== y) & (DATA["x"]==
                    x)]
                temp_data_frame_out = DATA.loc[indi].copy()
                new_frame = pd.concat([new_frame,
                    temp_data_frame_out])

        final_new_frame = new_frame.loc[~((new_frame["x"]==1000)
            &(new_frame["y"]==1000))]
        list_name = "L"+str(t)

        list_names.append(list_name)
        list_dict_A[list_name] = final_new_frame

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Plot last generated strip of points
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
ax.scatter(final_new_frame.loc[:,'x'],final_new_frame.loc[:,'y'],
    color="r")

# . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    . . . . . #

                #       B U I L D    M E S H       #

# . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    . . . . . #

N1 = []
N2 = []
N3 = []
N4 = []

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# There may be a 'jagged edge' due to skew
# Therefore, combine all y values in L0 (the left-hand edge) and
    Ln (the right-hand edge)
# Cull those that are very close
# The final list is a representative selection of y-values from
    both lists
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #

ALL_YS = pd.concat([L0, Ln]).sort_values(by=["y"],axis=0,
    ascending=True).loc[:,"y"].tolist()
nra = len(ALL_YS)
yn = 0
```

```
drops = []

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Delete y-values closer than node_spacing
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #

while yn + 1 < nra:
    if yn + 2 <nra:
        while ALL_YS[yn] + node_spacing > ALL_YS[yn+1]:
            del ALL_YS[yn+1]
            nra = len(ALL_YS)
    else:
        if ALL_YS[yn] + node_spacing > ALL_YS[yn+1]:
            del ALL_YS[yn+1]
    nra = len(ALL_YS)
    yn +=1

LY = ALL_YS
n = 0
while n+1 < len(LY):
    ytargeta = LY[n]
    ytargetb = LY[n+1]

    m = 0
    while m+1< len(list_dict_A):

        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #
        # DATA_A and DATA_B are adjacent 'strips'
        # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            - - - - - - - - - #

        DATA_A = list_dict_A[list_names[m]]
        DATA_B = list_dict_A[list_names[m+1]]

        nra, nca = DATA_A.shape
        nrb, ncb = DATA_B.shape

        lista = DATA_A.loc[:,"y"].tolist()
        listb = DATA_B.loc[:,"y"].tolist()
        listc = DATA_A.loc[:,"x"].tolist()
        listd = DATA_B.loc[:,"x"].tolist()

        dist1 = []
        dist2 = []
        dist3 = []
        dist4 = []
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - #
# Due to skew, the mesh is not perfectly aligned
# Nodes in adjacent 'strips' do not share identical y-
   values
# Some y-values may also be missing/extra
# Hence need to find the closest y co-ordinate to the
   target
# (i.e. the list of possible values from the combined
   left and right edge lists)
# This ensures that all nodes are aligned with
   neighbouring nodes and prevents highly irregular
   elements
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - #

for ya in lista:
    dist1.append(abs(ya-ytargeta))
    dist2.append(abs(ya-ytargetb))
for yb in listb:
    dist3.append(abs(yb-ytargeta))
    dist4.append(abs(yb-ytargetb))

y1 = lista[dist1.index(min(dist1))]
x1 = listc[dist1.index(min(dist1))]

y2 = lista[dist2.index(min(dist2))]
x2 = listc[dist2.index(min(dist2))]

y3 = listb[dist3.index(min(dist3))]
x3 = listd[dist3.index(min(dist3))]

y4 = listb[dist4.index(min(dist4))]
x4 = listd[dist4.index(min(dist4))]

if ytargeta - node_spacing<= y1 <= ytargeta +
   node_spacing and ytargeta - node_spacing<= y3 <=
   ytargeta + node_spacing and ytargetb - node_spacing<=
   y2 <= ytargetb + node_spacing and ytargetb -
   node_spacing<= y4 <= ytargetb + node_spacing:
    N1.append((DATA_A.index[(DATA_A["y"]== y1) & (DATA_A
       ["x"]== x1)])[0])
    N2.append((DATA_B.index[(DATA_B["y"]== y3) & (DATA_B
       ["x"]== x3)])[0])
    N3.append((DATA_B.index[(DATA_B["y"]== y4) & (DATA_B
       ["x"]== x4)])[0])
    N4.append((DATA_A.index[(DATA_A["y"]== y2) & (DATA_A
       ["x"]== x2)])[0])
```

```
        print(m,n)
        m+=1
    n +=1

DATA_ELES = {"N1":N1,"N2":N2,"N3":N3,"N4":N4}
D_E_DF = pd.DataFrame(DATA_ELES)

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Remove erroneous elements
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
D_NEW = (D_E_DF.index[(D_E_DF["N1"] == D_E_DF["N2"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)
D_NEW = (D_E_DF.index[(D_E_DF["N1"] == D_E_DF["N3"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)
D_NEW = (D_E_DF.index[(D_E_DF["N1"] == D_E_DF["N4"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)
D_NEW = (D_E_DF.index[(D_E_DF["N2"] == D_E_DF["N3"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)
D_NEW = (D_E_DF.index[(D_E_DF["N2"] == D_E_DF["N4"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)
D_NEW = (D_E_DF.index[(D_E_DF["N3"] == D_E_DF["N4"])]).tolist()
D_E_DF.drop(D_NEW, axis=0, inplace=True)

D_E_2_r1 = np.array((D_E_DF).iloc[:,0])
D_E_2_r2 = np.array((D_E_DF).iloc[:,1])
D_E_2_r3 = np.array((D_E_DF).iloc[:,2])
D_E_2_r4 = np.array((D_E_DF).iloc[:,3])

newindex = list(range(1,len(D_E_2_r1)+1))
DATA_ELES = {"N1":D_E_2_r1,"N2":D_E_2_r2,"N3":D_E_2_r3,"N4":
   D_E_2_r4}

D_E_DF = pd.DataFrame(DATA_ELES,index=newindex)
er, ec = D_E_DF.shape

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Save elements to a new sheet
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
with pd.ExcelWriter(FIL,mode='a') as writer:
    D_E_DF.to_excel(writer,"ELEMENTS")

# . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    . . . . . . #

                #       F O R C E    D I R E C T I O N       #
```

```
# . . . . . .
    . . . . . #

XL = list(L0.index.values)
XL_MID = XL[int(len(XL)/2)]
PL_MID = L0.loc[XL_MID,:]

XR = list(Ln.index.values)
XR_MID = XR[int(len(XR)/2)]
PR_MID = Ln.loc[XR_MID,:]

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Direction of force, save to the Excel file
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
direction = DIRECTION(PR_MID,PL_MID)
out = pd.DataFrame({"x":direction[0],"y":direction[1],"z":
   direction[2]})
#with pd.ExcelWriter(FIL,mode='a') as writer:
#    out.to_excel(writer,"FORCE")

# . . . . . .
    . . . . . #

            #       I M P E R F E C T I O N    D A T A       #

# . . . . . .
    . . . . . #

DATAI = pd.read_excel(FIL,sheet_name="IMPERFECTION",index_col="
   index")

node_spacing = 0.55

# CANNOT BREAK DOWN BY SKIPPING TWO ELEMENTS AS THEN IT GOES ALL
   HIGGLEDY PIGGLEDY
xs_out = DATAI.loc[:,'x'].tolist()
ys_out = DATAI.loc[:,'y'].tolist()
zs_out = DATAI.loc[:,'z'].tolist()

lendat = len(xs_out)
inds = list(range(1,lendat+1))
outs = []
for i in inds:
    out = "Assembly.PART-1-1."+str(i)
    outs.append(out)
```

```
NEW_DATA = {"a":outs,"x":xs_out,"y":ys_out,"z":zs_out}
D_NEW_DF = pd.DataFrame(NEW_DATA)

outdir = DIR + EXT + "_IMPERFECTION.txt"
D_NEW_DF.to_csv(outdir,header=False,index=False)

# . . . . . .
    . . . . . #

          #       B U I L D   T H E   I N P U T   F I L E
             #

# . . . . . .
    . . . . . #
D_N_DF = pd.read_excel(FIL,sheet_name="PLANE")
D_N_DF_inds = pd.read_excel(FIL,sheet_name="PLANE",index_col=0)
nr, nc = D_N_DF.shape
number_of_nodes=nr

D_E_DF = pd.read_excel(FIL,sheet_name='ELEMENTS')
D_E_DF_inds = pd.read_excel(FIL,sheet_name='ELEMENTS',index_col
   =0)
nr, nc = D_E_DF.shape
number_of_elements=nr

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Vectorised merge of values in L0, Ln, LB and LT into the node
   sets
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
X_L_NSET = (L0.merge(D_N_DF_inds.reset_index(),on=["x", "y"],how
   ="left")["index"]).tolist()
X_R_NSET = (Ln.merge(D_N_DF_inds.reset_index(),on=["x", "y"],how
   ="left")["index"]).tolist()

Y_T_NSET=(LT.merge(D_N_DF_inds.reset_index(),on=["x", "y"],how="
   left")["index"]).tolist()
Y_B_NSET=(LB.merge(D_N_DF_inds.reset_index(),on=["x", "y"],how="
   left")["index"]).tolist()

X_L_MID = X_L_NSET[int(len(X_L_NSET)/2)]
X_R_MID = X_R_NSET[int(len(X_R_NSET)/2)]

Y_B_MID = Y_B_NSET[int(len(Y_B_NSET)/2)]
Y_T_MID = Y_T_NSET[int(len(Y_T_NSET)/2)]

# SETS
# LOADED POINT (MAKE IN LINE WITH MID-POINT)
```

```
co_ord_x_load_right = PR_MID["x"] + 10
co_ord_y_load_right = PR_MID["y"]
co_ord_z_load_right = PR_MID["z"]


# FIRST LINES FOR THE HEADING AND JOB NAMES ETC
D_N_DF.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES.txt",
   header=False,index=False)
D_E_DF.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_ELEMENTS.
   txt",header=False,index=False)

D_N_DF_inds.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES
   /COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES_INDS
   .txt",header=False,index=False)
D_E_DF_inds.to_csv(r"C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES
   /COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS_INDS.txt",header=False,index=False)

NODES = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES.txt"
ELEMENTS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_ELEMENTS.
   txt"

NODES_INDS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/OUTPUT_ABAQUS_NODES_INDS.
   txt"
ELEMENTS_INDS = "C:/Users/pemb6626/Documents/YEAR2/WP1_PLATES/
   COMPUTATIONAL/LIBRARY_TOOLKIT/CODE/SS/
   OUTPUT_ABAQUS_ELEMENTS_INDS.txt"

WRITE_FILE = "10_06_PRACTICE"
READ_FILE = [NODES, ELEMENTS]

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define nonlinear set-up
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P1 ="*Heading\n** Job name: "+str(EXT)+"_NL Model name: "+str(EXT
   )+"_NL"
P1 +="\n** Generated by: Abaqus/CAE 2022\n*Preprint, echo=NO,
   model=NO, history=NO, contact=NO\n**"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Create Part
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P2 = "\n** PARTS\n**\n*Part, name=PART-1\n*Node\n"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Create Elements
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P3 = "*Element, type=S4\n"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define element set for whole geometry and set material
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P4 = "*Elset, elset=WHOLE_GEOMETRY, generate\n1,"+str(
   number_of_elements)+",1"
P4 += F"\n** Section: Section-1-WHOLE_GEOMETRY\n*Shell Section,
   elset=WHOLE_GEOMETRY, material=Aluminium\n"
P4 += str(thickness)+", 5\n*End Part\n**\n**\n"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define assembly and loaded nodes
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P5 ="** ASSEMBLY\n**\n*Assembly, name=Assembly\n**\n*Instance,
   name=PART-1-1, part=PART-1\n*End Instance\n**"
P5 += "\n*Node\n1,"+str(co_ord_x_load_right)+","+str(
   co_ord_y_load_right)+","+str(co_ord_z_load_right)
P5 += "\n*Nset, nset=loaded\n1,"
P5 += "\n*Nset, nset=WHOLE_GEOMETRY, instance=PART-1-1, generate\
   n1," +str(number_of_nodes)+",1"
P5 += "\n*Nset, nset=XL, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER([X_L_MID]))+","
P5 += "\n*Nset, nset=XR, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER([X_R_MID]))+","
P5 += "\n*Nset, nset=YB, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER([Y_B_MID]))+","
P5 += "\n*Nset, nset=YT, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER([Y_T_MID]))+","
P5 += "\n*Nset, nset=X_L_EDGE, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER(X_L_NSET))
P5 += "\n*Nset, nset=X_R_EDGE, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER(X_R_NSET))
P5 += "\n*Nset, nset=Y_B_EDGE, instance=PART-1-1\n"+EXTRACTOR(
   LIST_WRITER(Y_B_NSET))
P5 += "\n*Nset, nset=Y_T_EDGE, instance=PART-1-1\n"+EXTRACTOR(
```

```
   LIST_WRITER(Y_T_NSET))
P5 += "\n*Surface, type=NODE, name=X_R_EDGE_CNS_, internal\
   nX_R_EDGE, 1."
P5 +="\n** Constraint: Constraint-1\n*Coupling, constraint name=
   Constraint-1, ref node=loaded, surface=X_R_EDGE_CNS_\n*
   Distributing, weighting method=UNIFORM\n*End Assembly\n**\n"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define material properties
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P6 = f"""** MATERIALS
**
*Material, name=Aluminium
*Elastic
 60073.5, 0.33
*Plastic
78.1940198843,0.0
79.6613562487,2.796891362299995e-05
81.2350799302,6.0642284141e-05
82.844683037,9.897870598700001e-05
84.4244305842,0.000144095466931
85.966781042,0.00019904471613300001
87.5029654442,0.000268136416818
89.0508683222,0.000350789028487
90.600364695,0.00043759851280599995
92.1474576945,0.000540417624247999 9
93.6951629925,0.000673509455851
95.2427901504,0.0008434682501439999
96.7904365378,0.001068058555958
98.33807901590001,0.001325819874558
99.8857216869,0.001642928445018
101.433364369,0.002054626081468
102.981007016,0.0025868334405479997
104.528649681,0.0030630848009379997
106.076292326,0.0035964618567979998
107.623934932,0.004232752135188
109.171577566,0.004800359154168
110.719220189,0.005397693904838
112.266862768,0.006100070118398
113.814505362,0.006765008647398
115.362147968,0.007405577694538
116.909790541,0.008119009138868
118.457433115,0.008830538849248
120.005075685,0.009553031333358001
121.552718251,0.010281995253338
123.100360771,0.011102233830038001
124.648003324,0.011857025942038
```

```
126.19564586,0.012648575279538
127.743288423,0.013379456274338
129.290931065,0.013924660105938
130.838573625,0.014667526753338002
132.386216138,0.015502739008138
133.933858661,0.016317907389738
135.48150119,0.017122875924938
137.02914371,0.017946260972938
138.576786242,0.018745268408838
140.124428756,0.019577917060837998
141.672071255,0.020438042475537998
143.21971377,0.021268132773138
144.767356219,0.022215773798137998
146.314998636,0.023211037401137998
147.862641107,0.024120199800238
149.410283545,0.025084535412538
150.957925967,0.026073207428837997
152.505568409,0.027028981353737998
154.053210823,0.028028462449637997
155.600853214,0.029060354555637997
157.14849557,0.030146134822437998
158.696137919,0.031244545409737996
160.243780196,0.032436895747338004
161.791422545,0.033532104830238
163.339064879,0.034649325127038
164.886707138,0.035866400954438
166.434349379,0.037104213133038
167.9819915,0.038485853315138
169.529633716,0.039753499810038
171.077275927,0.041030735925538
172.624917987,0.042471031003138
174.172560005,0.043958825064438004
175.720202036,0.045434127299638
177.267844166,0.046797158759238004
178.815486231,0.048235373979838
180.363128109,0.049860963704138
181.910769909,0.051537018068638
183.458411692,0.053243700621338
185.006053584,0.054853882507438
186.553695417,0.056521324528038
188.101337087,0.058329304850138
189.648978666,0.060217522666738
191.196620179,0.062136623147038
192.744261314,0.06432580998413799
194.291902329,0.066616520626038
195.839543638,0.06870190001493799
197.387184484,0.071075067333538
198.934825611,0.073290675569838
200.482466314,0.075702065854938
```

```
202.030106767,0.07832739720833799
203.577746785,0.081170216900238
205.125387606,0.083590335976738
206.673028241,0.086073424169238
208.220668807,0.088566093362638
209.768309444,0.091061476066838
211.315947515,0.094662187638138
212.863586088,0.098091921146738
214.411224549,0.101571985540338
215.958864047,0.104608789553338
217.506499996,0.108793827457338
219.054136283,0.113011848780338
220.60177335,0.116946700754338
222.149410812,0.12070003600033799
223.697046559,0.125146112432338
225.24467775,0.130914163538338
**"""
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define imperfection
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
PI = "\n*IMPERFECTION, SYSTEM=R\n"
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define boundary conditions
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P7 = """**BOUNDARY CONDITIONS
**
** Name: Disp-BC-1 Type: Displacement/Rotation
*Boundary
XL, 2, 2
** Name: Disp-BC-2 Type: Displacement/Rotation
*Boundary
X_L_EDGE, 1, 1
** Name: Disp-BC-3 Type: Displacement/Rotation
*Boundary
X_L_EDGE, 3, 3
** Name: Disp-BC-4 Type: Displacement/Rotation
*Boundary
X_R_EDGE, 3, 3
** Name: Disp-BC-5 Type: Displacement/Rotation
*Boundary
Y_T_EDGE, 3, 3
** Name: Disp-BC-6 Type: Displacement/Rotation
*Boundary
Y_B_EDGE, 3, 3
**
```

```
   ------------------------------------------------------------------

"""
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define nonlinear solution control
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P8 = "**\n** STEP: Step-1\n**\n*Step, name=Step-1, nlgeom=YES,
   inc="+str(250)+"\n*Static\n"+str(0.001)+","+str(1)+","+str(1e
   -06)+","+str(0.15)+"\n**\n"
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Define displacement control
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P9 =f"""**LOADS
**
** Name: BC-7 Type: Displacement/Rotation
*Boundary
loaded, 1, 1, {20*direction[0][0]}
loaded, 2, 2, {20*direction[1][0]}
loaded, 3, 3, {20*direction[2][0]}
**
**
   ------------------------------------------------------------------

"""
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
# Nonlinear Output
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - #
P10 = """** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
CF, RF, U
*Element Output, directions=YES
LE, PE, PEEQ, PEMAG, S, SF
*Contact Output
CDISP, CSTRESS
**
** HISTORY OUTPUT: H-Output-1
**
```

```
*Output, history
*Node Output, nset=loaded
CF1, CF2, CF3, CM1, CM2, CM3, RF1, RF2
RF3, RM1, RM2, RM3, U1, U2, U3, UR1
UR2, UR3
*End Step
"""

with open(OUT_FIL_NAME,"w+") as f:
    f.writelines(P1)
    f.writelines(P2)
    f.write(EXTRACTOR(NODES))
    f.write(P3)
    f.write(EXTRACTOR(ELEMENTS))
    f.write(P4)
    f.write(P5)
    f.write(P6)
    f.write(PI)
    f.write(EXTRACTOR(DIR + EXT + "_IMPERFECTION.txt"))
    f.write(P7)
    f.write(P8)
    f.write(P9)
    f.write(P10)
```

D

Code to Measure Geometric Imperfection

```
# -*- coding: utf-8 -*-

# . . . . . .
    . . . . . #

                    #       S Y N O P S I S       #

# . . . . . .
    . . . . . #

# This code is used to extract geometric imperfection data from
   DIC data

# On the NODES sheet are x, y, z spatial co-ordinates of one of
   the initial photos extracted from DIC

# The NODES sheet data only informs of the position of the plate
   from the camera, it does not account for any skew of the plate
   in the 3D plane relative to the camera

# This can be visualised by setting STEP equal to 1

# The user creates the GUIDE sheet in the Excel file. This is
   used to list the position of three co-ordinates from the NODES
   sheet:

    # A >> A node in the bottom left corner

    # B >> A node in the top left corner
```

```
    # C >> A node at the centre of the right edge

# These occupy the first 3 rows and 5 columns of the guide sheet
   as follows:

    # (for example) A , NODES index, x, y, z

# Visualise this plane by setting STEP equal to 2

# The z co-ordinate for every x, y co-odinate in the NODES sheet
   is determined and output to a new sheet called PLANE_AMPS (set
   STEP = 3)

# Once complete, open the Excel and copy the 'zs_out' column from
    the PLANE_AMPS sheet onto the NODES sheet

# Create a new column called 'amp' which is the difference
   between 'zs_out' and the 'z' column. This is the geometric
   imperfection - the displacement about the central plane.

# Close the Excel file and set STEP = 4. May need to adjust minz
   and maxz as desired.

# Plot and save the result with STEP = 5, which uses the minz and
    maxz from STEP 4.

##############################################################################

                            # IMPORT FUNCTIONS #
##############################################################################

from openpyxl import load_workbook
from openpyxl.utils.dataframe import dataframe_to_rows

import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib import rcParams
from mpl_toolkits.axes_grid1.inset_locator import
   zoomed_inset_axes, mark_inset

import pandas as pd
import numpy as np
from math import log10, pi, sqrt, sin, cos

from scipy.interpolate import interp1d
from scipy.integrate import quad, dblquad
from scipy.misc import derivative
```

```
import pylab
from scipy.interpolate import griddata

import sympy as sm

##########################################################################

                                    # DATA #
##########################################################################

# SEE FOR SURFACE https://matplotlib.org/stable/gallery/mplot3d/
   contourf3d_2.html#sphx-glr-gallery-mplot3d-contourf3d-2-py
# SEE FOR EIGEN AMP SIZE https://classes.engineering.wustl.edu
   /2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?
   startat=pt04ch11s03aus55.html
#EXT = "15_05_TEST8"
#FIL = "C:/Users/eleno/OneDrive - Nexus365/Documents/YEAR2/
   WP1_PLATES/EXPERIMENTS_IMPERFECTION/TEST8_CASE_STUDY/"
TEST = "TEST1"

EXT = "17_05_TEST1_MESH_FINAL"
FIL = "C:/Users/pemb6626/OneDrive - Nexus365/Documents/YEAR2/
   WP1_PLATES/EXPERIMENTS_IMPERFECTION/MESH_DATA/"
DATA = pd.read_excel(FIL+EXT+".xlsx",sheet_name="NODES")

# STEP

# STEP = 1 TO SEE THE CO-ORDINATES PLOTTED
# STEP = 2 TO CREATE A PLANE
# STEP = 3 TO EXPORT THE POINTS ON THE PLANE TO SEE WHERE THE MAX
    AND MIN DIFFERENCE IS
# STEP = 4 TO DETERMINE THE COLOUR BAR SCALE AND VISUALISE
   IMPERFECTION
# STEP = 5 TO OUTPUT IMAGES

STEP = 5

##########################################################################


# CANNOT BREAK DOWN BY SKIPPING TWO ELEMENTS AS THEN IT GOES ALL
   HIGGLEDY PIGGLEDY
xs= DATA.loc[:,'x'].tolist()
ys= DATA.loc[:,'y'].tolist()
zs= DATA.loc[:,'z'].tolist()

fig, ax = plt.subplots()
ax = plt.figure().add_subplot(projection="3d")
tsfont = {"fontname":"Times New Roman"}
```

```
if STEP == 1:
    print("check1")
    ax.scatter(xs,ys,zs,c=zs,vmin=min(zs),vmax=max(zs),cmap="hsv
       ",alpha=0.4,s=1)
    ax.view_init(elev=35, azim=-45, roll=0)
    #ax.set_zticks([-5, 0, 5],fontsize=16)
    ax.set_xlabel("x",fontsize=16,**tsfont)
    ax.set_ylabel("y",fontsize=16,**tsfont)
    ax.set_zlabel("z",fontsize=16,**tsfont)
    ax.dist=15

    ax.set_aspect('equal', adjustable='box')
    ax.view_init(elev=90, azim=-90, roll=0)
    print("check2")


if 1< STEP< 4:
    DATA_p = pd.read_excel(FIL+EXT+".xlsx",sheet_name="GUIDE",
       header=None)

    #HAD TO SUBTRACT 1 FROM THE DESIRED NODE BECAUSE READING FROM
        LIST WHERE INDEX STARTS AT 0
    a = int(DATA_p.iloc[0,1]) - 1
    b = int(DATA_p.iloc[1,1]) - 1
    c = int(DATA_p.iloc[2,1]) - 1
    # DEFINE THREE POINTS
    A = np.array([xs[a],ys[a],zs[a]]).reshape(3,1)
    B = np.array([xs[b],ys[b],zs[b]]).reshape(3,1)
    C = np.array([xs[c],ys[c],zs[c]]).reshape(3,1)

    ax.scatter(A[0][0],A[1][0],A[2][0],color="k",s=100)
    ax.scatter(B[0][0],B[1][0],B[2][0],color="k",s=100)
    ax.scatter(C[0][0],C[1][0],C[2][0],color="k",s=100)

    p = (B-A).reshape(1,3)
    q = (C-A).reshape(1,3)
    r = np.cross(p,q).reshape(3,1)

    #print(r[1][0])
    x = np.linspace(0, 300)
    y = np.linspace(0, 120)

    x, y = np.meshgrid(x, y)
    z = +A[2][0] - (r[0][0]*(x-A[0][0]) + r[1][0]*(y-A[1][0]))/(r
       [2][0])
    ax.plot_surface(x, y, z)
    ax.scatter(xs,ys,zs,c=zs,vmin=min(zs),vmax=max(zs),cmap="hsv
       ",alpha=0.4,s=1)
```

```
    ax.view_init(elev=35, azim=-45, roll=0)
    #ax.view_init(elev=90, azim=-90, roll=0)

if STEP==3:

    zout = []
    for i in xs:
        z_point = +A[2][0] - (r[0][0]*(i-A[0][0]) + r[1][0]*(ys[
           xs.index(i)]-A[1][0]))/(r[2][0])
        zout.append(z_point)

    data_out = pd.DataFrame({"zs_out":zout})
    with pd.ExcelWriter(FIL+EXT+".xlsx",mode='a') as writer:
        data_out.to_excel(writer,"PLANE_AMPS")


if STEP ==4:

    # COLOURBAR SEE https://matplotlib.org/stable/users/explain/
       axes/colorbar_placement.html


    data_out = pd.read_excel(FIL+EXT+".xlsx",sheet_name="NODES")
    zout=data_out.loc[:,'amp'].tolist()

    minz = min(zout)
    maxz = max(zout)

    plotted = ax.scatter(xs,ys,zs,c=zout,vmin=minz,vmax=maxz,cmap
       ="jet",alpha=0.4,s=1)
    #plotted = ax.scatter(xs,ys,zs,c=zout,vmin=-0.05,vmax=0,cmap
       ="cool",alpha=0.4,s=2)
    ax = plt.gca()
    ax.set_aspect('equal', adjustable='box')
    #ax.set_axis_off()
    ax.view_init(elev=90, azim=-90, roll=0)
    cb = fig.colorbar(plotted,ax=ax,ticks=[round(minz,3),round(
       maxz,3),0],shrink=0.4,location = 'bottom',pad=0.005)
    #cb = fig.colorbar(plotted,ax=ax,ticks=[-0.05,0,0],shrink
       =0.4,location = 'bottom',pad=0.005)
    cb.ax.tick_params(labelsize=12)

    #ax.set_xlim(270,286)
    #ax.set_ylim(15,25)

if STEP == 5:

    FOLDER = "C:/Users/eleno/OneDrive - Nexus365/Documents/YEAR2/
       WP1_PLATES/EXPERIMENTS_IMPERFECTION/IMPERFECTION_IMAGES/
```

```
        FINAL/"
    # COLOURBAR SEE https://matplotlib.org/stable/users/explain/
        axes/colorbar_placement.html

    data_out = pd.read_excel(FIL+EXT+".xlsx",sheet_name="NODES")
    zout=data_out.loc[:,'amp'].tolist()

    minz = min(zout)
    maxz = max(zout)

    ##########################################################################

    plotted = ax.scatter(xs,ys,zs,c=zout,vmin=minz,vmax=maxz,cmap
        ="jet",alpha=0.4,s=1)

    # Create a grid
    xi = np.linspace(min(xs), max(xs), 80)
    yi = np.linspace(min(ys), max(ys), 80)
    X, Y = np.meshgrid(xi, yi)

    # FOR THE DIAGRAM TO SCALE
    #ax.set_aspect('equalxy', adjustable='box')
    ax.set_aspect('equal', adjustable='box')

    # SET THE VIEW OF THE PLATE
    ax.view_init(elev=-35, azim=-45, roll=15)
    #ax.view_init(elev=35, azim=-45, roll=15)
    #ax.view_init(elev=90, azim=-90, roll=0)

    ax.set_axis_off()

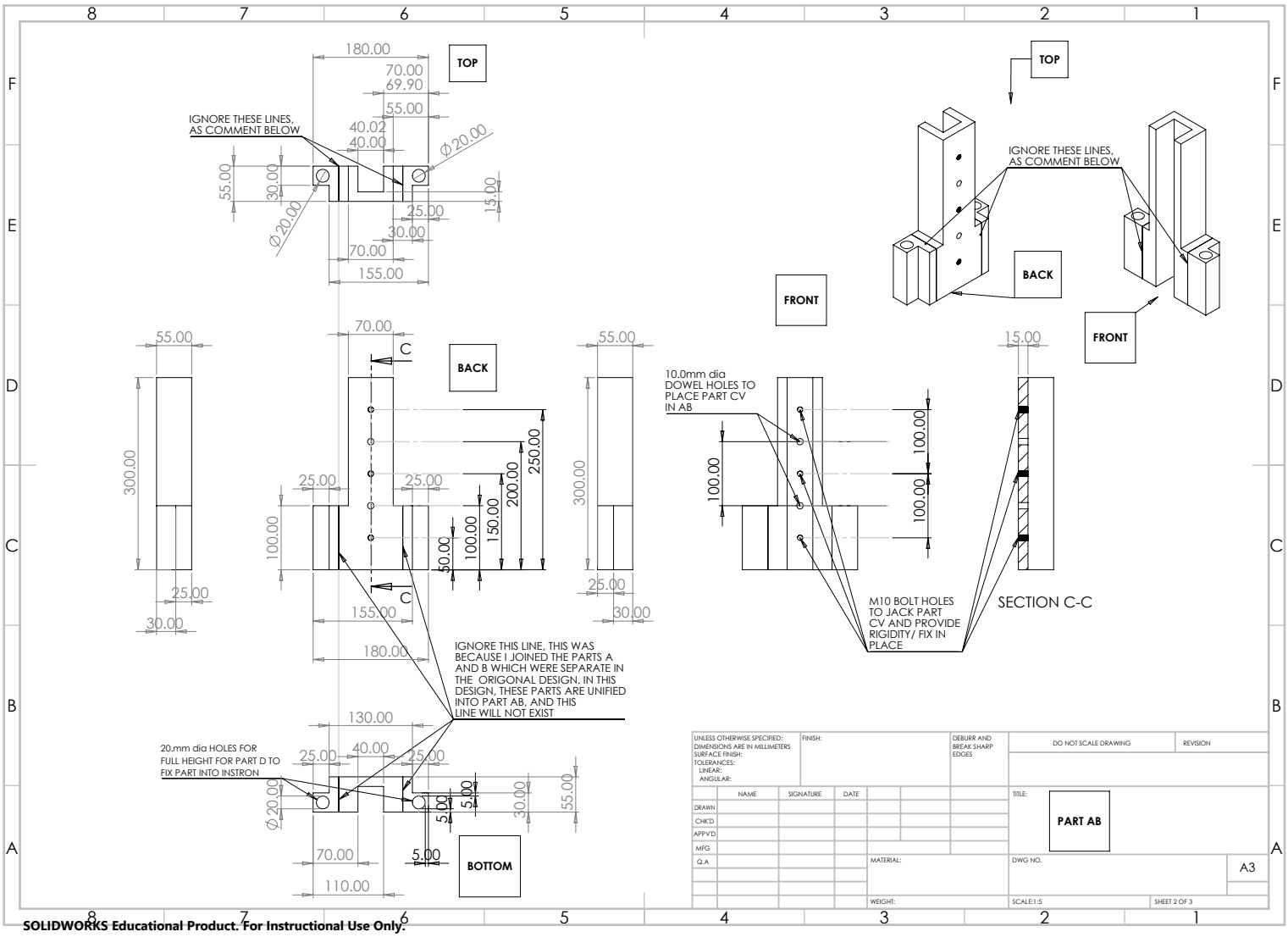
    plt.show()

    # SET THE NAME OF THE FILE
    NAME = "20_06_"+TEST+"_3D"
    #plt.savefig(FOLDER+NAME+".png",dpi=1500, bbox_inches = '
        tight')

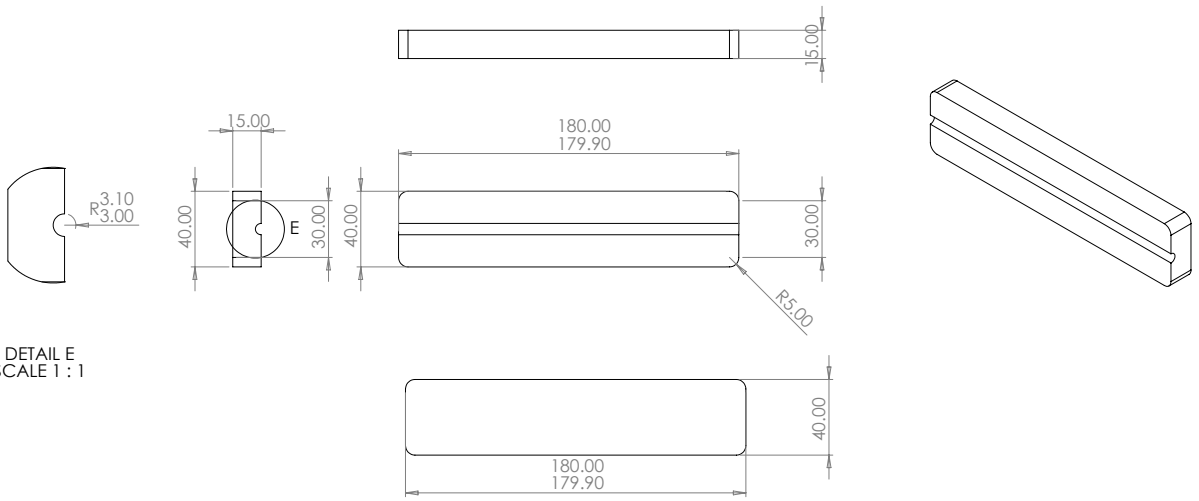
plt.show()
```

E

Load Frame Drawings



SOLIDWORKS Educational Product. For Instructional Use Only.

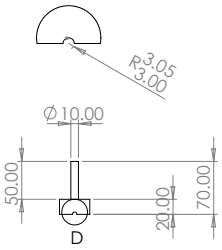


DETAIL E
SCALE 1 : 1

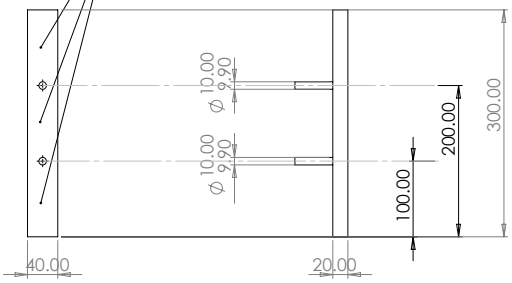
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE				TITLE: PART CB — PART CB IS TO FIT INTO THE DEPRESSION AT THE CENTRE OF PART PB
DRAWN							
CHKD							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. A3
				WEIGHT:			SCALE:1:2 / SHEET 4 OF 4

SOLIDWORKS Educational Product. For Instructional Use Only.

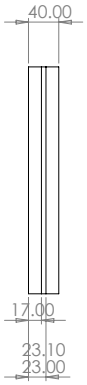
DETAIL D
SCALE 1 : 2



THERE WILL BE DIMPLES/
DEPRESSIONS
TO GUIDE THE BUTT OF THE
JACK-BOLT
FROM PART AB INTO PART CV

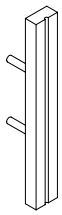


BACK

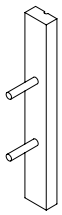


SIDE

FRONT



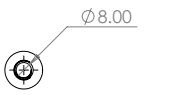
FRONT

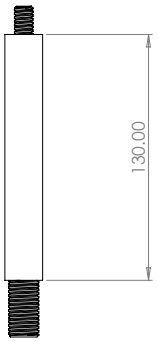


BACK

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN						PART CV	
CHK'D							
APP'V'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
				WEIGHT:		SCALE:1:5	SHEET 3 OF 3

SOLIDWORKS Educational Product. For Instructional Use Only.

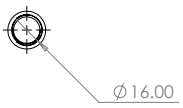




PART D TO THREAD THROUGH DOWEL HOLES OF PART AB AND THROUGH THE PART PB.
THE PART WILL SCREW INTO THE RIG.

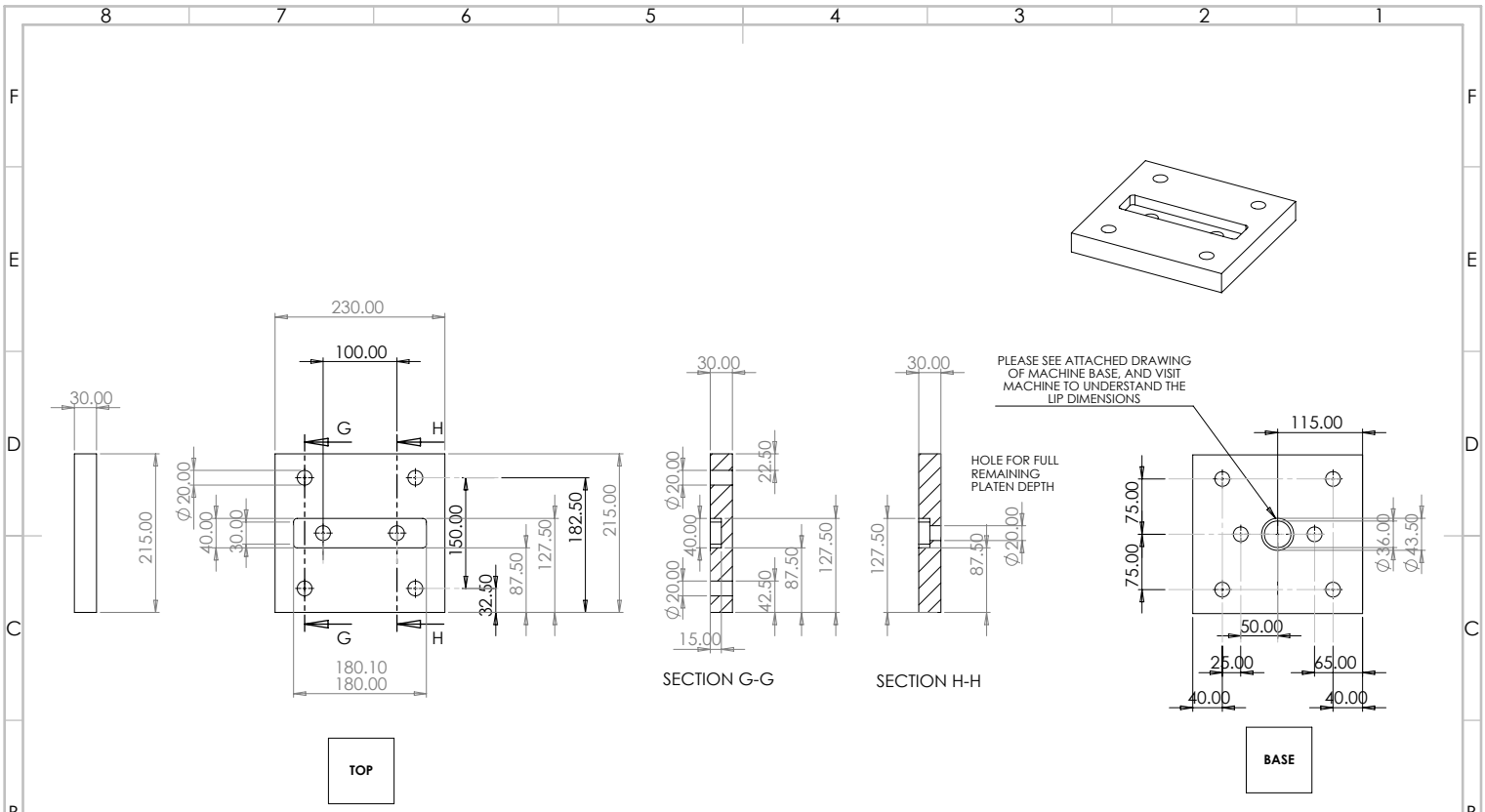
PLEASE VISIT THE INSTRON 5582 LOADING RIG TO UNDERSTAND THE PITCH OF THE 16 DIA THREAD END.
PLEASE SEE THE PLAN OF THE RIG. THE THREADS IN QUESTION ARE LABELLED AS 'B' ON THE TOP-LEFT IMAGE, AT THE CENTRE OF THE PLAN. THERE IS A CALL-OUT LABELLING THESE AS M16 X 30 DEEP.

THE 8 DIA THREAD END WILL SECTURE AB INTO THE BASE, WITH A WASHER



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE				TITLE: PART D	
DRAWN								
CHK'D								
APP'V'D								
MFG								
Q.A				MATERIAL:			DWG NO.	A3
				WEIGHT:			SCALE:1:2	SHEET 6 OF 6

SOLIDWORKS Educational Product. For Instructional Use Only.

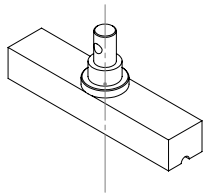


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN						PARTPB	
CHKD							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A3
				WEIGHT:		SCALE:1:5	SHEET 7 OF 7

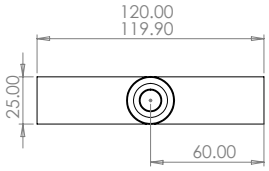
SOLIDWORKS Educational Product. For Instructional Use Only.

PLEASE CONFER WITH IGOR FOR THE COMPLEX DETACHABLE 'HEAD'. I HAVE ASKED HIM TO SHOW YO U PREVIOUSLY MANUFACTURED PARTS WITH THIS GEOMETRY.
THIS IS TO FIX PART PT INTO THE ACTUATOR OF THE INSTRON LOADING RIG.

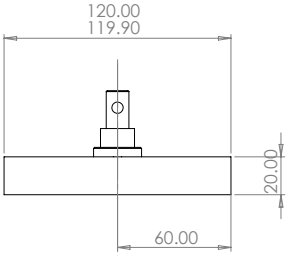
PLEASE ALSO NOTE THAT THE LENGTH OF THIS MEMBER IS 120mm WHEREAS THE PREVIOUS REVISION WAS 110mm.



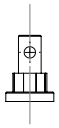
TOP

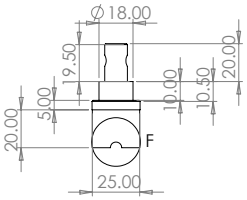


SIDE

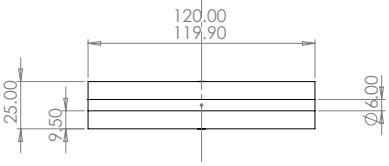


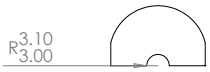
'HEAD'





UNDER SIDE





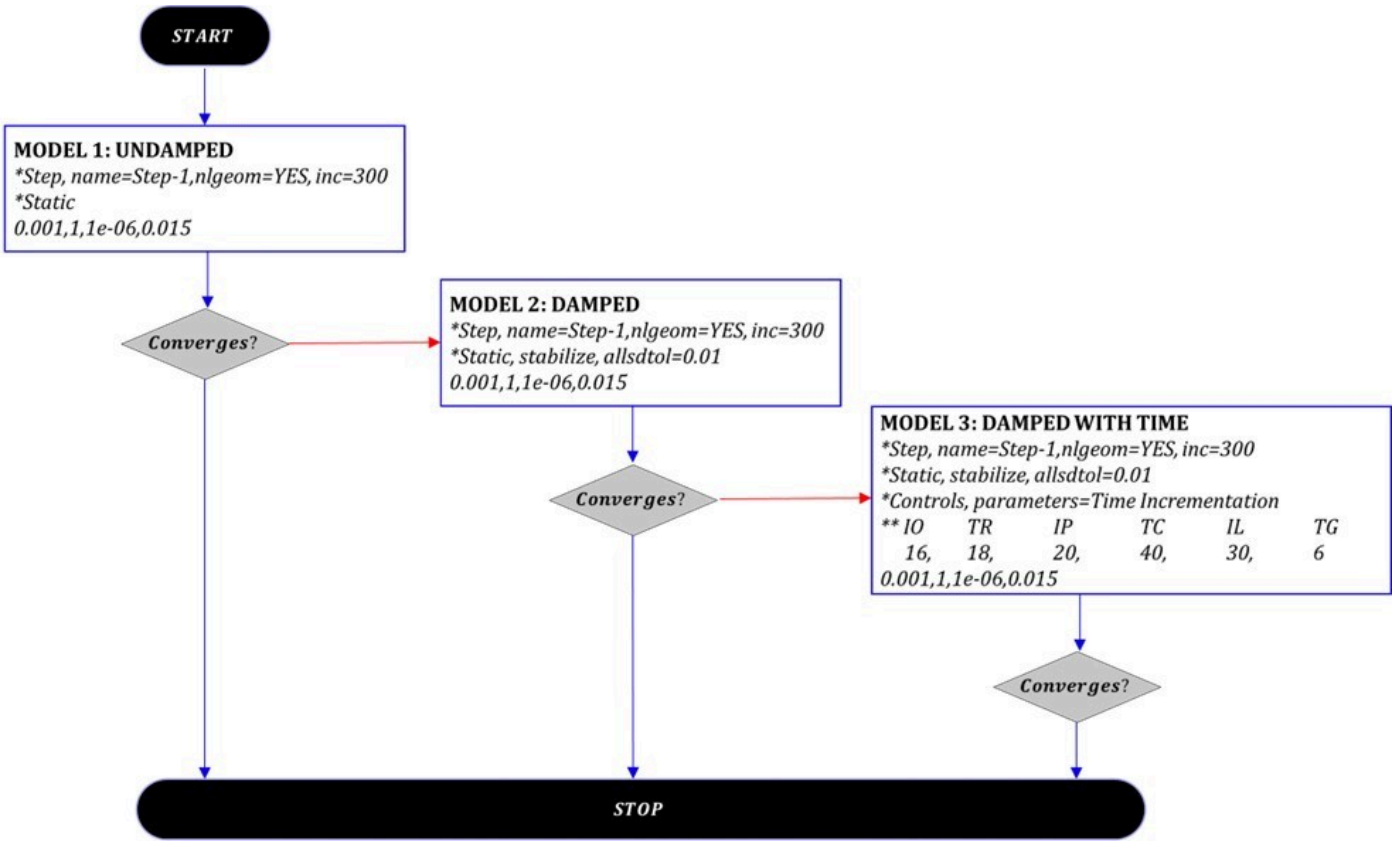
DETAIL F
SCALE 1 : 1

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
							THE 'HEAD' OF PART PT IS DETACHABLE	
	NAME	SIGNATURE	DATE				TITLE:	
DRAWN							PART PT	
CHK'D								
APP'V'D								
MFG								
Q.A				MATERIAL:			DWG NO.	A3
				WEIGHT:			SCALE:1:2	SHEET 5 OF 5

SOLIDWORKS Educational Product. For Instructional Use Only.

F

Nonlinear Model Setup


START
MODEL 1: UNDAMPED
*Step, name=Step-1,nlgeom=YES, inc=300
*Static
0.001,1,1e-06,0.015
Converges?
MODEL 2: DAMPED
*Step, name=Step-1,nlgeom=YES, inc=300
*Static, stabilize, allsdtol=0.01
0.001,1,1e-06,0.015
Converges?
MODEL 3: DAMPED WITH TIME
*Step, name=Step-1,nlgeom=YES, inc=300
*Static, stabilize, allsdtol=0.01
*Controls, parameters=Time Incrementation
** IO TR IP TC IL TG
16, 18, 20, 40, 30, 6
0.001,1,1e-06,0.015
Converges?
STOP

G
Mathematical Formulations

G.1 Chapter 6

G.1.1 Modified Slenderness $\lambda_{l,m}$

The modified slenderness introduced in this study is defined as:

$$\lambda_{l,m} = \sqrt{\frac{t}{b}}\sqrt{\frac{N_{y,net}}{N_0}}. \tag{G.1a}$$

To examine the physical meaning of this expression, consider the net section yield capacity of a perforated plate as:

$$N_{y,net} = f_y t(b - nl_y), \tag{G.2a}$$

Where n is the number of perforations across the plate width and l_y is the perforation height. The elastic local buckling capacity of the equivalent unperforated plate may be written as:

$$N_0 = k\frac{\pi^2 D}{b}, \tag{G.3a}$$

$$D = \frac{Et^3}{12(1-\nu^2)}. \tag{G.3b}$$

For a simply-supported plate under uniform compression, the buckling coefficient is:

$$k = \left(\frac{mb}{a} + \frac{a}{mb}\right)^2, \tag{G.4a}$$

$$= \frac{m^2b^2}{a^2} + 2 + \frac{a^2}{m^2b^2}, \tag{G.4b}$$

Where m is the number of longitudinal buckling halfwaves. It follows that:

$$\frac{N_{y,net}}{N_0} = \frac{f_y t(b - nl_y)}{\left(\frac{\pi^2}{b}\right)\left(\frac{Et^3}{12(1-\nu^2)}\right)k}, \tag{G.5a}$$

$$= \frac{12(1-\nu^2)}{\pi^2}\frac{f_y}{E}\left(\frac{b}{t}\right)^2\left(1 - \frac{nl_y}{b}\right)\frac{1}{k}. \tag{G.5b}$$

Substituting the expanded expression for k gives:

$$\frac{N_{y,net}}{N_0} = \frac{12(1-\nu^2)}{\pi^2}\frac{f_y}{E}\left(\frac{b}{t}\right)^2\left(1 - \frac{nl_y}{b}\right)\left[\frac{1}{\frac{m^2b^2}{a^2} + 2 + \frac{a^2}{m^2b^2}}\right]. \tag{G.6a}$$

.

The conventional slenderness according to [116] based on the unperforated elastic buckling capacity is therefore:

$$\lambda_l = \sqrt{\frac{N_{y,net}}{N_0}}, \tag{G.7a}$$

$$= \frac{b}{t}\sqrt{\frac{12(1-\nu^2)}{\pi^2}\frac{f_y}{E}\left(1 - \frac{nl_y}{b}\right)\left[\frac{1}{\frac{m^2b^2}{a^2} + 2 + \frac{a^2}{m^2b^2}}\right]} \tag{G.7b}$$

.

Thus, the conventional slenderness retains a direct dependence on b/t. However, the ordinate used in the DSM-style relationship is already normalised by the net section yield capacity, namely $N_u/N_{y,net}$. Therefore, much of the net-area and yield-strength dependence has already been removed from the capacity ratio. If λ_l is used on the abscissa, the data retain a comparatively strong horizontal dependence on b/t, which can increase scatter between plates that otherwise possess similar ultimate-to-yield behaviour. By contrast, the modified slenderness becomes:

$$\lambda_{l,m} = \sqrt{\frac{t}{b}}\sqrt{\frac{N_{y,net}}{N_0}}, \tag{G.8a}$$

$$= \sqrt{\frac{12(1-\nu^2)}{\pi^2}\frac{f_y}{E}\left(\frac{b}{t}\right)\left(1 - \frac{nl_y}{b}\right)\left[\frac{1}{\frac{m^2b^2}{a^2} + 2 + \frac{a^2}{m^2b^2}}\right]} \tag{G.8b}$$

.

The effect of the factor $\sqrt{t/b}$ is therefore to reduce the thickness dependence from:

$$\lambda_l \propto \frac{b}{t}, \tag{G.9a}$$

to:

$$\lambda_{l,m} \propto \sqrt{\frac{b}{t}} \tag{G.10a}$$

.

This provides a physical explanation for the reduced scatter observed when $N_u/N_{y,net}$ is plotted against $\lambda_{l,m}$. The modified slenderness preserves the influence of material stiffness, net section loss, aspect ratio and buckling halfwave number, while reducing the excessive dominance of the geometrical slenderness b/t. It follows that $\lambda_{l,m}$ provides a more suitable measure of the residual plate-like instability behaviour of perforated plates than the conventional force-based slenderness $\sqrt{N_{y,net}/N_0}$.

G.1.2 Geometric Parameter Ω

It is important to acknowledge that Ω is neither dimensionless nor dimensionally homogeneous (Equation 6.10). Therefore, it should not be interpreted as a general similarity parameter. Instead, it should be interpreted as a calibrated geometric index, valid for the units used in the regression analysis. In this study, all geometric variables entering Ω are expressed in millimetres. To make this explicit, the same parameter may be written in dimensionless form by introducing a reference length $L_0 = 1$ mm:

$$\Omega^* = L_0 \frac{l_x}{S_{yo}^2} + \sqrt{L_0} \frac{\sqrt{l_y}}{S_{xo}}. \tag{G.11a}$$

When l_x, l_y, S_{yo}, and S_{xo} are expressed in millimetres, Ω^* has the same numerical value as Ω. This reformulation clarifies that the proposed expression is not dimensionally universal, but is a unit-calibrated empirical index. Similar calibrated indices are common in design-oriented regression models, provided that the variables, units and range of applicability are clearly stated. The physical meaning of Ω is therefore the tendency for perforations to produce localised collapse; the term l_x/S_{yo}^2 measures the susceptibility of narrow transverse ligaments to local strip behaviour, while $\sqrt{l_y}/S_{xo}$ measures the disturbance of the longitudinal load path caused by tall perforations close to the loaded edge. Small values ($\Omega \leq 0.265$) correspond to plates whose ultimate response remains comparatively globalised, whereas large values ($\Omega > 0.265$) correspond to geometries where ultimate failure occurs due to localised ligament behaviour.

G.2 Chapter 7

G.2.1 Introduction

Let the equation found with regression for each of the four hole height ratios (l_y/b) investigated be $f_1(l_y, S_{xo}, b)$ and the equation found with the scale factors ψ_i be $f_2(l_y, S_{xo}, b)$.

The equations for each of the scale factors ψ_i (Equation 7.3) were determined by minimising the error between the turning point found with f_1 and f_2 for each $l_y/b = 0.6$, 0.7, 0.8 and 0.9. These scale factors control (1) what proportion of $f_1(l_y/b = 0.6)$ is in f_2, (2) what proportion of $f_1(l_y/b = 0.9)$ is in f_2, and (3) how f_2 should be scaled in the $x-$ direction. These are ψ_1, ψ_2 and ψ_3 respectively.

By iteration, the values in Table G.1 were modified, and a cubic line-of-best fit was determined through these points for each ψ_i (Figure G.1(a)). These equations are given in Equation 7.3, and repeated here for ease. This gives the continuous series of curves already seen in Chapter 7 (Figure G.1(b)).

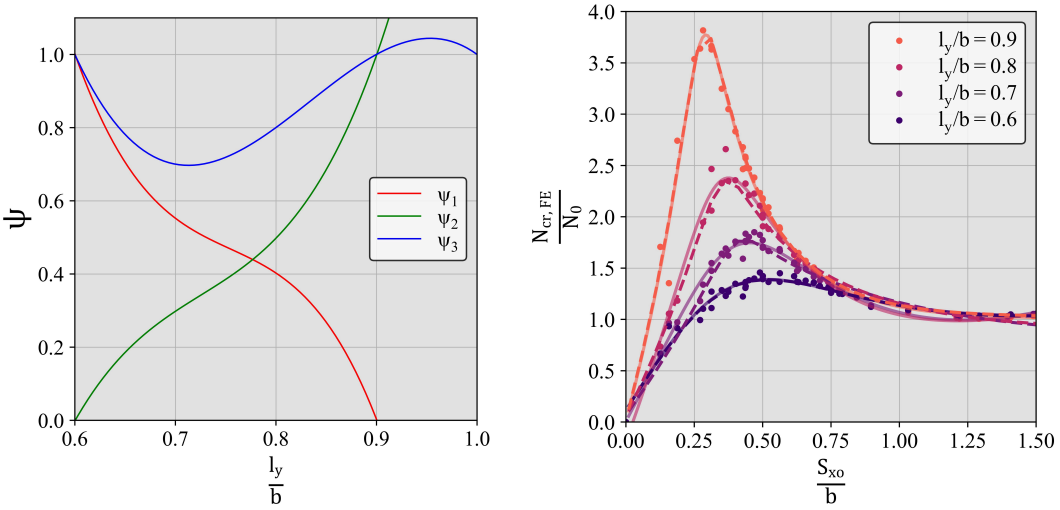
$$\psi_1 = -91.667 \cdot \left(\frac{l_y}{b}\right)^3 + 207.5 \left(\frac{l_y}{b}\right)^2 - 157.83 \left(\frac{l_y}{b}\right) + 40.8 \tag{G.12a}$$

$$\psi_2 = 66.667 \cdot \left(\frac{l_y}{b}\right)^3 - 145 \left(\frac{l_y}{b}\right)^2 + 106.83 \left(\frac{l_y}{b}\right) - 26.3 \tag{G.12b}$$

$$\psi_3 = -50 \cdot \left(\frac{l_y}{b}\right)^3 + 125 \left(\frac{l_y}{b}\right)^2 - 102 \left(\frac{l_y}{b}\right) + 28 \tag{G.12c}$$

Table G.1: Values used to parametrise the scale factors ψ in Chapter 7

		l_y/b			
		0.6	0.7	0.8	0.9
ψ	ψ_1	1	0.55	0.4	0
	ψ_2	0	0.3	0.5	1
	ψ_3	1	0.7	0.8	1



(a) The values for ψ given in Equation G.12 shown for the range of admissible l_y/b

(b) f_1 (solid) and f_2 for $l_y/b = 0.6, 0.7, 0.8, 0.9$ found when the values for ψ shown on the left are used in Equation 7.3(a)

Figure G.1: The final expressions for f_1 and f_2, parametrised by ψ_i, as given in Chapter 7

To demonstrate how each of the values in Table G.1 affect ψ_i, three examples are now shown for $i = 1, 2, 3$. In each case, only one value is changed, corresponding to the $l_y/b = 0.7$ curve.

G.2.2 Scale Factor ψ_1

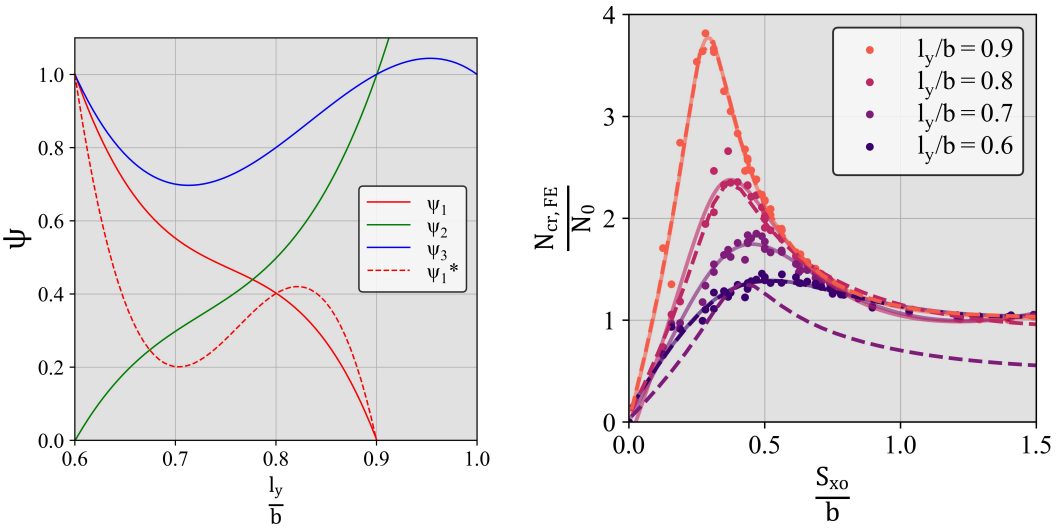
In this example, $f_2(l_y/b = 0.7)$ is composed with a smaller proportion of the $f_1(l_y/b = 0.6)$ curve (0.55 changes to 0.7, compare Table G.1 to Table G.2). Equation G.12 (a) becomes:

$$\psi_1 = -266.67 \cdot \left(\frac{l_y}{b}\right)^3 + 610 \left(\frac{l_y}{b}\right)^2 - 462.33 \left(\frac{l_y}{b}\right) + 116.4 \tag{G.13}$$

Observe how $f_2(l_y/b = 0.7)$ 'shrinks' whereas $f_2(l_y/b = 0.6, 0.8, 0.9)$ remain unaffected. Interpolation between these curves also changes.

Table G.2: Values used to parametrise the scale factors ψ in Section G.2.2

		l_y/b			
		0.6	0.7	0.8	0.9
ψ	ψ_1	1	0.2	0.4	0
	ψ_2	0	0.3	0.5	1
	ψ_3	1	0.7	0.8	1



(a) The values for ψ_1 given in Equation G.13, and ψ_2, ψ_3 given in Equation G.12 shown for the range of admissible l_y/b

(b) f_1 (solid) and f_2 for $l_y/b = 0.6, 0.7, 0.8, 0.9$ found with the values for ψ_1 given in Equation G.13, and ψ_2, ψ_3 given in Equation G.12

Figure G.2: The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.2

G.2.3 Scale Factor ψ_2

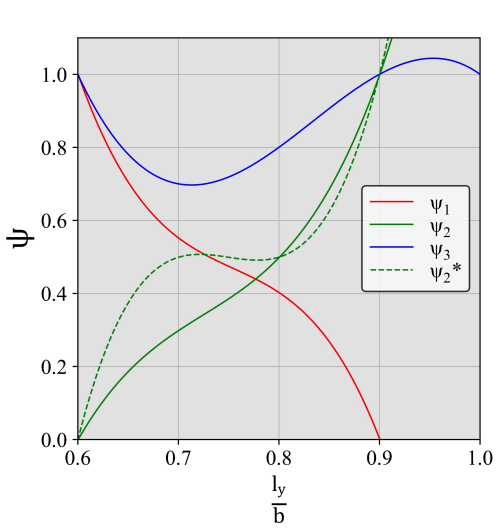
In this example, $f_2(l_y/b = 0.7)$ is composed with a larger proportion of the $f_1(l_y/b = 0.9)$ curve (0.3 changes to 0.5, compare Table G.1 to Table G.3). Equation G.12(b) becomes:

$$\psi_2 = 166.67 \cdot \left(\frac{l_y}{b}\right)^3 - 375\left(\frac{l_y}{b}\right)^2 + 280.8\left(\frac{l_y}{b}\right) - 69.5 \tag{G.14}$$

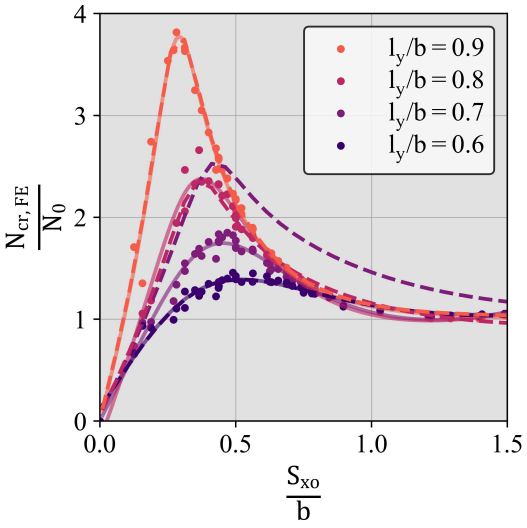
Observe how $f_2(l_y/b = 0.7)$ 'grows' whereas $f_2(l_y/b = 0.6, 0.8, 0.9)$ remain unaffected. Interpolation between these curves also changes.

Table G.3: Values used to parametrise the scale factors ψ in Section G.2.3

		l_y/b			
		0.6	0.7	0.8	0.9
ψ	ψ_1	1	0.55	0.4	0
	ψ_2	0	0.5	0.5	1
	ψ_3	1	0.7	0.8	1



(a) The values for ψ_2 given in Equation G.14, and ψ_1, ψ_3 given in Equation G.12 shown for the range of admissible l_y/b



(b) f_1 (solid) and f_2 for $l_y/b = 0.6, 0.7, 0.8, 0.9$ found with the values for ψ_2 given in Equation G.14, and ψ_1, ψ_3 given in Equation G.12

Figure G.3: The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.3

G.2.4 Scale Factor ψ_3

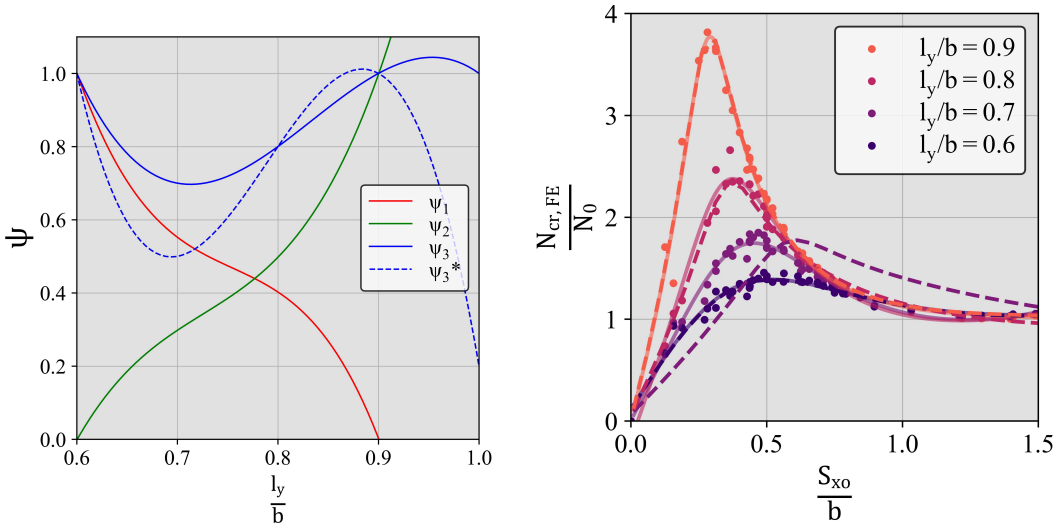
In the final example, $f_2(l_y/b = 0.7)$ is composed with a larger stretch in the $x-$direction (0.7 changes to 0.5, compare Table G.1 to Table G.4). Equation G.12(c) becomes:

$$\psi_3 = -150 \cdot \left(\frac{l_y}{b}\right)^3 + 355\left(\frac{l_y}{b}\right)^2 - 276\left(\frac{l_y}{b}\right) + 71.2 \tag{G.15}$$

Observe how $f_2(l_y/b = 0.7)$ is stretched more in the $x-$ direction, whereas $f_2(l_y/b = 0.6, 0.8, 0.9)$ remain unaffected. Once again, interpolation between these curves also changes.

Table G.4: Values used to parametrise the scale factors ψ in Section G.2.4

		l_y/b			
		0.6	0.7	0.8	0.9
ψ	ψ_1	1	0.55	0.4	0
	ψ_2	0	0.3	0.5	1
	ψ_3	1	0.5	0.8	1



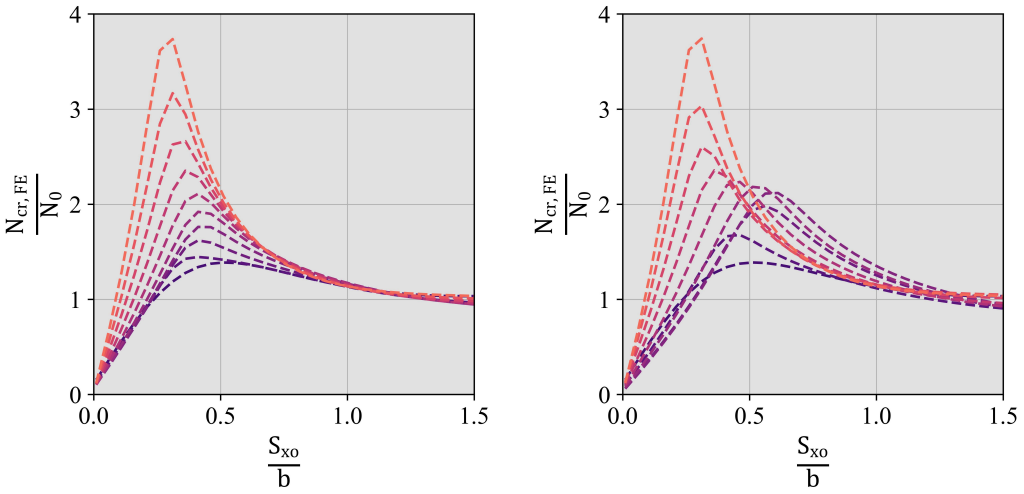
(a) The values for ψ_3 given in Equation G.15, and ψ_1, ψ_2 given in Equation G.12 shown for the range of admissible l_y/b

(b) f_1 (solid) and f_2 for $l_y/b = 0.6, 0.7, 0.8, 0.9$ found with the values for ψ_3 given in Equation G.15, and ψ_1, ψ_2 given in Equation G.12

Figure G.4: The expressions for f_1 and f_2, parametrised by ψ_i, given in Table G.4

G.2.5 Conclusion

Notice that the value changed in Section G.2.2 to Section G.2.4 only affects f_2 for $l_y/b = 0.7$. This is how the values for $l_y/b = 0.7$ and 0.8 were fine-tuned in Table G.1; the average percentage error between f_1 and f_2 at the turning point maximum was minimised to give the equations presented in Equation 7.3. In Figure G.5 compare how f_2 changes when each value for ψ_i is determined with Equation G.12 (Figure G.5(a)), to when ψ_1 is found with Equation G.13, ψ_2 is found with Equation G.14 and ψ_3 is found with Equation G.15 (Figure G.5(b)).



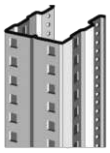
(a) The values for ψ given in Equation G.12 used to determine f_2 for $0.6 \leq l_y/b \leq 0.9$

(b) The value for ψ_1 given in Equation G.13, ψ_2 given in Equation G.14, and ψ_3 given in Equation G.15 used to determine f_2 for $0.6 \leq l_y/b \leq 0.9$

Figure G.5: Examples of f_2 with different expressions for ψ_i

H

Typical Rack Sections

Manufacturer	Section Description	Section Dimensions (mm)	Perforation Description	Section Perforation Pattern	Image
SSI Schaefer	Section name: PR 350 frame. Channel section with perforated webs and flanges. **Welded frame construction**	Unsupported length range: 1000 - 2500 Width range: 70 - 100	Webs: Two rows of square perforations Flanges: One row of circular perforations	50mm spacing in width and length of section. Perforation dimension unspecified, but approx. equal to spacing.	
SSI Schaefer	Section name: PR 600 frame. Lipped channel section with perforated webs and flanges. **Bolted frame construction**	Unsupported length range: 750 - 2500 Width range: 70 - 120	Webs: Two rows of square perforations Flanges: Two rows of square or circular perforations	50mm spacing in width and length of section. Perforation dimension unspecified, but approx. equal to spacing.	
Ridg-U-Rak	Slotted Frames Six-bend, roll-formed channel section with perforated webs and flanges **Welded frame**	Unsupported length range: 914 – 2743 Width range: 63.5 - 165	Webs: Two rows of slotted (rectangular) perforations Flanges: One row of circular perforations	100mm spacing in length of section. Perforation dimension unspecified, but length approx. equal to spacing.	
Ridg-U-Rak	Teardrop Frames Six-bend, roll-formed channel section with perforated webs and flanges **Welded frame**	Unsupported length range: 914 – 2743 Width range: 63.5 - 165	Webs: Two rows of teardrop perforations Flanges: One row of square perforations	50mm spacing in length of section. Perforation dimension unspecified, but length approx. equal to spacing.	
Steel King	SK3000 Channel	Unsupported length range: 1219 - 4876 Width range: 76.2 – 152	Webs: Two rows of circular perforations Flanges: Unperforated	50mm spacing in length of section. Perforation dimension unspecified, but approx. equal to 10mm	
UNCARO	Open or closed-tube interchangeable non-bolted upright pallet rack	Unsupported length range: 914 - 2438 Width range: 63.5 – 152.4	Webs: Two rows of teardrop perforations Flanges: One row of square, or unperforated	50mm spacing in length of section. Perforation dimension unspecified, but length approx. equal to spacing.	
UNCARO	Open or closed-tube T-bolt upright pallet rack	Unsupported length range: 914 - 2438 Width range: 63.5 – 152.4	Webs: Two rows of rectangular perforations Flanges: One row of rectangular, or unperforated	75mm spacing in length of section. Perforation dimension unspecified.	

Manufacturer	Section Description	Section Dimensions (mm)	Perforation Description	Section Perforation Pattern	Image
UNCARO	Structural rack c-section	Unsupported length range: 914 - 2438 Width range: 76.2 – 101.6	Webs: two rows of circular perforations Flanges: unperforated	100mm centres in length direction. Perforation dimension unspecified, but approx. equal to 10mm.	
Interlake MECALUX	7-9 bend open channel. Section name: FR. 315	Unsupported length range: 1524 – 4877 Width range: 80 - 122	Webs: two rows of teardrop perforations. Flanges: one row of circular perforations.	Flange perforations to accommodate ~8mm dia bolts on min 50mm centres	
ProMan	Hot-rolled C-section pallet racking	Unsupported length range: 750 - 1100 Width range: 100 - 120	Webs: diagonal rectangular and circular pattern in 2 rows Flanges: one row of circular perforations	70mm in length of section	

I
Abalone

Grasshopper is a visual programming language, built into Rhinoceros 3D CAD software [31], widely used in architecture, engineering, product design, construction and digital fabrication workflows. At the time of development, there is no tool to enable users to optimise the arrangement of perforations within rack sections or cellular members subject to axial compression, which may trigger buckling, a dangerous and sudden instability phenomenon. Abalone is the first buckling-aware Grasshopper plug-in for the design of thin plates with perforations, and can be downloaded for free under CC0 licensing [29][30].

Abalone uses the plate buckling design equations published in [23] (Chapter 4), in addition to the findings presented in Chapter 7. Using these new formulae, I developed a python script which can optimise the arrangement of perforations subject to design conditions, using shape annealing. This exactly follows the research presented in Chapter 8.

The code for Abalone is intellectual property (IP), protected under copywrite with Oxford University Innovation, project number 24955. For the development of this IP, Barabara Rossi provided the idea of researching the elastic buckling of perforated plates, and oversaw the development of the buckling design equations which are an essential component of the IP. I (Elenor Naraidoo) independently suggested the implementation of these findings into an optimisation routine, and independently chose shape annealing after undertaking research into various techniques. I am responsible for the core research into perforated plate buckling, shape annealing optimisation and how the two can be

implemented in design (see Chapter 8). I am responsible for the primary development of this idea into python, and headed an examination into three case studies which used this python tool and compared their performance with finite element analysis (see Chapter 8). I guided my collaborator Ioannis Mirtsopoulos from ETH Zurich on the translation of the python script into C#, which was required to create a plug-in using the same language as Grasshopper. Ioannis Mirtsopoulos took the lead on translating the python script, and suggested minor improvements to the code.

Detail on each of the six Abalone components can be found on GitHub [29], in addition to two introductory videos I created [198]. The following sections provide a general overview on each of these components (Figure I.1). The name 'Abalone' was chosen as (i) it is a shell (appropriate as this research is founded on shell/plate theory) and (ii) it is a naturally perforated shell. For this reason, I designed each of the icons to be 'shell-themed'.

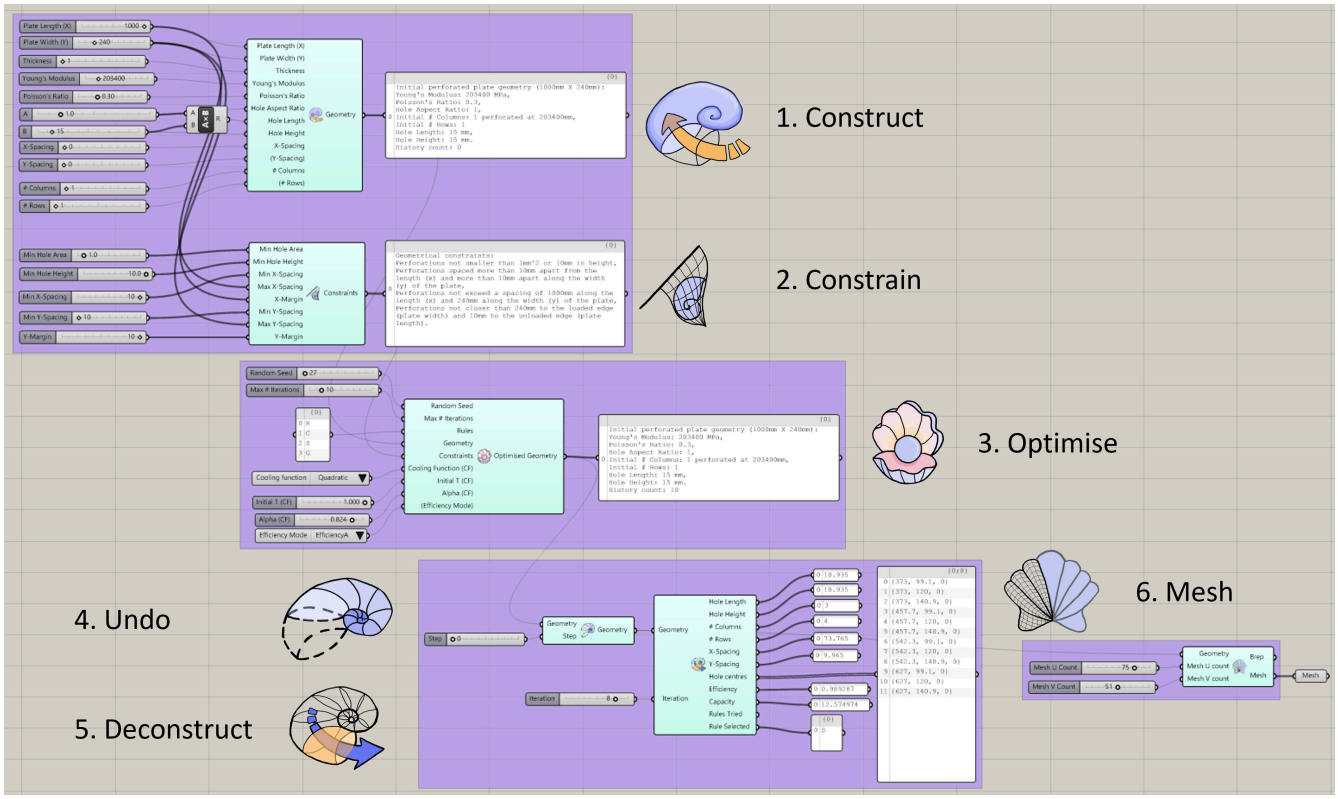


Figure I.1: Abalone component network pipeline

I.1 Construct

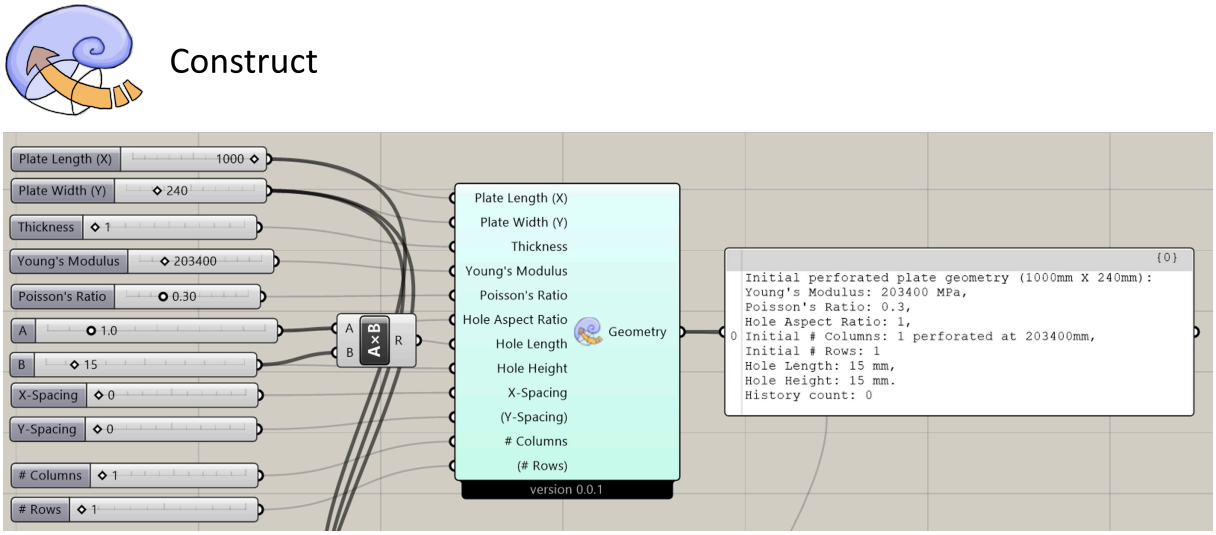


Figure I.2: Abalone construct component

The user inputs the dimensions of the plate (e.g. the dimensions of a rack section web):

- X – Length (referred to as a in this thesis) (mm)
- Y – Width (referred to as b in this thesis) (mm)
- t – Thickness (mm)

Please note that Abalone was developed for thin plates, i.e. those where $Y/t > 75$. The user inputs the material properties:

- E – Young's Modulus (MPa)
- v – Poisson's Ratio (referred to as ν in this thesis)

Abalone finds the most optimal arrangement of rectangular perforations defined by a height h (mm), and length l (mm) (referred to as l_y and l_x respectively in this thesis). The user must specify the aspect ratio (l/h) of these perforations, which will be maintained throughout optimisation. However, note that the size of these perforations may shrink or grow (i.e. h or l may change, while the ratio l/h remains the same). The user inputs:

- h_0 – The initial hole height (mm)
- l_0 – The initial hole length (mm)
- l/h - The hole aspect ratio

Finally, the user inputs information about the initial number of columns ($n_{c,0}$) in the plate, and the initial spacing of these columns ($s_{x,0}$). Note that these values will change as Abalone optimises the perforation geometry. The user may optionally input the initial number of rows ($n_{r,0}$), and row spacing ($s_{y,0}$). These are 1 and 0 respectively by default. This allows the user to force a minimum number of columns and/or rows, as it was found in Chapter 8 that plates with one row of perforations tend to be optimal under the objective function proposed in Equation 8.1.

I.2 Constrain

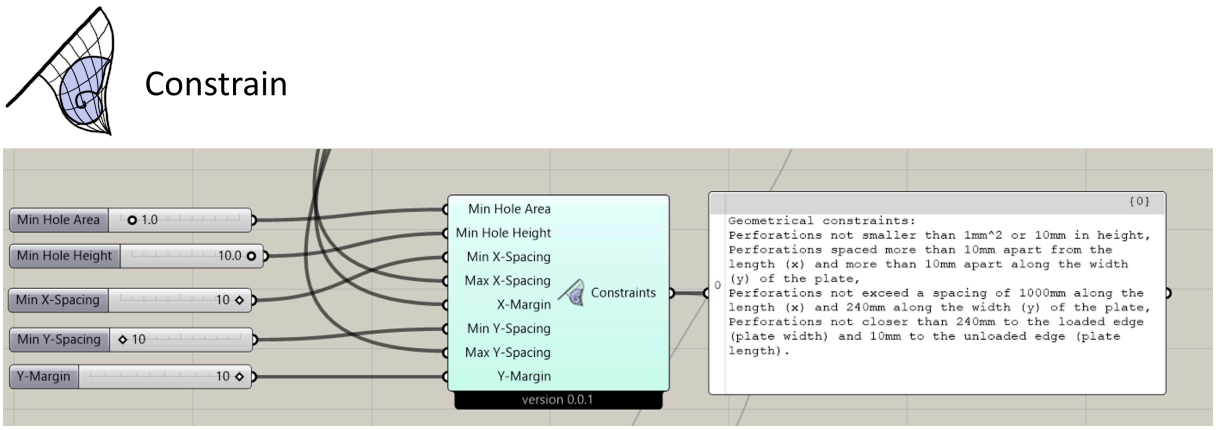


Figure I.3: Abalone constrain component

When Abalone optimises the geometry and arrangement of perforations within the plate, it must not exceed limits imposed by the user:

- Minimum Hole Area (mm^2)
- Minimum Hole Height (mm)
- Minimum X-Spacing ($s_{x,min}$)

I. Abalone

- Maximum X-Spacing ($s_{x,max}$)
- X-Margin (S_{xo})
- Minimum Y-Spacing ($s_{y,min}$)
- Maximum Y-Spacing ($s_{y,max}$)
- Y-Margin (S_{yo})

The 'margins' are shown in Figure 1.4, and restrict how close to either edge the perforations may reach. It is recommended to set $S_{xo} = Y$ (the plate width), which has been found to improve performance [119][46][72][38]. If there is no minimum, set this equal to 0. If there is no maximum, set $s_{x,max} = X$ and $s_{y,max} = Y$.

I.3 Optimise

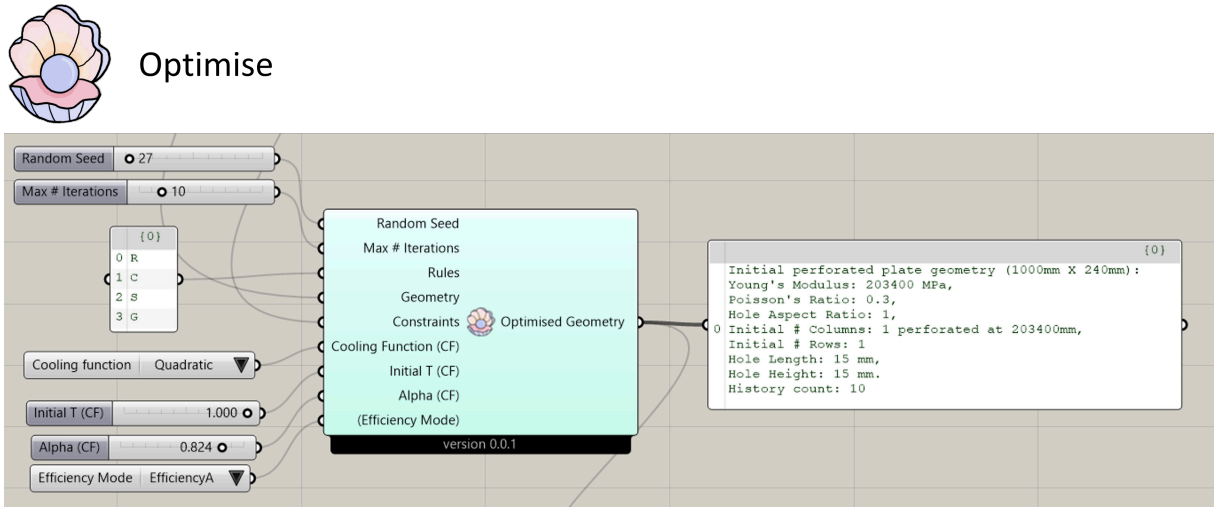


Figure I.4: Abalone optimise component

This is where shape annealing optimisation takes place. The user defines the maximum number of iterations, after which optimisation is considered complete. However, optimisation can terminate prematurely if no eligible rule can be selected at a given step; the history count in the output panel will detail how many iterations can be completed (Figure I.4).

I. Abalone

The user defines a list of rules that can be selected; Column (C), Row (R), Grow (G) and Shrink (S). The order of this list doesn't matter, and any of the 15 combinations of these rules can be considered. This list is connected to the 'Rules' label of the component.

- C – The addition of perforations in the X-direction (i.e. an additional 'column' of perforations)
- R – The addition of perforations in the Y-direction (i.e. an additional 'row' of perforations)
- G – An increase to the hole length by 10
- S – A reduction to the hole length by 10

Chapter 8 only examines exponential and reciprocal cooling functions, although in Abalone, the user can select from five. These are linear (Equation I.1a), exponential (Equation I.1b), logarithmic (Equation I.1c), reciprocal (Equation I.1d) or quadratic (Equation I.1e). These functions are described by the initial temperature (T_0, simply taken as 1 in Chapter 8), and alpha (α, taken as 0.5 in Chapter 8), which define the cooling rate $T(i)$ in terms of the iteration number i.

$$T(i) = \max(T_0 - \alpha i, 0) \tag{I.1a}$$

$$T(i) = T_0(\alpha^i) \tag{I.1b}$$

$$T(i) = \frac{T_0}{\ln(i + \alpha)} \tag{I.1c}$$

$$T(i) = \frac{T_0}{1 + \alpha i} \tag{I.1d}$$

$$T(i) = \frac{T_0}{1 + \alpha i^2} \tag{I.1e}$$

To understand how these parameters control the 'probability of acceptance' of a less efficient design state, see the tutorial video [198]. It is recommended to employ exponential cooling with $T_0 = 1$ and $\alpha = 0.5$, as per Chapter 8.

Finally, the user must select the 'efficiency mode', i.e. how the objective function is defined. The default is given in Equation 8.1 as:

$$\phi(i) = \lambda \cdot \frac{N_{i,h}}{N_0} \cdot \frac{A_0}{A_0 - A} \tag{I.2}$$

Where $N_{i,h}$ is the elastic buckling capacity at design state d_i, and N_0 is the elastic buckling capacity of the unperforated plate. The gross plate area is given by A_0, and the perforated area is A (see Equation 8.4). The user can select from three options. The default is $\lambda = 1$, but the user can opt to favour more columns ($n_m = n_c$), or more rows ($n_m = n_r$) by selecting Efficiency A and Efficiency B respectively. Additionally, there is the flexibility to choose $1 \leq k \leq 5$, allowing the user to bias the solution towards large n_m if so desired (Figure 8.12).

$$\lambda = \frac{1}{1 + e^{-(n_m - 4)}} \tag{I.3}$$

As this is a heuristic optimisation procedure, the user should change the random seed at least 30 times to create a pool of possible solutions, recording the seed with the best output. A more experienced user will also explore how α, T_0 and $T(i)$ affect the output.

I.4 Undo

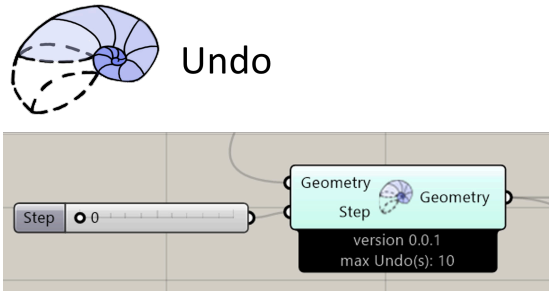


Figure I.5: Abalone undo component

The user can examine the state of the output at every step with the 'undo' component. This takes the final optimised geometry and 'steps back' the desired number of steps. Then, the result can be fed into the Deconstruct component.

I.5 Deconstruct

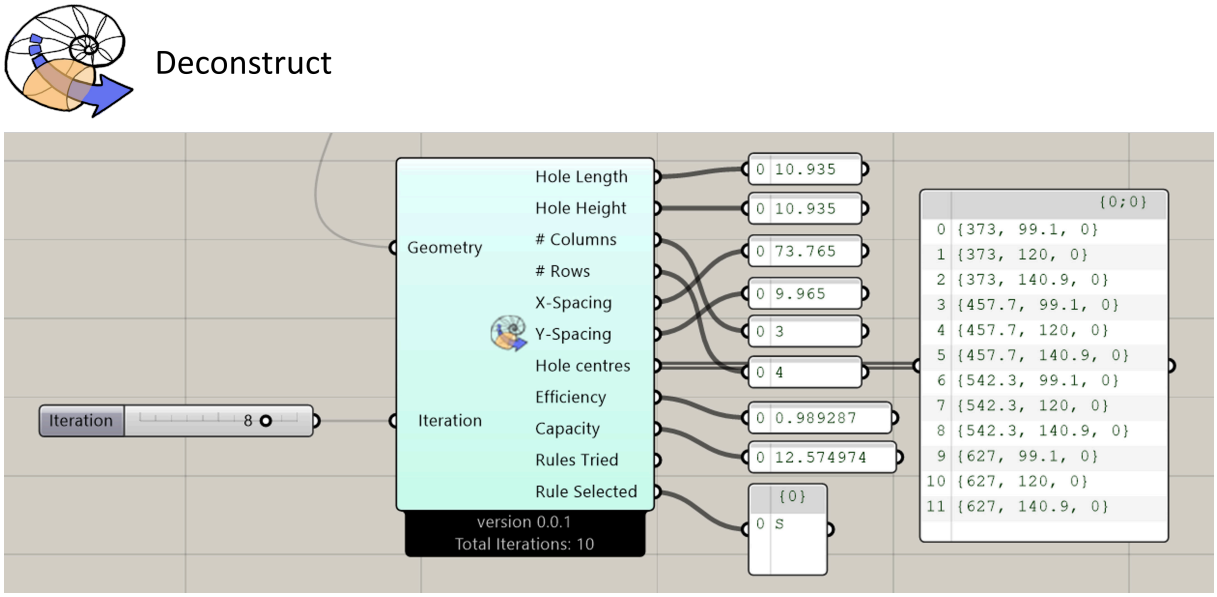


Figure I.6: Abalone deconstruct component

An optimised geometry (be this the final result from the Optimise component, or a back-stepped result from the Undo component) can be 'taken apart' with the Deconstruct component. This allows the user to record the geometry, efficiency, capacity, and rules selected at the chosen design state.

I.6 Mesh

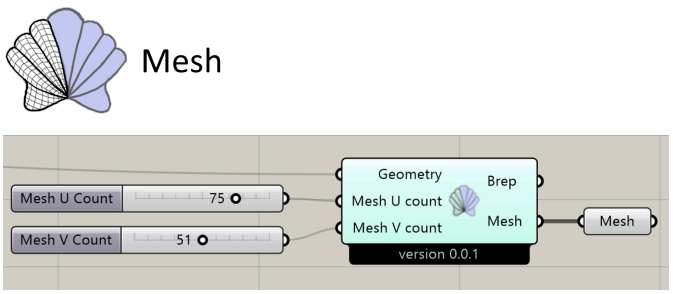


Figure I.7: Abalone mesh component

The geometry output from the Optimise component, or the Undo component (at the desired step) is input to the Mesh component so that the geometry can be 'baked' into

the Rhino3D GUI. The user can select either a brep or a mesh (controlled by the U-Count and V-count inputs) (Figure I.8).

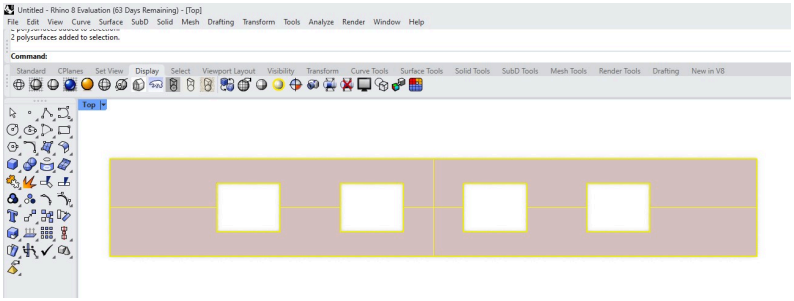


Figure I.8: An example of a perforated plate brep after optimisation with Abalone

J
Physics-Informed Neural Network

J.1 Main Code

```
# -*- coding: utf-8 -*-

# . . . . . .
    . . . . . #

                    #       S Y N O P S I S       #

# . . . . . .
    . . . . . #

# This is the 'main page' of code for training a PINN to predict
   the elastic buckling capacity of a certain plate geometry.

# The user defines the name of the file where the trained model
   is saved or stored, with a particular number of neurons and
   hidden layers.

# Note the list format so that multiple models can be trained in
   succession, all with the same training inputs, defined next.

# The capacities are taken from ABAQUS (N) and scaled according
   to the reduced PINN input geometry.

# See that particular home-made functions are imported from a
   folder called Functions. These de-clutter this 'main page' and
   are defined separately.
```

```
# . . . . . .
    . . . . . #

import torch
import torch.optim as optim
import torch.utils
import torch.utils.data
import numpy as np
import pandas as pd
import os

from Functions.Utilities import halfwaves, RayleighPINN,
   init_xavier
from Functions.Math_Disp import Get_Hole_Border_Points
from Functions.ReLoBRaLo_V2 import ReLoBRaLoLoss
from Functions.Parametric_PINNS_Functions import BuildDomains,
   fit_rq, Plot_Prediction,  Plot_Prediction_V2

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# File management:
    # 1. Saving the model (pth folder, dirp)
    # 2. Saving the training results (excel folder, dire)
    # 3. Saving any images (image folder, diri)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
DIR = "C:/Users/eleno/OneDrive - Nexus365/PINNS_Grid_Search/"
dirp = DIR + "PTH/"
dire = DIR + "EXCEL/"
diri = "C:/Users/eleno/OneDrive - Nexus365/Documents/YEAR4/
   COMPUTATIONAL/PINNS_Outputs/Universal_PINN_Study/Grid_Search/
   IMG/FINAL/Test/"#
shna = "Forward_LBFGS_1"

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Set the following controls:
    # 1. Print = True, then a new model with the name in FILES is
        trained.
    #    Print = False, then a model with the name in FILES is
       loaded and used on test set.

    # 2. Result = True, then the model is used to predict the
       capacity of cases, and any images are saved.

    # 3. Test = True, then the model is run on the validation set
        (*see 'Manual').
    #    Test = False, then the model is run on the training set
       (*see 'Manual').
```

```
    # 4. Manual = True, then the model is run on data
       automatically.
    #    Manual = False, then the user can directly input data to
       test.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
Print = False
Result = True
Test = True
Manual = False
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Name of the file, number of hidden layers, number of neurons in
    each hidden layer
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
FILES = ["04_02_ep500_NL4_NN150"]
n_hidden_layers_list = [4]
n_neurons_list = [150]


for FILE, n_hidden_layers, n_neurons in zip(FILES,
   n_hidden_layers_list, n_neurons_list):
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Training data:
    # Geometry of the plate (a, b, T).
    # Material properties (v, E).
    # Plate flexural constant (D).

    # The length of the perforation l_x (L_X is a list of lengths
       for all training input cases).
    # The height of the perforation l_y (L_Y is a list of heights
       for all training input cases).
    # The x-centre of the perforation c_x (C_X is a list of x-
       centres for all training input cases).
    # The y-centre of the perforation c_y (C_Y is a list of y-
       centres for all training input cases).
    # The known capacity of each training input case.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    a, b = 2.2, 1
    T = 0.01
    m, n = halfwaves(a/b), 1

    v = 0.3
    E = 203.4E3
```

```
    D = (E*T**3)/(12*(1-v**2))

    L_X = [0.05, 0.05, 0.05, 0.05, 0.1, 0.1, 0.1, 0.1, 0.15,
       0.15, 0.15, 0.15, 0.2, 0.2, 0.2, 0.2, 0.25, 0.25, 0.25,
       0.25, 0.3, 0.3, 0.3, 0.3]
    L_Y = [0.05, 0.05, 0.05, 0.05, 0.1, 0.1, 0.1, 0.1, 0.15,
       0.15, 0.15, 0.15, 0.2, 0.2, 0.2, 0.2, 0.25, 0.25, 0.25,
       0.25, 0.3, 0.3, 0.3, 0.3]
    C_X = [a/2, a/4, a/4, a/2, a/2, a/4, a/4, a/2, a/2, a/4, a/4,
        a/2, a/2, a/4, a/4, a/2, a/2, a/4, a/4, a/2, a/2, a/2, a
       /4, a/4]
    C_Y = [b/2, b/2, b/4, b/4, b/2, b/2, b/4, b/4, b/2, b/2, b/4,
        b/4, b/2, b/2, b/4, b/4, b/2, b/2, b/4, b/4, b/2, b/4, b
       /4, b/2]
    Capacities = [0.73924, 0.73276, 0.73526, 0.73745, 0.74708,
       0.73426, 0.72504, 0.72166, 0.75542, 0.70319, 0.70717,
       0.72828, 0.76493, 0.68775, 0.68151, 0.71897, 0.76079,
       0.67476, 0.64684, 0.7042, 0.76657, 0.67851, 0.6024,
       0.66908]
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Training controls:
    # 'oversampling' multiplier for each input (oms)
    # The number of points along each boundary edge (npoints)
    # The number of collocation points within the plate (
       residual_pts)
    # A random seed for point selection
    # The maximimum number of iterations in the LBFGS algorithm
    # The LBFGS learning rate
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    oms = [1.02, 1.02, 1.02, 1.02, 1.05, 1.05, 1.05, 1.05, 1.08,
       1.08, 1.08, 1.08, 1.12, 1.12, 1.12, 1.12, 1.14, 1.14, 1.14,
        1.14, 1.16, 1.16, 1.16, 1.16]
    npoints = 2000
    residual_pts = 8000
    rseed = 1
    num_epochs = 2000
    mit = 1                                                 # DEFAULT
        : 1
    ler = 0.1                                               # DEFAULT
        : 0.1
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# See the BuildDomains funtion
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    Number_Of_Loss_Terms = 3
    Num_Domains = len(L_X)
```

```
    Domain = BuildDomains(Num_Domains)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# These dataframes are output to an Excel file
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    dout = {"Epoch":[], "Rayleigh Quotient Loss":[], "Hole Loss
       ":[], "Capacity Loss": [], "Total Loss": []}
    vout = {"a":[a],"b":[b],"m":[m], "n":[n],"l_x":[L_X],"c_x":[
       C_X],"l_y":[L_Y],"c_y":[C_Y], "N. Residual Points":[4*
       npoints], "N. epochs": [num_epochs], "N. Hidden Layers": [
       n_hidden_layers],"N. Neurons Per Layer": [n_neurons]}

    if Print:

        for p, (l_x, l_y, c_x, c_y, om) in enumerate(zip(L_X, L_Y
           , C_X, C_Y, oms)):

            oversample_multiplier = om
            interior, n_hb, border_pts_left, border_pts_right,
               border_pts_bot, border_pts_top =
               Get_Hole_Border_Points(a, b, m, n, l_x, l_y, c_x,
               c_y, npoints, residual_pts, oversample_multiplier,
               rseed)
            Domains = Domain(p, interior, l_x, l_y, c_x, c_y,
               border_pts_left, border_pts_right, border_pts_bot,
               border_pts_top)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Initialising the PINN
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
        Model = RayleighPINN(N_modes = 1,a = a, b = b,
           input_dimension = 6, output_dimension = 1,
           n_hidden_layers= n_hidden_layers, neurons=n_neurons)
        init_xavier(Model, rseed)

        optimizer_ = optim.LBFGS(Model.parameters(), lr=ler,
           max_iter=mit, max_eval=50000, tolerance_change=1.0 * np
           .finfo(float).eps)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Adaptive weights using Relative Lookbacks (see ReLoBRaLoLoss
   function)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
        relo = ReLoBRaLoLoss(Number_Of_Loss_Terms)
        My_Network = fit_rq(relo, v, D, Model, optimizer_,
           Domains, Capacities)
```

```
# - - -
```

```
        E = 203.4E3
        T = 0.01
        D = (E*T**3)/(12*(1-v**2))

        test_fil = "C:/Users/eleno/OneDrive - Nexus365/Documents/
           YEAR4/COMPUTATIONAL/PINNS_Outputs/Universal_PINN_Study/
           Grid_Search/EXCEL/Test_Data.xlsx"
        test_data = pd.read_excel(test_fil, sheet_name = "In
           paper", index_col = "Number")

        L_X = test_data.loc[:,"lx"]
        L_Y = test_data.loc[:,"ly"]

        test_areas = [x*y for x, y in zip(L_X, L_Y)]
        oms = [0.13*(a-0.01)/0.08 + 1.02 for a in test_areas]

        C_X = test_data.loc[:,"cx"]
        C_Y = test_data.loc[:,"cy"]

        Capacities = test_data.loc[:,"Pcr"]
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# If Manual is true (see explanation at the start)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    if Manual:

        if Result:
            dfout = {"E":[], "t":[], "a":[], "b":[], "l_x":[], "
               l_y":[], "c_x":[], "c_y":[], "P_pr":[], "P_cr":[]}
            another = "Y"
            filresult = FILE + "_Results"
            Result = False
            count = 0

            directory = diri+FILE + "/Train/"
            #os.makedirs(directory)

        if Test:
            filresult = FILE + "_Results_Test"
            directory = diri+FILE + "/Test/"
            os.makedirs(directory)

        while not Result:

            c_x = float(input("c_x = "))
            c_y = float(input("c_y = "))
            l_x = float(input("l_x = "))
            l_y = float(input("l_y = "))
```

```
            v5 = float(input("P_cr = "))

            n_x = 500
            n_y = 500

            view = 3

            Plot_Prediction_V2(view, FILE, dirp, directory, FILE
               + str(int(count)) + "_Plot", n_hidden_layers,
               n_neurons, D, v, a, b, m, n, c_x, c_y, l_x, l_y,
               n_x, n_y)

            n_x = 60
            n_y = 60

            Predicted = Plot_Prediction(FILE, dirp,
               n_hidden_layers, n_neurons, D, v, a, b, m, n, c_x,
               c_y, l_x, l_y, n_x, n_y)

            dfout["E"].append(E)
            dfout["t"].append(T)
            dfout["a"].append(a)
            dfout["b"].append(b)
            dfout["l_x"].append(l_x)
            dfout["l_y"].append(l_y)
            dfout["c_x"].append(c_x)
            dfout["c_y"].append(c_y)
            dfout["P_pr"].append(Predicted)
            dfout["P_cr"].append(v5)

            count +=1

            another = str(input("Another Prediction? (Y/N)"))

            if another != "N":
                Result = False
            else:
                Result = True

        if Result:
            shna = f"Results_with_{FILE}"
            dfout = pd.DataFrame(dfout)
            with pd.ExcelWriter(dire+filresult+".xlsx",mode='w',
               engine='openpyxl') as writer:
                dfout.to_excel(writer,shna)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# If Manual is not true (see explanation at the start)
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    else:

        if Result:
            dfout = {"E":[], "t":[], "a":[], "b":[], "l_x":[], "
               l_y":[], "c_x":[], "c_y":[], "P_pr":[], "P_cr":[]}
            another = "Y"
            filresult = FILE + "_Results"

            directory = diri +FILE + "/Train/"
            #os.makedirs(directory)


        if Test:
            filresult = FILE + "_Results_Test_Good"
            directory = diri+FILE +"/"
            #os.makedirs(directory)

        for count,(v1, v2, v3, v4, v5) in enumerate(zip(C_X, C_Y,
            L_X, L_Y, Capacities)):

            c_x = v1#
            c_y = v2#
            l_x = v3#
            l_y = v4#

            n_x = 100
            n_y = 100

            view = 2

            Plot_Prediction_V2(view, FILE, dirp, directory, FILE
               + "_" + str(int(count)) + "_", n_hidden_layers,
               n_neurons, D, v, a, b, m, n, c_x, c_y, l_x, l_y,
               n_x, n_y)

            n_x = 60
            n_y = 60

            Predicted = Plot_Prediction(FILE, dirp,
               n_hidden_layers, n_neurons, D, v, a, b, m, n, c_x,
               c_y, l_x, l_y, n_x, n_y)

            dfout["E"].append(E)
            dfout["t"].append(T)
            dfout["a"].append(a)
            dfout["b"].append(b)
            dfout["l_x"].append(l_x)
```

```
            dfout["l_y"].append(l_y)
            dfout["c_x"].append(c_x)
            dfout["c_y"].append(c_y)
            dfout["P_pr"].append(Predicted)
            dfout["P_cr"].append(v5)

        if Result:
            shna = f"Results_with_{FILE}"
            dfout = pd.DataFrame(dfout)
            with pd.ExcelWriter(dire+filresult+".xlsx",mode='w',
               engine='openpyxl') as writer:
                dfout.to_excel(writer,shna)
```

J.2 Functions

The main code imports home-made functions from the Functions folder. The various files where these functions are stored are given in the following sub-sections.

J.2.1 Math Disp

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 21 10:06:32 2025

@author: eleno
"""
import torch
import torch.utils
import torch.utils.data
import numpy as np
from scipy.stats import qmc
import matplotlib.pyplot as plt
from math import pi
#.....................................................................


                #   F U N C T I O N S / C L A S S E S  #

#.....................................................................


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Receives a set of points and a mask where points must be
   excluded
```

```
# Thus defining a location of a hole within a field of points
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

def FilteredCoords(i, n,interior, Border, Mask):

    pts_filt = interior[Mask]
    d = torch.abs(pts_filt[:,i] - Border)
    d_sorted, idx_sorted = torch.sort(d)
    idx_n = idx_sorted[:n]
    Name = pts_filt[idx_n]

    return Name

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Defines a function called 'triangular_sobol' used later
# This spits a domain into a number of specified intervals, and
   randomly selects points using Sobol
# There are more points selected in the middle interval, and
   fewer at the ends of the domain
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def Triangular_Sobol(a, b, number_intervals, residual_pts, Seed):
    N_t = residual_pts
    N_i = number_intervals

    h_p_max = 2*N_t/a

    h_p = h_p_max/2

    P = 0.5*(2*N_t/a + h_p)

    Q = P - h_p

    k   = torch.arange(N_i, dtype=torch.float32)
    t   = k / (N_i-1)
    f   = 1 - torch.abs(2*t - 1)

    raw = Q + (P - Q) * f

    raw = raw / raw.sum() * N_t
    n_p = raw.round().long()    # integer counts

    # Adjust any rounding drift to the peak
    diff = int(N_t - n_p.sum().item())
    if diff != 0:
        peak_idx = f.argmax().item()
        n_p[peak_idx] += diff
```

```
    xs, ys = [], []

    for i, cnt in enumerate(n_p.tolist(), start=1):

        j_0 = (i-1)*a/N_i
        j_1 = i*a/N_i

        # Sobol samping
        eng = qmc.Sobol(d = 2, scramble = True, seed = Seed)
        m_p = int(np.ceil(np.log2(cnt)))
        pts = torch.from_numpy(eng.random_base2(m_p)[:cnt]).float
           ()
        xs.append(pts[:,0:1]*(j_1 - j_0) + j_0)
        ys.append(pts[:,1:2]*b)

    x_data_interior = torch.vstack(xs)
    y_data_interior = torch.vstack(ys)

    interior = torch.cat([x_data_interior, y_data_interior],1)

    return interior

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# To define the points excluding the location of the hole
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def Build_Interior(a, b, m, n, l_x, l_y, c_x, c_y, residual_pts,
   oversample_multiplier, rseed):

    kx = m*pi/a
    ky = n*pi/b

    oversample = int(oversample_multiplier * residual_pts)

    # Sample using sobol with 5 intervals
    # Triangular so that central interval has most points
    pts = Triangular_Sobol(a, b, 5, oversample, rseed)

    #Exclude gap
    mask = (pts[:,0] < c_x - l_x/2) | (pts[:,0] > c_x + l_x/2) |
       (pts[:,1] < c_y - l_y/2) | (pts[:,1] > c_y + l_y/2)

    #Select first npoints valid ones
    filtered = pts[mask]
    plt.scatter(filtered[:,0], filtered[:,1], s=2, alpha=0.2)
    plt.show()
```

```
    data_u_true = torch.sin(kx*filtered[:,0]) * torch.sin(ky*
       filtered[:,1])

    if filtered.shape[0] < residual_pts:
        print("Need to oversample more!")
        return filtered, data_u_true

    else:
        print(f"Size of interior is {filtered.size(dim = 0)}\
           nWhereas size of interior should be {residual_pts}\nTry
            reducing oversample_multiplier if size of interior is
           too large.")
        return filtered, data_u_true

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# This finds the points closest to the edge of the perforation
# The approximate number is n_hb, normalised by a hole of
   dimensions 0.225 x 0.225
# Note that this was determined by trial-and-error, and is
   appropriate for the plate geometry considered here
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def Get_Hole_Border_Points(a, b, m, n, l_x, l_y, c_x, c_y,
   npoints, residual_pts, oversample_multiplier, rseed):
    # Total number of border points on hole edge
    n_hb = int(npoints*(l_x + l_y)/(30*(0.225 + 0.225)))

    interior, data_u_true = Build_Interior(a, b, m, n, l_x, l_y,
       c_x, c_y, residual_pts, oversample_multiplier, rseed)

    # Hole Edge Ratio
    hole_ar = l_x / l_y
    n_hb_x_float = n_hb / (2 * (1 + hole_ar))
    n_hb_x = int(round(n_hb_x_float))
    n_hb_y = n_hb // 2 - n_hb_x
    # Recalculate n_hb in case rounding issues
    n_hb = 2*(n_hb_x + n_hb_y)

    #.....................................................................


    # Hole Edges
    x_left_bord = c_x - l_x/2
    x_left_mask = (interior[:,0] <= x_left_bord ) & (interior
       [:,1] >= c_y - l_y/2) & (interior[:,1] <= c_y + l_y/2)

    x_right_bord = c_x + l_x/2
```

```
    x_right_mask = (interior[:,0] >= x_right_bord ) & (interior
       [:,1] >= c_y - l_y/2) & (interior[:,1] <= c_y + l_y/2)

    y_bot_bord = c_y - l_y/2
    y_bot_mask = (interior[:,1] <= y_bot_bord ) & (interior[:,0]
       >= c_x - l_x/2) & (interior[:,0] <= c_x + l_x/2)

    y_top_bord = c_y + l_y/2
    y_top_mask = (interior[:,1] >= y_top_bord ) & (interior[:,0]
       >= c_x - l_x/2) & (interior[:,0] <= c_x + l_x/2)

    border_pts_left = FilteredCoords(0, n_hb_x, interior,
       x_left_bord, x_left_mask)
    border_pts_right = FilteredCoords(0, n_hb_x, interior,
       x_right_bord, x_right_mask)
    border_pts_bot = FilteredCoords(1, n_hb_y, interior,
       y_bot_bord, y_bot_mask)
    border_pts_top = FilteredCoords(1, n_hb_y, interior,
       y_top_bord, y_top_mask)

    return interior, n_hb, border_pts_left, border_pts_right,
       border_pts_bot, border_pts_top
```

J.2.2 Parametric PINNS Functions

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jan  7 12:15:16 2026

@author: eleno
"""

import torch
import torch.nn as nn
import numpy as np
import matplotlib.pyplot as plt

from Functions.Utilities import predict_capacity, plot_hole_pts,
   RayleighPINN
from Functions.Math_Disp import Get_Hole_Border_Points
from Functions.Prediction import Predictor

#.....................................................................

                #   F U N C T I O N S / C L A S S E S  #
```

```
#.....................................................................


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# BuildDomains manages the massive list of inputs, so that each
   interior matrix is treated independently
# Each input will have a different total number of interior
   points
# Similarly, each input will have a different total number of
   border points (hole_l, _r, _b, _t)
# Hence the necessity to first initialise the list, and then
   expand it accordingly.
# These variables are not trainable
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
class BuildDomains(nn.Module):

    def __init__(self, number_of_inputs):

        super().__init__()
        self.number_of_inputs = number_of_inputs

        # Create lists of variables which won't be updated during
            training
        self.interiors = [torch.empty(0, 6) for _ in range(
           number_of_inputs)]
        self.hole_l = [torch.empty(0, 2) for _ in range(
           number_of_inputs)]
        self.hole_r = [torch.empty(0, 2) for _ in range(
           number_of_inputs)]
        self.hole_b = [torch.empty(0, 2) for _ in range(
           number_of_inputs)]
        self.hole_t = [torch.empty(0, 2) for _ in range(
           number_of_inputs)]

    def forward(self, p, interior, l_x, l_y, c_x, c_y, hpl, hpr,
       hpb, hpt):

        device = interior.device
        dtype = interior.dtype

        def ExpandedTensor(x):
            g_cols = torch.tensor([l_x, l_y, c_x, c_y], dtype=
               dtype, device=device).expand(x.size()[0], 4).clone
               ()
            return torch.cat((x, g_cols), dim = 1)

        interior_combined = ExpandedTensor(interior)
```

```
        hole_l_combined = ExpandedTensor(hpl)
        hole_r_combined = ExpandedTensor(hpr)
        hole_b_combined = ExpandedTensor(hpb)
        hole_t_combined = ExpandedTensor(hpt)

        with torch.no_grad():
            self.hole_t[p] = hole_t_combined.detach()
            self.hole_l[p] = hole_l_combined.detach()
            self.hole_r[p] = hole_r_combined.detach()
            self.hole_b[p] = hole_b_combined.detach()
            self.interiors[p] = interior_combined.detach()

        return self.interiors, self.hole_l, self.hole_r, self.
           hole_b, self.hole_t

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# This is the main code for training the PINN
# It receives the collocation points (interior) and the boundary
   points (hole_l, _r, _t, _b) which vary from input to input (see
    above).
# In forward loss, all of the relevant differentials are
   determined using automatic differentiation.
# Then, these are used to determine the Rayleigh Quotient (
   physical loss, loss_pde).
# Gramm-Schmidt is #'ed out because this should only find the
   first mode and so G-S not necessary here.
# The capacity that was input to training is used to determine
   the capacity loss (loss_cap).
# The appropriate physical conditions are applied on the hole
   boundary to determine the hole border loss (loss_hole).
# Adaptive weights are determined using relative lookbacks (see
   RELO).
# The loss is returned.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
class fit_rq(nn.Module):

    def __init__(self, RELO, v, D, model, optimizer, domains,
       capacities):
        super().__init__()
        self.RELO = RELO

        self.v = v
        self.D = D
        self.model = model
        self.optimizer = optimizer

        (self.interiors,
```

```
        self.hole_l,
        self.hole_r,
        self.hole_b,
        self.hole_t) = domains

        self.capacities = capacities

        self.history = []
        self._running_loss = 0.0

    def _forward_loss(self, epoch, dout, interior, hpl, hpr, hpb,
        hpt, N_true):

        #.....................................................................


        # P L A T E    P D E

        #.....................................................................


        u_hat = self.model(interior)
        print(f"u_hat = {u_hat}")
        inputs = torch.ones(interior.size()[0], 1 )

        grad_u_hat = torch.autograd.grad(u_hat, interior,
           grad_outputs=inputs, create_graph=True)[0]
        u_x = grad_u_hat[:, 0]
        u_y = grad_u_hat[:, 1]

        # Greater Derivatives

        grad_grad_u_x = torch.autograd.grad(u_x, interior,
           grad_outputs=torch.ones(interior.shape[0]),
           create_graph=True)[0]
        grad_grad_u_y = torch.autograd.grad(u_y, interior,
           grad_outputs=torch.ones(interior.shape[0]),
           create_graph=True)[0]
        u_xx = grad_grad_u_x[:, 0]
        u_xy = grad_grad_u_x[:, 1]
        u_yy = grad_grad_u_y[:, 1]

        #.....................................................................


        # H O L E   B O R D E R   P O I N T S

        #.....................................................................
```

```
hpl = hpl.clone().requires_grad_(True)
hpr = hpr.clone().requires_grad_(True)
hpb = hpb.clone().requires_grad_(True)
hpt = hpt.clone().requires_grad_(True)

u_bd_left_pred  = self.model(hpl)
u_bd_right_pred = self.model(hpr)
u_bd_top_pred   = self.model(hpt)
u_bd_bot_pred   = self.model(hpb)

# Greater Derivatives

# Need slope to determine the rotation
u_b_l = torch.autograd.grad(u_bd_left_pred, hpl,
   grad_outputs= torch.ones_like(u_bd_left_pred),
   create_graph=True)[0]
u_b_l_x = u_b_l[:, 0]
u_b_l_y = u_b_l[:, 1]

u_b_r = torch.autograd.grad(u_bd_right_pred, hpr,
   grad_outputs= torch.ones_like(u_bd_right_pred),
   create_graph=True)[0]
u_b_r_x = u_b_r[:, 0]
u_b_r_y = u_b_r[:, 1]

u_b_b = torch.autograd.grad(u_bd_bot_pred, hpb,
   grad_outputs= torch.ones_like(u_bd_bot_pred),
   create_graph=True)[0]
u_b_b_x = u_b_b[:, 0]
u_b_b_y = u_b_b[:, 1]

u_b_t = torch.autograd.grad(u_bd_top_pred, hpt,
   grad_outputs= torch.ones_like(u_bd_top_pred),
   create_graph=True)[0]
u_b_t_x = u_b_t[:, 0]
u_b_t_y = u_b_t[:, 1]

# Rotation

u_b_l_xx = torch.autograd.grad(u_b_l_x, hpl, grad_outputs
   =torch.ones_like(u_b_l_x), create_graph=True)[0][:, 0]
u_b_r_xx = torch.autograd.grad(u_b_r_x, hpr, grad_outputs
   =torch.ones_like(u_b_r_x), create_graph=True)[0][:, 0]
u_b_l_yy = torch.autograd.grad(u_b_l_y, hpl, grad_outputs
   =torch.ones_like(u_b_l_y), create_graph=True)[0][:, 1]
u_b_r_yy = torch.autograd.grad(u_b_r_y, hpr, grad_outputs
   =torch.ones_like(u_b_r_y), create_graph=True)[0][:, 1]
u_b_t_xx = torch.autograd.grad(u_b_t_x, hpt, grad_outputs
```

```
   = torch.ones_like(u_b_t_x), create_graph=True)[0][:, 0]
u_b_b_xx = torch.autograd.grad(u_b_b_x, hpb, grad_outputs
   = torch.ones_like(u_b_b_x), create_graph=True)[0][:, 0]
u_b_t_yy = torch.autograd.grad(u_b_t_y, hpt, grad_outputs
   = torch.ones_like(u_b_t_y), create_graph=True)[0][:, 1]
u_b_b_yy = torch.autograd.grad(u_b_b_y, hpb, grad_outputs
   = torch.ones_like(u_b_b_y), create_graph=True)[0][:, 1]

# Shear

u_b_l_xxx = torch.autograd.grad(u_b_l_xx, hpl,
   grad_outputs=torch.ones_like(u_b_l_xx), create_graph=
   True)[0][:, 0]
u_b_r_xxx = torch.autograd.grad(u_b_r_xx, hpr,
   grad_outputs=torch.ones_like(u_b_r_xx), create_graph=
   True)[0][:, 0]
u_b_l_yyx = torch.autograd.grad(u_b_l_yy, hpl,
   grad_outputs=torch.ones_like(u_b_l_yy), create_graph=
   True)[0][:, 0]
u_b_r_yyx = torch.autograd.grad(u_b_r_yy, hpr,
   grad_outputs=torch.ones_like(u_b_r_yy), create_graph=
   True)[0][:, 0]
u_b_t_xxy = torch.autograd.grad(u_b_t_xx, hpt,
   grad_outputs=torch.ones_like(u_b_t_xx), create_graph=
   True)[0][:, 1]
u_b_b_xxy = torch.autograd.grad(u_b_b_xx, hpb,
   grad_outputs=torch.ones_like(u_b_b_xx), create_graph=
   True)[0][:, 1]
u_b_t_yyy = torch.autograd.grad(u_b_t_yy, hpt,
   grad_outputs=torch.ones_like(u_b_t_yy), create_graph=
   True)[0][:, 1]
u_b_b_yyy = torch.autograd.grad(u_b_b_yy, hpb,
   grad_outputs=torch.ones_like(u_b_b_yy), create_graph=
   True)[0][:, 1]


#.....................................................................


# L O S S

#.....................................................................



#.....................................................................

# Rayleigh Quotient
#.....................................................................
```

```
U = ((u_xx + u_yy)**2 - 2*(1-self.v)*(u_xx*u_yy - u_xy
   **2))

V = (u_x)**2

# Specify dim=0 guarantees the reduction is over the
   interior-point dimension of the geometry
U_mean = U.mean(dim=0)
V_mean = V.mean(dim=0)
R = (U_mean)/(V_mean + 1e-8)

#G = u_hat.t() @ u_hat
#loss_ortho = ((G - torch.eye(G.shape[0], device=G.device
   ))**2).sum()

loss_pde = R #+ loss_ortho

# Make N_true a tensor on the same device/dtype as R
N_true_t = torch.tensor(
N_true,
device=R.device,
dtype=R.dtype
)

# Connect the capacity output to the network
D_t = torch.tensor(self.D, device=R.device, dtype=R.dtype
   )
Capacity = R * D_t

loss_cap = 0.5 * ((Capacity - N_true_t)**2)

#.....................................................................

# Hole Boundary
#.....................................................................


loss_hb_x = (

    # Shear condition = 0 on all free edges

    torch.mean((u_b_l_xxx.reshape(-1,) + (2 - self.v)*
       u_b_l_yyx.reshape(-1,))**2) +
    torch.mean((u_b_r_xxx.reshape(-1,) + (2 - self.v)*
       u_b_r_yyx.reshape(-1,))**2) +

    # Moment condition = 0 on all free edges
```

```
            torch.mean((u_b_l_xx.reshape(-1,) + self.v*(u_b_l_yy.
               reshape(-1,)))**2) +
            torch.mean((u_b_r_xx.reshape(-1,) + self.v*(u_b_r_yy.
               reshape(-1,)))**2)

            )

        loss_hb_y = (

            # Shear condition = 0 on all free edges

            torch.mean((u_b_b_yyy.reshape(-1,) + (2 - self.v)*
               u_b_b_xxy.reshape(-1,))**2) +
            torch.mean((u_b_t_yyy.reshape(-1,) + (2 - self.v)*
               u_b_t_xxy.reshape(-1,))**2) +

            # Moment condition = 0 on all free edges

            torch.mean((u_b_t_yy.reshape(-1,) + self.v*(u_b_t_xx.
               reshape(-1,)))**2) +
            torch.mean((u_b_b_yy.reshape(-1,) + self.v*(u_b_b_xx.
               reshape(-1,)))**2)

            )

        loss_hole = loss_hb_x + loss_hb_y

        #.....................................................................

        # Adaptive Weights
        #.....................................................................


        # If NOT using class ReLoBRaLoLoss(nn.Module)
        #loss = loss_pde + loss_cap + loss_hole

        # If using class ReLoBRaLoLoss(nn.Module)
        losses_raw = [loss_pde , loss_hole, loss_cap]
        loss = self.RELO(epoch, losses_raw, dout)

        return loss

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Note that as using LBFGS, there must be a closure call.
# This then determines the weights by calling forward_loss.
# These are back-propogated.
# This process is repeated for each epoch in step.
```

```
# The history and total loss at each epoch is stored in a
   dataframe (dout).
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
    def make_closure(self, epoch, dout):

        def closure():
            # zero gradients
            self.optimizer.zero_grad()
            total_loss = torch.tensor(0, device=next(self.model.
               parameters()).device)
            total_shape = 0

            for i, (interior, hpl, hpr, hpb, hpt) in enumerate(
               zip(self.interiors, self.hole_l, self.hole_r, self.
               hole_b, self.hole_t)):

                interior = interior.clone().requires_grad_(True)
                N_true = self.capacities[i]
                loss_i = self._forward_loss(epoch, dout, interior
                   , hpl, hpr, hpb, hpt, N_true)

                total_loss = total_loss + loss_i
                total_shape = total_shape + interior.shape[0]

            mean_total_loss = total_loss / total_shape

            mean_total_loss.backward()

            self._running_loss += float(mean_total_loss.item())

            return mean_total_loss

        return closure

    def step(self, epoch, dout):

        # reset running-loss accumulator for this epoch
        self._running_loss = 0.0

        closure = self.make_closure(epoch, dout)

        self.optimizer.step(closure=closure)

        self.history.append(self._running_loss)

        return self._running_loss, self.history

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
    - - - - - - #
# This is used to plot the deformed shape, and compute the
   predicted perforated capacity (P_pr)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def Plot_Prediction(FILE, DIRP, n_hidden_layers, n_neurons, D, v,
    a, b, m, n, c_x, c_y, l_x, l_y, n_x, n_y):
    rseed = 31

    n_pts = n_x*n_y
    oversample = 1.2
    plotxy, n_hb, border_pts_left, border_pts_right,
       border_pts_bot, border_pts_top = Get_Hole_Border_Points(a,
       b, m, n, l_x, l_y, c_x, c_y, n_pts, 4*n_pts, oversample,
       rseed)

    plot_hole_pts(a, b, c_x, c_y, l_x, l_y, plotxy, n_hb,
       border_pts_left, border_pts_right, border_pts_bot,
       border_pts_top)

    n = plotxy.shape[0]
    geom_cols = torch.tensor([[l_x, l_y, c_x, c_y]], dtype=plotxy
       .dtype).repeat(n, 1)
    plotxy_cond = torch.cat([plotxy, geom_cols], dim=1).float().
       detach().requires_grad_(True)

    # uncomment below when you need to test for different points
    model = RayleighPINN(N_modes=1, a=a, b=b, input_dimension=6,
       output_dimension=1, n_hidden_layers=n_hidden_layers,
       neurons=n_neurons)
    model.load_state_dict(torch.load(DIRP + FILE + "_model.pth"))
    model.eval()
    u_hat = model(plotxy_cond)   # predicted output

    # Compute predicted perforated capacity
    P_pr = predict_capacity(u_hat, plotxy_cond, D, v)
    print(f"Predicted Critical Force = {P_pr} N")

    Title = f"c_x = {c_x}, c_y = {c_y} \n l_x = {l_x}, l_y = {l_y
       }"
    Predictor(D, v, a, b, m, n, c_x, c_y, l_x, l_y).
       Plot_Predicted_Displacement(n_x, n_y, plotxy, u_hat, Title)

    return P_pr
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Similar to Plot_Prediction, however it does not return a
   prediction for the capacity
# This is because many fewer points are assumed by default, in
```

```
   order to create a more appealing image.
# This is 'more appealing' as it is set to a particular view, the
    grid is not showing, and a pleasant colour scheme is chosen.
# Fewer points are needed here, in order to create a mesh that is
    not overcrowded, overlayed on the surface output.
# The image is saved to the desired location
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def Plot_Prediction_V2(VIEW, FILE, DIRP, DIRI, NAME,
   n_hidden_layers, n_neurons, D, v, a, b, m, n, c_x, c_y, l_x,
   l_y, n_x, n_y):

    xss = np.linspace(0, a, n_x)
    yss = np.linspace(0, b, n_y)
    X, Y = np.meshgrid(xss, yss)

    pts_numpy = np.array([(x, y) for y in yss for x in xss])
    pts = torch.from_numpy(pts_numpy)

    mask = (pts[:,0] < c_x - l_x/2) | (pts[:,0] > c_x + l_x/2) |
       (pts[:,1] < c_y - l_y/2) | (pts[:,1] > c_y + l_y/2)

    plotxy_s = pts[mask]
    plt.scatter(plotxy_s[:,0], plotxy_s[:,1], s=2, alpha=0.2)

    n = plotxy_s.shape[0]
    geom_cols = torch.tensor([[l_x, l_y, c_x, c_y]], dtype=
       plotxy_s.dtype).repeat(n, 1)
    plotxy_cond_s = torch.cat([plotxy_s, geom_cols], dim=1).float
       ().detach().requires_grad_(True)

    model = RayleighPINN(N_modes=1, a=a, b=b, input_dimension=6,
       output_dimension=1, n_hidden_layers=n_hidden_layers,
       neurons=n_neurons)
    model.load_state_dict(torch.load(DIRP + FILE + "_model.pth"))
    model.eval()
    u_hat_s = model(plotxy_cond_s)   # predicted output

    xsw = np.linspace(0, a, 25)
    ysw = np.linspace(0, b, 25)
    X, Y = np.meshgrid(xsw, ysw)

    pts_numpy = np.array([(x, y) for y in ysw for x in xsw])
    pts = torch.from_numpy(pts_numpy)

    mask = (pts[:,0] < c_x - l_x/2) | (pts[:,0] > c_x + l_x/2) |
       (pts[:,1] < c_y - l_y/2) | (pts[:,1] > c_y + l_y/2)

    plotxy_w = pts[mask]
```

```
    n = plotxy_w.shape[0]
    geom_cols = torch.tensor([[l_x, l_y, c_x, c_y]], dtype=
       plotxy_w.dtype).repeat(n, 1)
    plotxy_cond_w = torch.cat([plotxy_w, geom_cols], dim=1).float
       ().detach().requires_grad_(True)
    u_hat_w = model(plotxy_cond_w)   # predicted output

    Title = f"c_x = {c_x}, c_y = {c_y} \n l_x = {l_x}, l_y = {l_y
       }"
    Predictor(D, v, a, b, m, n, c_x, c_y, l_x, l_y).
       Plot_Predicted_Displacement_V2(n_x, n_y, plotxy_s, u_hat_s,
        len(xsw), len(ysw), plotxy_w, u_hat_w, Title, DIRI, NAME,
       VIEW)
```

J.2.3 Prediction

```
# -*- coding: utf-8 -*-
"""
Created on Fri Nov 14 14:35:45 2025

@author: eleno
"""

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# This outputs two predicted displacement functions, which are
   used by either of the two plot_prediction functions in
   Parametric_PINNS_Functions
# See the explanation in Parametric_PINNS_Functions to understand
    why each are necessary.
# Essentialy, Plot_Predicted_Displacement is used to determine
   P_cr whereas Plot_Predicted_Displacement_V2 is used to output
   and save an image.
# Plot_Predicted_Displacement was not developed with formatting
   and making appealing images in mind! Hence why
   Plot_Predicted_Displacement_V2 was created.
# However, Plot_Predicted_Displacement_V2 cannot be used to
   accurately determine P_cr, as it uses many fewer points, so
   that the wireframe is not overcrowded.
# Plot_Predicted_Displacement_V2 can output and save images to
   the desired location.
# The user can change the view from 1 to 4 which changes the
```

```
   position of the 'camera'.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
class Predictor:

    def __init__(self, D, v, a, b, m, n, c_x, c_y, l_x, l_y):

        self.D = D
        self.v = v
        self.a = a
        self.b = b
        self.m = m
        self.n = n
        self.c_x = c_x
        self.c_y = c_y
        self.l_x = l_x
        self.l_y = l_y

    def Plot_Predicted_Displacement(self, nx, ny,
       InputCoordinates, DisplacementField, Title):

        #.....................................................................

        # assume plotxy is (n_pts,2) and u_hat is (n_pts,1) torch
            tensor
        pts = InputCoordinates.numpy()                  # (n_pts
           ,2)
        vals = DisplacementField.detach().cpu().numpy().ravel()
             # (n_pts,)

        # define regular grid
        x_vals = np.linspace(0, self.a, nx)
        y_vals = np.linspace(0, self.b, ny)
        Xg, Yg = np.meshgrid(x_vals, y_vals)

        print("pts.shape:", pts.shape)
        print("vals.shape:", vals.shape)
        print("plotxy.shape:", InputCoordinates.shape)

        # interpolate (requires scipy)
        from scipy.interpolate import griddata
        Ug = griddata(pts, vals, (Xg, Yg), method='linear')   #
           result has shape (ny, nx)

        # mask hole
        mask = ~((np.abs(Xg - self.c_x) < self.l_x/2) & (np.abs(
           Yg - self.c_y) < self.l_y/2))
        Ug_masked = np.where(mask, Ug, np.nan)
        u_max = np.nanmax(np.abs(Ug_masked))
```

```
        # plot
        fig = plt.figure(figsize=(6,5))
        ax = fig.add_subplot(111, projection='3d')
        ax.plot_surface(Xg, Yg, Ug_masked / u_max, cmap=cm.
           coolwarm, alpha=0.6)
        ax.view_init(elev=0, azim=-90, roll=0)
        ax.set_title(Title)
        plt.show()

    def Plot_Predicted_Displacement_V2(self, nxs, nys,
       InputCoordinates_s, DisplacementField_s, nxw, nyw,
       InputCoordinates_w, DisplacementField_w, Title, DIRI, NAME,
        VIEW):
        fig = plt.figure(figsize=(6,5))
        ax = fig.add_subplot(111, projection='3d')

        # SURFACE
        #.....................................................................

        # assume plotxy is (n_pts,2) and u_hat is (n_pts,1) torch
            tensor
        pts = InputCoordinates_s.numpy()                # (n_pts
           ,2)
        vals = DisplacementField_s.detach().cpu().numpy().ravel()
              # (n_pts,)

        # define regular grid
        x_vals = np.linspace(0, self.a, nxs)
        y_vals = np.linspace(0, self.b, nys)
        Xg, Yg = np.meshgrid(x_vals, y_vals)

        print("pts.shape:", pts.shape)
        print("vals.shape:", vals.shape)
        print("plotxy.shape:", InputCoordinates_s.shape)

        # interpolate (requires scipy)
        from scipy.interpolate import griddata
        Ug = griddata(pts, vals, (Xg, Yg), method='linear')   #
           result has shape (ny, nx)

        # mask hole
        mask = ~((np.abs(Xg - self.c_x) < self.l_x/2) & (np.abs(
           Yg - self.c_y) < self.l_y/2))
        Ug_masked = np.where(mask, Ug, np.nan)
        u_max = np.nanmax(np.abs(Ug_masked))

        # plot surface
        ax.plot_surface(Xg, Yg, Ug_masked / u_max, cmap = plt.cm.
```

```
    cividis, alpha=0.6)
ax.set_box_aspect([self.a, self.b, 1])  # keep aspect
    ratio

# WIREFRAME
#.....................................................................

# assume plotxy is (n_pts,2) and u_hat is (n_pts,1) torch
    tensor
pts = InputCoordinates_w.numpy()                  # (n_pts
    ,2)
vals = DisplacementField_w.detach().cpu().numpy().ravel()
      # (n_pts,)

# define regular grid
x_vals = np.linspace(0, self.a, nxw)
y_vals = np.linspace(0, self.b, nyw)
Xg, Yg = np.meshgrid(x_vals, y_vals)

print("pts.shape:", pts.shape)
print("vals.shape:", vals.shape)
print("plotxy.shape:", InputCoordinates_w.shape)

# interpolate (requires scipy)
from scipy.interpolate import griddata
Ug = griddata(pts, vals, (Xg, Yg), method='linear')   #
    result has shape (ny, nx)

# mask hole
mask = ~((np.abs(Xg - self.c_x) < self.l_x/2) & (np.abs(
    Yg - self.c_y) < self.l_y/2))
Ug_masked = np.where(mask, Ug, np.nan)
u_max = np.nanmax(np.abs(Ug_masked))

# plot surface
ax.plot_wireframe(Xg, Yg, Ug_masked / u_max, color="grey
    ", linewidth=0.25)
ax.set_box_aspect([self.a, self.b, 1])  # keep aspect
    ratio

ax.grid(False)
ax.tick_params(labelsize=12)
ax.set_xticks(np.linspace(0, self.a, 3))
ax.set_yticks(np.linspace(0, self.b, 3))
ax.set_zticks(np.linspace(-1, 1, 5))

ax.set_xlabel('x', fontsize = 12, fontfamily = "Cambria",
     labelpad = 10)
ax.set_ylabel('y', fontsize = 12, fontfamily = "Cambria",
```

```
            labelpad = 10)
        ax.set_zlabel(r'$u_{pred}(x, y)$', fontsize = 12,
           fontfamily = "Cambria", labelpad = 10)

        #ax.set_title(f'PINN Predicted Surface, n.Epochs = {
           n_epochs}')
        if VIEW == 1:
            ax.view_init(elev=-165, azim=30)
            extn = "V1"
        elif VIEW == 2:
            ax.view_init(elev=30, azim=-50)
            extn = "V2"
        elif VIEW == 3:
            ax.view_init(elev=210, azim=50)
            extn = "V3"
        elif VIEW == 4:
            ax.view_init(elev=30, azim=130)
            extn = "V4"

        ax.axis('off')

        plt.tight_layout()
        plt.savefig(DIRI+NAME+ extn + ".jpg", dpi=1500)
        plt.show()
```

J.2.4 ReLoBRaLo V2

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan  6 18:28:52 2026

@author: eleno
"""

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# See the paper:
#   @article{Bischof_Kraus_2025,
#   title={Multi-Objective Loss Balancing for Physics-Informed
   Deep Learning},
#   volume={439},
#   ISSN={0045-7825},
#   url={http://dx.doi.org/10.1016/j.cma.2025.117914},
#   DOI={10.1016/j.cma.2025.117914},
#   journal={Computer Methods in Applied Mechanics and
   Engineering},
#   publisher={Elsevier BV},
```

```
#   author={Bischof, Rafael and Kraus, Michael A.},
#   year={2025},
#   month=may, pages={117914}
#}
# Which showcases Relative lookbacks, and the blog post here:
# https://medium.com/data-science/improving-pinns-through-
   adaptive-loss-balancing-55662759e701
# This defines what alpha, temperature and rho are, and how they
   control the Relative Loss Balancing with Random Lookbacks
   scheme (ReLoBRaLo).
# The blog post provides code by the authors for ReLoBRaLo in
   tensor_flow.
# I modified this to torch, and so that it can be implemented in
   my code.
# The physics, hole and capacity losses are determined, and saved
    in the dataframe.
# The loss is returned (see forward_loss in the fit_rq of the
   Parametric_PINNS_Functions.py).
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #

import torch

EPS = 1e-8

import torch.nn as nn

class ReLoBRaLoLoss(nn.Module):

    def __init__(self, number_of_loss_terms, alpha:float=0.999,
       temperature:float=0.1, rho:float=0.99):
        # super() returns a temporary object from the superclass
        # This allows the subclass to inherit and extend the
           initialisation behaviour of the parent class without
           rewriting it

        super().__init__()
        self.alpha = alpha
        self.temperature = temperature
        self.rho = rho
        self.number_of_loss_terms = number_of_loss_terms

        # Create lists of variables which won't be updated during
            training
        self.lambdas = [torch.tensor(1.0) for _ in range(
           number_of_loss_terms)]
        self.init_losses = [torch.tensor(1.0) for _ in range(
           number_of_loss_terms)]
        self.last_losses = [torch.tensor(1.0) for _ in range(
```

```
            number_of_loss_terms)]

    def forward(self, epoch, losses, dout):

        print(epoch)

        alpha = 1 if epoch == 0 else 0 if epoch == 1 else self.
            alpha
        rho = 1 if epoch == 0 else 1 if epoch == 1 else self.rho

        # Compute new lambdas wrt the losses in the PREVIOUS
            iteration
        # torch.stack ensures that lambdas_hat is a torch.tensor
        lambdas_hat = torch.stack([losses[i]/(self.last_losses[i
            ]*self.temperature + EPS) for i in range(self.
            number_of_loss_terms)])
        lambdas_hat = torch.softmax(lambdas_hat - torch.max(
            lambdas_hat), dim=0) * torch.tensor(len(losses), dtype=
            torch.float)

        # Compute new lambdas wrt the losses in the FIRST
            iteration
        init_lambdas_hat = torch.stack([losses[i]/(self.
            init_losses[i]*self.temperature + EPS) for i in range(
            self.number_of_loss_terms)])
        init_lambdas_hat = torch.softmax(init_lambdas_hat - torch
            .max(init_lambdas_hat), dim=0) * torch.tensor(len(
            losses), dtype=torch.float)

        # Use rho for deciding whether a random lookback should
            be performed
        new_lambdas = [(rho*alpha*self.lambdas[i] + (1-rho)*alpha
            *init_lambdas_hat[i] + (1-alpha)*lambdas_hat[i]) for i
            in range(len(losses))]
        for var, nl in zip(self.lambdas, new_lambdas):
            var.data = nl.detach()

        for var, loss in zip(self.last_losses, losses):
            var.data = loss.detach()

        if epoch == 0:
            # This updates the elements in the init_losses list
                in place
            for var, loss in zip(self.init_losses, losses):
                var.data = loss.detach()

        loss = torch.sum(torch.stack([lam*loss for lam, loss in
            zip(self.lambdas, losses)]))
```

```
        dout["Rayleigh Quotient Loss"].append(self.lambdas[0].
           item()*losses[0].item())
        dout["Hole Loss"].append(self.lambdas[1].item()*losses
           [1].item())
        dout["Capacity Loss"].append(self.lambdas[2].item()*
           losses[2].item())
        dout["Total Loss"].append(loss.item())
        dout["Epoch"].append(epoch)

        # Inspect raw loss magnitudes
        print("Loss magnitudes:")
        print("  loss_pde       :", self.lambdas[0].item()*losses
           [0].item())
        print("  loss_hole      :", self.lambdas[1].item()*losses
           [1].item())
        print("  loss_cap       :", self.lambdas[2].item()*losses
           [2].item())

        return loss
```

J.2.5 Utilities

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 19 09:50:10 2025

@author: eleno
"""
import torch
import torch.nn as nn
import numpy as np
import matplotlib.pyplot as plt

#.....................................................................


                #   F U N C T I O N S / C L A S S E S  #

#.....................................................................


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Determines m and n of a buckling plate, according to its aspect
    ratio
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
```

```
def halfwaves(ar):

    CPOINTS = []
    MS = list(range(1,50))

    for mi in MS:
        CP = np.sqrt(1/((1/mi)-(1/(mi+1))))
        CPOINTS.append(CP)

    m = 1

    while ar>=CPOINTS[m-1]:
        m +=1

    return(m)

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Builds a feed-forward neural network with the number of hidden
   layers and neurons per layers specified by the user
# Tanh activation on all layers
# Hard-codes the boundary conditions on the output from the
   forward function
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
class RayleighPINN(nn.Module):

    def __init__(self, N_modes, a, b, input_dimension,
       output_dimension, n_hidden_layers, neurons):
        super(RayleighPINN, self).__init__()
        # Number of modes wish to find
        self.N_modes = N_modes
        self.a = a # Length of plate
        self.b = b # Width of plate

        # Number of input dimensions n
        self.input_dimension = input_dimension
        # Number of output dimensions m
        self.output_dimension = output_dimension
        # Number of neurons per layer
        self.neurons = neurons
        # Number of hidden layers
        self.n_hidden_layers = n_hidden_layers
        # Activation function
        self.activation = nn.Tanh()

        self.input_layer = nn.Linear(self.input_dimension, self.
           neurons)
```

```
        self.hidden_layers = nn.ModuleList([nn.Linear(self.
           neurons, self.neurons) for _ in range(n_hidden_layers)
           ])
        self.output_layer = nn.Linear(self.neurons, self.
           output_dimension)

    def forward(self, x):
        # Includes hard-coding of the boundary conditions.
        # This is necessary in order for the Rayleigh QUotient to
            make sense, and it follows that the Neumann conditions
            will be automatically satsified and should NOT be
           softly imposed

        # x: (batch,2) with columns [x,y]
        x_in = self.activation(self.input_layer(x))
        for L in self.hidden_layers:
            x_in = self.activation(L(x_in))
        N_out = self.output_layer(x_in)   # shape (batch, N_modes
           )

        xi = x[:,0:1]   # (batch,1)
        yi = x[:,1:2]   # (batch,1)
        Phi = xi*(self.a - xi) * yi*(self.b - yi)  # (batch,1)

        # if N_modes>1, N_out is (batch,N_modes); broadcast Phi
        return Phi * N_out

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# Initialisation using Xavier with tanh activation
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
def init_xavier(model, retrain_seed):
    torch.manual_seed(retrain_seed)
    def init_weights(m):
        if type(m) == nn.Linear and m.weight.requires_grad and m.
           bias.requires_grad:
            g = nn.init.calculate_gain('tanh')
            torch.nn.init.xavier_uniform_(m.weight, gain=g)
            m.bias.data.fill_(0)
    model.apply(init_weights)

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# This receives the predicted displacement field, and determines
   the appropriate variables
# to calculate the internal strain energy (U) and external work
   done (V) for the Rayleigh quotient.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
    - - - - - - #
def predict_capacity(u_hat, XY, D, v):
    # Compute gradients
    grad_u_hat = torch.autograd.grad(u_hat, XY,
                                     grad_outputs=torch.ones_like
                                        (u_hat),
                                     create_graph=True)[0]
    u_x = grad_u_hat[:, 0]
    u_y = grad_u_hat[:, 1]

    # Second derivatives
    grad_grad_u_x = torch.autograd.grad(u_x, XY,
                                        grad_outputs=torch.
                                           ones_like(u_x),
                                        create_graph=True)[0]
    grad_grad_u_y = torch.autograd.grad(u_y, XY,
                                        grad_outputs=torch.
                                           ones_like(u_y),
                                        create_graph=True)[0]
    u_xx = grad_grad_u_x[:, 0]
    u_xy = grad_grad_u_x[:, 1]
    u_yy = grad_grad_u_y[:, 1]

    # Strain energy term
    U = ((u_xx + u_yy)**2 - 2*(1-v)*(u_xx*u_yy - u_xy**2))
    # External work term
    V = u_x**2

    # Rayleigh quotient
    R = torch.sum(U).item()/torch.sum(V).item()
    #R = ((U.nanmean()) / (V.nanmean() + 1e-8)).item()  # nanmean
        ignores masked NaNs

    return (R * D)

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    - - - - - - #
# This displays the points around the perforation so that the
   user can visualise this.
# If unhappy with the point selection, see Math_Disp >>
   Get_Hole_Border_Points and
# change the definition for n_hb to select more or less points
# (see the comment before that funciton, which explains that n_hb
    was determined
# through trial-and-error for the particular problem examined in
   this paper).
# The points are plotted and can be 'zoomed' into for closer
   examination.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
    - - - - - - #
def plot_hole_pts(a, b, c_x, c_y, l_x, l_y, interior, n_hb,
   border_pts_left, border_pts_right, border_pts_bot,
   border_pts_top):

    border_pts_all_display = torch.cat([border_pts_left,
       border_pts_right, border_pts_bot, border_pts_top], dim=0)
         # shape (K_left+K_right+K_bot+K_top, 2)
    border_pts_all_display = torch.unique(border_pts_all_display,
        dim=0)

    #Plot to verify
    plt.figure(figsize=(4,4))
    plt.scatter(interior[:,0], interior[:,1], s=2, alpha=0.2,
       label='interior')
    plt.scatter(border_pts_all_display[:,0],
       border_pts_all_display[:,1], s=20, c='r', label=f'N={n_hb}
       closest')
    #Draw the hole
    rect = plt.Rectangle((c_x - l_x/2, c_y-l_y/2), l_x, l_y, fill
       =False, edgecolor='k')
    plt.gca().add_patch(rect)
    plt.gca().set_aspect('equal')
    plt.xlim(0, a)
    plt.ylim(0, b)
    plt.show()

    # Zoom in on hole
    plt.figure(figsize=(4,4))
    plt.scatter(interior[:,0], interior[:,1], s=2, alpha=0.2,
       label='interior')
    plt.scatter(border_pts_all_display[:,0],
       border_pts_all_display[:,1], s=20, c='r', label=f'N={n_hb}
       closest')
    rect = plt.Rectangle((c_x - l_x/2, c_y-l_y/2), l_x, l_y, fill
       =False, edgecolor='k')
    plt.gca().add_patch(rect)
    plt.gca().set_aspect('equal')
    plt.xlim(c_x - l_x, c_x + l_x)
    plt.ylim(c_y - l_y, c_y + l_y)
    plt.show()
```

References

[1] European Committee for Standardization. *Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings*. Standard EN 1993-1-1. Brussels, Belgium: European Committee for Standardization, 2003.

[2] C. D. Moen and B. W. Schafer. "Elastic Buckling of thin plates with holes in compression or bending". In: *Thin-Walled Structures* 47.12 (2009), pp. 1597–1607. URL: http://dx.doi.org/10.1016/j.tws.2009.05.001.

[3] F. Smith and C. D. Moen. "Finite strip elastic buckling solutions for thin-walled metal columns with perforation patterns". In: *Thin-Walled Structures* 2014.79 (2014), pp. 187–201. URL: http://dx.doi.org/10.1016/j.tws.2014.02.009.

[4] I. C. Scheperboer, E. Efthymiou, and J. Maljaars. "Local buckling of aluminium and steel plates with multiple holes". In: *Thin-Walled Structures* 2016.99 (2016), pp. 132–141. URL: http://dx.doi.org/10.1016/j.tws.2015.11.009.

[5] Z. Yao and K. J. Rasmussen. "Perforated Cold-Formed Steel Members in Compression. II: Design". In: *Journal of Structural Engineering* 143.5 (2017). URL: http://dx.doi.org/10.1061/(asce)st.1943-541x.0001636.

[6] European Committee for Standardization. *Eurocode 3: Design of steel structures - Part 1-13: Beams with large web openings*. Standard EN 1993-1-13:2024. Brussels, Belgium: European Committee for Standardization, 2003.

[7] American Iron, Steel Institute, and CSA Group. *North American Specification for the Design of Cold-Formed Steel Structural Members: 2016 Edition (Reaffirmed 2020) With Supplement 3, 2022 Edition*. Standard AISI S100-16 (2020) w/S3-22. USA: American Iron and Steel Institute, 2022.

[8] S. W.-K. S. Timoshenko. *Theory of Plates and Shells*. 2nd ed. ed., New York: McGraw-Hill, 1959.

[9] S. Timoshenko and J. Gere. *Theory of Elastic Stability*. 2nd ed. Singapore: McGraw-Hill Book Company, 1963.

[10] C. L. Navier. "Extrait des recherches sur la flexion des plans élastiques". In: *Société Philomathique de Paris* 22 (1823), pp. 92–102. URL: https://archive.org/details/bulletindesscien22soci/page/n73/mode/1up.

[11] Unknown. *Strut channel image*. Accessed: 2026-03-19. n.d. URL: https://5.imimg.com/data5/LF/KW/MY-3664113/strurt-channel-1000x1000.jpg.

[12] Unknown. *Unistrut channel image*. Accessed: 2026-03-19. 2023. URL: https://5.imimg.com/data5/SELLER/Default/2023/6/320069833/LX/PO/YK/3170315/unistrut-channel-500x500.jpg.

[13] Swastik Lugs. *PVC channel (Rolycab)*. Accessed: 2026-03-19. 2022. URL: https://swastiklugs.in/wp-content/uploads/2022/08/High-qulaity-image-Rolycab-PVC-Channel.webp.

[14] Tube Pierce Manufacturing. *Industries Served*. Accessed: 2026-03-19. n.d. URL: https://tubepiercemfg.com/industries-served/.

[15] Great Locker. *Industrial steel slotted angle shelving*. Accessed: 2026-03-19. n.d. URL: https://www.greatlocker.com/uploads/201715378/industrial-steel-slotted-angle-iron-shelving52170919232.jpg.

[16] Unknown. *Slotted angle image*. Accessed: 2026-03-19. n.d. URL: https://5.imimg.com/data5/SY/RA/WH/SELLER-98873689/sloted-angle-500x500.jpg.

[17] Steel Construction Institute. *Figure K1 Fig18*. https://steelconstruction.info/File:K1_Fig18.png. Accessed: 2026-03-26. n.d.

[18] European Committee for Standardization. *Eurocode 3: Design of steel structures - Part 1-5: General rules and rules for buildings*. Standard EN 1993-1-1:2005. Brussels, Belgium: European Committee for Standardization, 2005.

[19] Chinese National Standard. *Technical Code for Cold-Formed Thin-Walled Steel Structures*. Standard GB 50017-2017. Beiging, China: Chinese National Standard, 2017.

[20] T. H. Miller and T. Peköz. "Behavior of Cold-Formed Steel Wall Stud Assemblies". In: *Journal of Structural Engineering* 119.2 (1993), pp. 641–651. URL: https://doi.org/10.1061/(ASCE)0733-9445(1993)119:2(641).

[21] T. H. Miller and T. Peköz. "Unstiffened Strip Approach for Perforated Wall Studs". In: *Journal of Structural Engineering* 120.2 (1994), pp. 410–421. URL: https://doi.org/10.1061/(ASCE)0733-9445(1994)120:2(410).

[22] Joint Technical Committee BD-082. *Cold-formed steel structures*. Standard AS/NZS 4600:2018. Sydney / Wellington: Standards Australia / Standards New Zealand, 2018.

[23] E. Naraidoo and B. Rossi. "Local buckling load of a perforated plate - A computational study". In: *Thin-Walled Structures* 215, Part A (Oct. 2025). URL: http://dx.doi.org/10.1016/j.tws.2025.113445.

[24] E. Naraidoo and B. Rossi. "The Direct Strength Method for Plates with Perforations". In: *Journal of Structural Engineering* Forthcoming (Accepted for publication) (2026).

[25] E. Naraidoo and B. Rossi. "An experimental, numerical and analytical study of local buckling in perforated plates". In: *Thin-Walled Structures* 219 (Feb. 2026), p. 114269. URL: http://dx.doi.org/10.1016/j.tws.2025.114269.

[26] E. Naraidoo, Z. You, and B. Rossi. "Elastic Buckling of Slender Steel Plates wtih Perforations". In: *Nordic Steel*. Luleå, June 2024.

[27] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Cambridge, UK: Cambridge University Press, 1888.

[28] G. Winter. "Commentary on the Strength of Thin Steel Compression Flanges". In: *Journal of the Structural Division, ASCE* 86.ST7 (1960), pp. 1–23.

[29] E. Naraidoo and I. Mirtsopoulos. *Abalone: Grasshopper Plug-in for Rhino*. GitHub repository. Available through Rhino3D PackageManager. 2026. URL: https://github.com/EllieNar/Abalone.

[30] E. Naraidoo and I. Mirtsopoulos. *Abalone: Grasshopper Plug-in for Rhino*. Food4Rhino platform for Rhino/Grasshopper plugins. Accessed: 2026-03-23. 2026. URL: https://www.food4rhino.com/en/app/abalone.

[31] Robert McNeel & Associates. *Rhinoceros 3D and Grasshopper*. 3D modelling software and visual programming plugin. Robert McNeel & Associates. Seattle, WA, USA, 2022.

[32] K. Kathage, T. Misiek, and H. Saal. "Stiffness and critical buckling load of perforated sheeting". In: *Thin-Walled Structures* 2006.44 (2006), pp. 1223–1230. URL: http://dx.doi.org/10.1016/j.tws.2007.01.009.

[33] V. Thevendran N. E. Shanmugam and Y. H. Tan. "Design formula for axially compressed perforated plates". In: *Thin-Walled Structures* 34.1 (1999), pp. 1–20. URL: http://dx.doi.org/10.1016/s0263-8231(98)00052-4.

[34] K. M. El-Sawy, A. S. Nazmy, and M. I. Martini. "Elasto-plastic buckling of perforated plates under uniaxial compression". In: *Thin-Walled Structures* 2004.42 (2004), pp. 1083–1101. URL: http://dx.doi.org/10.1016/j.tws.2004.03.002.

[35] A. M. Sweedan and K. M. El-Sawy. "Elastic local buckling of perforated webs of steel cellular beam-column elements". In: *Journal of Constructional Steel Research* 2011.67 (Feb. 1, 2011), pp. 1115–1127. URL: http://dx.doi.org/10.1016/j.jcsr.2011.02.004.

[36] C. Pellegrino E. Maiorana and C. Modena. "Elastic stability of plates with circular and rectangular holes subjected to axial compression and bending moment". In: *Thin-Walled Structures* 47.3 (2009), pp. 241–255. URL: http://dx.doi.org/10.1016/j.tws.2008.08.003.

[37] T. M. Shakerley and C. J. Brown. "Elastic Buckling of Plates with Eccentrically Positioned Rectangular Perforations". In: *Int. J. Mech. Sci.* 38.1996 (1996), pp. 825–838. URL: http://dx.doi.org/10.1016/0020-7403(95)00107-7.

[38] M. A. Komur and M. Sonmez. "Elastic buckling of rectangular plates under linearly varying in-plane normal load with a circualr cutout". In: *Mechanics Research Communications* 2008.35 (2008), pp. 361–371. URL: http://dx.doi.org/10.1016/j.mechrescom.2008.01.005.

[39] G. Lorenzini et al. "Numerical Evaluation of the Effect of Type and Shape of Perforations on the Buckling of Thin Steel Plates by Means of the Constructal Design Method". In: *International Journal of Heat and Technology* 34 (2016), S9–S20. URL: http://dx.doi.org/10.18280/ijht.34sp0102.

[40] Ahmed A. Matloub et al. "Flexural strength of steel girders with perforated web and tubular compression flange". In: *Proceedings of the Institution of Civil Engineers - Structures and Buildings* 177.8 (Aug. 2024), pp. 651–666. URL: http://dx.doi.org/10.1680/jstbu.23.00010.

[41] S. Adany. "Constrained shell finite element method for thin-walled members with holes". In: *Thin-Walled Structures* 121 (2017), pp. 41–56. URL: http://dx.doi.org/10.1016/j.tws.2017.09.021.

[42] Y. Guo and X. Yao. "Experimental Study and Effective Width Method for Cold-Formed Steel Lipped Channel Stud Columns with Holes". In: *Advances in Civil Engineering* 2021.1 (2021). URL: http://dx.doi.org/10.1155/2021/9949199.

[43] Y. Guo and X. Yao. "Buckling Behaviour and Effective Width Design Method for Thin Plates with Holes under Stress Gradient". In: *Mathematical Problems in Engineering* 2021 (2021). URL: http://dx.doi.org/10.1155/2021/5550749.

[44] J. K. Paik and A. K. Thayamballi. *Ultimate Limit State Design of Steel–Plated Structures*. Hoboken, NJ, USA: John Wiley & Sons, 2018.

[45] Z. P. Bazant and L. Cedolin. *Stability of Structures*. London: World Scientific Publishing Co., 2010.

[46] S. Minera et al. "Buckling-resistant topological design using sensitivities to variations in localised nominal stiffness". In: *Thin-Walled Structures* 167.2021 (2021), p. 108150. URL: http://dx.doi.org/10.1016/j.tws.2021.108150.

[47] G. Eccher, K. J. R. Rasmussen, and R. Zandonini. "Elastic buckling analysis of perforated thin-walled structures by the isoparametric spline finite strip method". In: *Thin-Walled Structures* 46.2 (2008), pp. 165–191.

[48] G. Eccher, K. J. R. Rasmussen, and R. Zandonini. "Linear elastic isoparametric spline finite strip analysis of perforated thin-walled structures". In: *Thin-Walled Structures* 46.3 (2008), pp. 242–260.

[49] G. Eccher, K. J. R. Rasmussen, and R. Zandonini. "Geometric nonlinear isoparametric spline finite strip analysis of perforated thin-walled structures". In: *Thin-Walled Structures* 47.2 (2009), pp. 219–232.

[50] Z. Yao and K. J. R. Rasmussen. "Material and geometric nonlinear isoparametric spline finite strip analysis of perforated thin-walled steel structures—Analytical developments". In: *Thin-Walled Structures* 49.11 (2011), pp. 1359–1373.

[51] Z. Yao and K. J. R. Rasmussen. "Material and geometric nonlinear isoparametric spline finite strip analysis of perforated thin-walled steel structures—Numerical investigations". In: *Thin-Walled Structures* 49.11 (2011), pp. 1374–1391.

[52] Z. Yao and K. J. R. Rasmussen. "Inelastic local buckling behaviour of perforated plates and sections under compression". In: *Thin-Walled Structures* 61 (2012), pp. 49–70.

[53] M. Djelosevic et al. "Mathematical Identification of Influential Parameters on the Elastic Buckling of Variable Geometry Plate". In: *The Scientific World Journal* 2013 (Jan. 2013). URL: http://dx.doi.org/10.1155/2013/268673.

[54] J. B. Gracia and F. G. Rammerstorfer. "Increase in buckling loads of plates by introduction of cutouts". In: *Acta Mech* 239.2019 (2019), pp. 2873–2889. URL: http://dx.doi.org/10.1007/s00707-019-02435-6.

[55] A. J. Osgouei, Y. Hosseinzadeh, and H. Ahmadi. "Local buckling analysis of cold-formed steel webs with stiffened rectangular openings". In: *Journal of Constructional Steel Research* 167 (2020), p. 105824.

[56] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations". In: *CoRR* abs/1711.10561 (2017). arXiv: 1711.10561. URL: http://arxiv.org/abs/1711.10561.

[57] M. H. Hassoun. *Fundamentals of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.

[58] 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". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[59] T. Kapoor et al. “Physics-Informed Neural Networks for Solving Forward and Inverse Problems in Complex Beam Systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.5 (2024), pp. 5981–5995.

[60] Y. Fuh-Gwo et al. “Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020”. In: *Proceedings of SPIE Smart Structures + Nondestructive Evaluation*. Vol. 11379. Online, 2020, p. 1137903.

[61] A. I. F. Al-Adly and P. Kripakaran. “Physics-informed neural networks for structural health monitoring: a case study for Kirchhoff–Love plates”. In: *Data-Centric Engineering* 5 (2024), e6.

[62] R. Bischof and M. A. Kraus. “Multi-Objective Loss Balancing for Physics-Informed Deep Learning”. In: *Computer Methods in Applied Mechanics and Engineering* 439 (May 2025), p. 117914. URL: http://dx.doi.org/10.1016/j.cma.2025.117914.

[63] J-H. Bastek and D. M. Kochmann. “Physics-Informed Neural Networks for shell structures”. In: *European Journal of Mechanics - A/Solids* 97 (2023), p. 104849. URL: https://www.sciencedirect.com/science/article/pii/S0997753822002790.

[64] Z. Guo et al. “Research and software development of compression buckling for laser-welded stiffened plates based on physics-informed neural networks”. In: *Ocean Engineering* 327 (2025), p. 121026. URL: https://www.sciencedirect.com/science/article/pii/S0029801825007395.

[65] F. Liu et al. “Application of physics-informed machine learning methods in buckling design of axially compressed cylindrical shells”. In: *Thin-Walled Structures* 200 (2024), p. 111963. URL: https://www.sciencedirect.com/science/article/pii/S0263823124004063.

[66] F. Liu et al. “Buckling design of externally pressurized spherical shells based on physics-informed artificial neural network”. In: *Structures* 75 (2025), p. 108721. URL: https://www.sciencedirect.com/science/article/pii/S2352012425005351.

[67] E. Samaniego et al. “An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications”. In: *Computer Methods in Applied Mechanics and Engineering* 362 (2020), p. 112790. URL: https://www.sciencedirect.com/science/article/pii/S0045782519306826.

[68] E. Weinan and B. Yu. *The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*. 2017. arXiv: 1710.00211 [cs.LG]. URL: https://arxiv.org/abs/1710.00211.

[69] W. Wang and H.-T. Thai. “A physics-informed neural network framework for laminated composite plates under bending”. In: *Thin-Walled Structures* 210 (2025), p. 113014. URL: https://www.sciencedirect.com/science/article/pii/S0263823125001089.

[70] C. Rowan et al. *Solving engineering eigenvalue problems with neural networks using the Rayleigh quotient*. 2025. arXiv: 2506.04375 [math.NA]. URL: https://arxiv.org/abs/2506.04375.

[71] M. A. Nabian, R. J. Gladstone, and H. Meidani. “Efficient training of physics-informed neural networks via importance sampling”. In: *Computer-Aided Civil and Infrastructure Engineering* 36.8 (2021), pp. 962–977.

[72] K. El-Sawy and A. S. Nazmy. "Effect of aspect ratio on the elastic buckling of uniaxially loaded plates with eccentric holes". In: *Thin-Walled Structures* 2001.39 (2001), pp. 938–998. URL: http://dx.doi.org/10.1016/s0263-8231(01)00040-4.

[73] W. P. Vann. "Compressive Buckling of Perforated Plate Elements". In: *International Specialty Conference on Cold-Formed Steel Structures*. Missouri: Missouri University if Science and Technology, Aug. 1971.

[74] W.-W. Yu and C. S. Davis. "Buckling Behavior and Post-Buckling Strength of Perforated Stiffened Compression Elements". In: *International Specialty Conference on Cold-Formed Steel Structures. CFSS Proceedings of International Specialty Conference on Cold-Formed Steel Structures (1971-2018)*. Missouri: Missouri University if Science and Technology, Aug. 1971.

[75] E. Maiorana, C. Pellegrino, and C. Modena. "Linear buckling analysis of perforated plates subjected to localised symmetrical load". In: *Engineering Structures* 30.11 (2008), pp. 3151–3158. URL: https://www.sciencedirect.com/science/article/pii/S0141029608001491.

[76] M. P. Kulatunga and M. Macdonald. "Investigation of cold-formed steel structural members with perforations of different arrangements subjected to compression loading". In: *Thin-Walled Structures* 67.2013 (June 2013), pp. 78–87. URL: http://dx.doi.org/10.1016/j.tws.2013.02.014.

[77] C. Moen. "Direct Strength Design of Cold-Formed Steel Members with Perforations". PhD thesis. Dissertation (Ph. D), Vols. Johns Hopkins University, Baltimore, Maryland, 2008.

[78] M. Bendsøe and O. Sigmund. "Extensions and applications". In: *Topology Optimization*. Springer Berlin Heidelberg, 2004, pp. 71–158. URL: http://dx.doi.org/10.1007/978-3-662-05086-6_2.

[79] I. Mirtsopoulos and C. Fivet. "Structural topology exploration through policy-based generation of equilibrium representations". In: *Computer-Aided Design* 160 (2023), p. 103518. URL: https://www.sciencedirect.com/science/article/pii/S0010448523000507.

[80] V. Tomar, M. Bansal, and P. Singh. "Metaheuristic Algorithms for Optimization: A Brief Review". In: *Engineering Proceedings* 59.1 (2024), p. 238.

[81] H. A. Eschenauer, V. V. Kobelev, and A. Schumacher. "Bubble method for topology and shape optimization of structures". In: *Structural Optimization* 8 (1994), pp. 42–51. URL: http://dx.doi.org/10.1007/bf01742933.

[82] M. Y. Wang, X. Wang, and D. Guo. "A level set method for structural topology optimization". In: *Computer methods in applied mechanics and engineering* (2003), pp. 227–246. URL: http://dx.doi.org/10.1016/s0045-7825(02)00559-5.

[83] Y.M. Xie and G.P. Steven. "A simple evolutionary procedure for structural optimization". In: *Computers and Structures* 49.5 (1993), pp. 885–896. URL: https://www.sciencedirect.com/science/article/pii/004579499390035C.

[84] K. D. Tsavdaridis, J. J. Kingman, and V. V. Toropov. "Application of structural topology optimisation to perforated steel beams". In: *Computers and Structures* (2015), pp. 108–123. URL: http://dx.doi.org/10.1016/j.compstruc.2015.05.004.

[85] J. A. Norato et al. “A topological derivative method for topology optimization”. In: *Structural and Multidisciplinary Optimization* 33 (2007), pp. 375–386. URL: http://dx.doi.org/10.1007/s00158-007-0094-6.

[86] C.-M. Chan and J. K. L. Chui. “Wind-induced response and serviceability design optimization of tall steel buildings”. In: *Engineering Structures* 28.4 (2006), pp. 503–513.

[87] K. Jármai et al. “Optimal design of a welded I-section frame using four conceptually different optimization algorithms”. In: *Structural and Multidisciplinary Optimization* 25.1 (2003), pp. 54–61.

[88] C. Jiang, X. Han, and G. R. Liu. “Optimization of structures with uncertain constraints based on convex model and satisfaction degree of interval”. In: *Computer Methods in Applied Mechanics and Engineering* 196.49-52 (2007), pp. 4791–4800.

[89] M. Liu, Y. K. Wen, and S. A. Burns. “Life cycle cost oriented seismic design optimization of steel moment frame structures with risk-taking preference”. In: *Engineering Structures* 26.10 (2004), pp. 1407–1421.

[90] M. Papadrakakis, N. D. Lagaros, and V. Plevris. “Design optimization of steel structures considering uncertainties”. In: *Engineering Structures* 27.9 (2005), pp. 1408–1418.

[91] J. Bonada, M. Casafont, and O. Bove. “Cross-sectional optimization of perforated pallet rack columns against distortional and global buckling”. In: *Journal of Constructional Steel Research* 210 (Nov. 2023). URL: http://dx.doi.org/10.1016/j.jcsr.2023.108105.

[92] M. M. Pastor et al. “Optimization of cold-formed steel pallet racking cross-sections for flexural-torsional buckling with constraints on the geometry”. In: *Engineering Structures* 31 (2009), pp. 2711–2722. URL: http://dx.doi.org/10.1016/j.engstruct.2009.06.017.

[93] British Standards Institution. *Steel static storage systems — Adjustable pallet racking systems — Principles for structural design*. Standard BS EN 15512:2020. London, UK: British Standards Institution, 2020.

[94] F. Erdal, E. Dogan, and M. P. Saka. “Optimum design of cellular beams using harmony search and particle swarm optimizers”. In: *Journal of Constructional Steel Research* 67 (2011), pp. 237–247. URL: http://dx.doi.org/10.1016/j.jcsr.2010.07.014.

[95] N. D. Lagaros et al. “Optimum design of steel structures with web openings”. In: *Engineering Structures* 30 (2008), pp. 2528–2537. URL: http://dx.doi.org/10.1016/j.engstruct.2008.02.002.

[96] J. Cagan and W. J. Mitchell. “Optimally Directed Shape Generation by Shape Annealing”. In: *Environment and Planning B: Planning and Design* 20.1 (Feb. 1993), pp. 5–12. URL: http://dx.doi.org/10.1068/b200005.

[97] C. Mueller. “Computational Exploration of the Structural Design Space”. PhD thesis. Massachusetts Institute of Technology, 2014.

[98] K. Shea, J. Cagan, and S. J. Fenves. “A Shape Annealing Approach to Optimal Truss Design with Dynamic Grouping of Members”. In: *Journal of Mechanical Design* 119.3 (1997), pp. 388–394.

[99] G. M. Reddy and J. Cagan. “An Improved Shape Annealing Algorithm for Truss Topology Generation”. In: *Journal of Mechanical Design* 117.3 (1995), pp. 315–321.

[100] K. Shea and J. Cagan. "Innovative Dome Design: Applying Geodesic Patterns with Shape Annealing". In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)* 11.5 (1997), pp. 379–394.

[101] J. Davies, P. Leach, and A. Taylor. "The Design of Perforated Cold-Formed Steel Sections Subject to Axial Load and Bending". In: *Thin-Walled Structures* 1997.29 (1997), pp. 141–157. URL: http://dx.doi.org/10.1016/s0263-8231(97)00024-4.

[102] A. B. Sabir and F. Y. Chow. "Elastic Buckling of Plates Containing Eccentrically Located Circular Holes". In: *Thin-Walled Structures* 4.1986 (1986), pp. 135–149. URL: http://dx.doi.org/10.1016/0263-8231(86)90020-0.

[103] C. D. Moen and B. W. Schafer. "Experiments on cold-formed steel columns with holes". In: *Thin-Walled Structures* 2008.46 (2008), pp. 1164–1182. URL: http://dx.doi.org/10.1016/j.tws.2008.01.021.

[104] J. Ketsi. "Local And Distortional Buckling of Perforated Steel Wall Studs". PhD thesis. Helsinki University of Technology, 2000.

[105] C. Featherston, K. Holford, and M. J. Eaton. "Imperfection Sensitivity In Plates - An Assessment Using DIC". In: *Orlando, Florida, USA, June 2008*. Orlando, Florida, USA: SEM Annual Conference & Exposition on Experimental and Applied Mechanics, June 2008.

[106] M. R. Bambach and K. J. Rasmussen. "Technical Note: Experimental Techniques for Testing Unstiffened Plates in Compression and Bending". In: *Experimental Mechanics* 44.2004 (Feb. 2004). 2004) . L. Dietrich, W. Kawahara, A. Phillips, An experimental study of plastic buckling of a simply supported plate under edge thrusts, Acta Mech. 29 (1978) pp. 257–267, pp. 91–96. URL: https://link.springer.com/article/10.1007/BF02427982.

[107] L. Dietrich, W. Kawahara, and A. Phillips. "An Experimental Study of Plastic Buckling of a Simply Supported Plate Under Edge Thrusts". In: *Acta Mechanica* 29 (1978), pp. 257–267.

[108] K. Prajapat, S. Ray-Chauduri, and A. Kumar. "A study on buckling of simply supported plates with edges partially restrained for in-plane movement". In: *Springer Nature* 1972.2 (Nov. 2020). URL: https://link.springer.com/article/10.1007/s42452-020-03756-1.

[109] R. Narayanan and F. Y. Chow. "Ultimate Capacity of Uniaxially Compressed Perforated Plates". In: *Thin-Walled Structures* 1984.2 (1984), pp. 241–264. URL: http://dx.doi.org/10.1016/0263-8231(84)90021-1.

[110] M. Shariati and A. Dadrasi. "Numerical and experimental investigation of loading band on buckling of perforated rectangular steel plates". In: Res. J. Recent Sci. 1 10. Oct. 2012, pp. 63–71.

[111] U.-N. Kim, I.-H. Choe, and J. K. Paik. "Buckling and ultimate strenght of perforated plate panels subject to axiam compression: expermiental and numerical investigations with design formulations". In: *Ships Offshore Structures* 2009.4 (2009), pp. 337–361.

[112] The South African Bureau of Standards. *Structural Use of Steel : The Design of Cold-Formed Stainless Steel Structural Members*. Standard SANS 10162-4. Pretoria: The South African Bureau of Standards, 2011.

[113] C. D. Moen and B. W. Schafer. "Direct Strength Method for Design of Cold-Formed Steel Columns with Holes". In: *Journal of Structural Engineering* 137.5 (2011), pp. 559–570. URL: http://dx.doi.org/10.1061/(asce)st.1943-541x.0000310.

[114] M. Casafont et al. "Design of Steel Storage Rack Columns via the Direct Strength Method". In: *Journal of Structural Engineering* 139 (2013), pp. 669–679. URL: http://dx.doi.org/10.1061/(asce)st.1943-541x.0000620.

[115] C. D. Moen and B. W. Schafer. "Elastic buckling of cold-formed steel columns and beams with holes". In: *Engineering Structures* 31.12 (2009), pp. 2812–2824. URL: http://dx.doi.org/10.1016/j.engstruct.2009.07.007.

[116] Z. Yao and K. J. R. Rasmussen. "Perforated Cold-Formed Steel Members in Compression. I: Parametric Studies". In: *Journal of Structural Engineering* 143 (2017). URL: http://dx.doi.org/10.1061/(asce)st.1943-541x.0001635.

[117] G. Taranu, S. Iacob, and N. Taranu. "Buckling Behavior of Perforated Cold-Formed Steel Uprights: Experimental Evaluation and Comparative Assessment Using FEM, EWM, and DSM". In: *Buildings* 15.9 (May 2025), p. 1561. URL: http://dx.doi.org/10.3390/buildings15091561.

[118] European Committee for Standardization. *Eurocode 3: Design of steel structures – Part 1-3: General rules – Supplementary rules for cold-formed members and sheeting*. Standard. Brussels, Belgium, 2006.

[119] R. M. Lawson and S. J. Hicks. *Design of Composite Beams With Large Web Openings. In accordance with Eurocodes and the UK National Annexes*. SCI P355. Berkshire: SCI, 2011.

[120] Approved American National Standard. *Design, Testing, and Utilization of Industrial Steel Storage Racks*. Standard ANSI MH16.1-2023. USA: Rack Manufacturers Institute, 2023.

[121] scikit. *scikit-learn Machine Learning in Python*. scikit-learn. 2025. URL: https://scikit-learn.org/stable/ (visited on 03/20/2025).

[122] ABAQUS. *11.2 Linear Perturbation analysis*. ABAQUS. Jan. 2009. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/gsa/default.htm?startat=ch11s02.html (visited on 12/15/2023).

[123] python. *random - Generate pseudo-random numbers*. python. Dec. 27, 2023. URL: https://docs.python.org/3/library/random.html (visited on 12/27/2023).

[124] SciPy. *SciPy API > Optimisation and root finding (scipy.optimiz) > minimize*. SciPy. 2025. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html (visited on 03/20/2025).

[125] D. Solav and A. Silverstein. "DuoDIC: 3D Digital Image Correlation in MATLAB". In: *Journal of Open Source Software* 74.7 (2022).

[126] Instron. *Universal Testing Systems*. Instron. 2026. URL: https://www.instron.com/en/products/testing-systems/universal-testing-systems/ (visited on 01/12/2026).

[127] *Metallic materials — Tensile testing — Part 1: Method of test at room temperature*. International Organization for Standardization, 2019.

[128] D. Solav et al. "MultiDIC: An Open-Source Toolbox for Multi-View 3D Digital Image Correlation". In: *IEEE Access* 6 (2018), pp. 30520–30535.

[129] D. Lecompte et al. "Quality assessment of speckle patterns for digital image correlation". In: *Opt. Lasers Eng.* (2006), pp. 132–1145. URL: http://dx.doi.org/10.1016/j.optlaseng.2005.10.004.

[130] International Digital Image Correlation Society. *A good practices guide for digital image correlation*. Tech. rep. Standardization, Good Practices, and Uncertainty Quantification Committee 2018. P. Reu, The art and application of DIC all about speckles: aliasing, Experim. Techn. . Allied Vision, "Manta G-865, " Allied Vision, 2025 1-3 2025-02-24. International Digital Image Correlation Society (iDICs), 2018. URL: https://doi.org/10.32720/idics/gpg.ed1/print.format.

[131] P. Reu. "All about speckles: Speckle Size Measurement". In: *Experimental Techniques* 38.6 (2014), pp. 1–2. URL: https://doi.org/10.1111/ext.12110.

[132] Allied Vision. *Manta G-895*. Allied Vision. 2025. URL: https://www.alliedvision.com/en/camera-selector/detail/manta/g-895/ (visited on 02/24/2025).

[133] National Instruments. "What is NI measurement & automation explorer (NI MAX)". In: (2024). URL: https://www.ni.com/en/support/documentation/supplemental/21/what-is-ni-measurement---automation-explorer--ni-max--.html.

[134] National Instruments. *LabVIEW*. Austin, TX, USA, 2022.

[135] European Committee for Standardization. *Aluminium and Aluminium Alloys - Chemical composition and Form of Wrought Products - Part 3: Chemical composition and Form of Products*. Standard. Brussels, Belgium: European Committee for Standardization, 2007.

[136] The Steel Construction Institute (SCI). *Design Manual For Structural Stainless Steel*. Berkshire, UK: SCI, 2017.

[137] B. Johansson et al. *Commentary and worked examples to EN 1993 plated structural elements, JRC-ECCS (2007)*. Tech. rep. Aachen. 2007. URL: https://publications.jrc.ec.europa.eu/repository/handle/JRC38239.

[138] T. Wierzbicki. *Massachusetts institute of technology structural mechanics, 13 01*. 2024. URL: https://eng.libretexts.org/Bookshelves/Mechanical_Engineering/Structural_Mechanics_(Wierzbicki)/10:_Buckling_of_Plates_and_Sections/10.07:_Effect_of_Initial_Imperfection.

[139] J. Pan et al. "Experimental and numerical study of 3D printed steel plates stiffened by sinusoidal waves for enhanced stability". In: *Engineering Structures* 293 (Oct. 2023), p. 116577. URL: http://dx.doi.org/10.1016/j.engstruct.2023.116577.

[140] Z. Yao and K. J. Rasmussen. *Finite Element Modelling and Parametric Studies of Perforated Thin-Walled Steel Columns (No. R948)*. Report, Research. Sydney The University of Sydney School of Civil Engineering. SeS faculties schools: Faculty of Engineering: School of Civil Engineering, 2014. URL: https://hdl.handle.net/2123/24065.

[141] M. A. Crisfield. "A fast incremental/iterative solution procedure that handles "snapthrough". In: *Comput. Struct.* 13.1-3 (1981), pp. 55–62.

[142] G. Powell and J. Simons. "Improved iterative strategy for nonlinear structures". In: *International Journal for Numerical Methods in Engineering* 17.10 (1981), pp. 1455–1467. URL: http://dx.doi.org/10.1002/nme.1620171003.

[143] E. Ramm. "Strategies for Tracing the Nonlinear Response Near Limit Points". In: *Nonlinear Finite Element Analysis in Structural Mechanics*. Ed. by Wunderlich. W, Stein. E, and K.J. Bathe. Springer Berlin Heidelberg, 1981, pp. 63–89.

[144] ABAQUS. *28.3.2 Coupling constraints, 15 05*. 2024. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?startat=pt08ch28s03aus108.html#usb-cni-pcoupling.

[145] ABAQUS. *11.3.1 Introducing a geometric imperfection into a model, 27 05*. 2024. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?startat=pt04ch11s03aus55.html.

[146] D. G. Williams and A. C. Walker. "Explicit solutions for plate buckling analysis". In: *Journal of the Engineering Mechanics Division* 103.4 (1977), pp. 549–568. URL: http://dx.doi.org/10.1061/jmcea3.0002250.

[147] A. Dadrasi. "On the effect of imperfection on buckling load of perforated rectangular steel plates". In: *Res. J. Recent Sci.* 2.3 (2012), pp. 36–43.

[148] J.-H. Kim et al. "Lateral deflection behaviour of perforated steel plates: experimental and numerical approaches". In: *J.* 9.498 (Mar. 2021). Sci. Eng. URL: http://dx.doi.org/10.3390/jmse9050498.

[149] Civil Thin-walled Structures Group and Johns Hopkins University Systems Engineering. *CUFSM: A Finite Strip Method Program for Thin-Walled Structures*. https://www.ce.jhu.edu/cufsm/about/. Accessed: 2025-05-26. 1997.

[150] Dassault Systèmes Simulia Corp. *ABAQUS*. Version 2022. France, 2022.

[151] E. Mirambell and E. Real. "On the calculation of deflections in structural stainless steel beams: an experimental and numerical investigation". In: *Journal of Constructional Steel Research* 54 (2000), pp. 109–133. URL: http://dx.doi.org/10.1016/s0143-974x(99)00051-6.

[152] L. Gardner and X. Yun. "Description of stress-strain curves for cold-formed steels". In: *Construction and Building Materials* 189.189 (2018), pp. 527–538. URL: http://dx.doi.org/10.1016/j.conbuildmat.2018.08.195.

[153] European Committee for Standardization. *Eurocode 3: Design of steel structures - Part 1-4: General rules - Supplementary rules for stainless steels*. Standard EN 1993-1-4. Brussels, Belgium: European Committee for Standardization, 2006.

[154] X. Yun, Z. Wang, and L. Gardner. "Full-Range Stress-Strain Curves for Aluminium Alloys". In: *Journal of Structural Engineering* 147.6 (2021). URL: http://dx.doi.org/10.1061/(asce)st.1943-541x.0002999.

[155] European Committee for Standardization. *Eurocode 3: Design of steel structures - Part 1-14: Design assisted by finite element analysis (Draft)*. Standard prEN 1993-1-14:2022. Brussels, Belgium: European Committee for Standardization, 2022.

[156] GRAITEC. *Displacement Control Nonlinear Analysis - Technical Background*. n.d. URL: https://www.graitec.com/Help/Advance_Design/En/displacement_control_nonlinear_analysis.htm (visited on 07/26/2024).

[157] ABAQUS. *6.2.2 Static stress analysis*. 2004. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?startat=pt03ch06s02at01.html (visited on 07/19/2024).

[158] Altair Simulation. *Abaqus Interface 2022 Release Notes*. 2022. URL: https://2022.help.altair.com/2022/hwcfdsolvers/altair_help/topics/release_notes/rn_2022_interface_abaqus_r.htm (visited on 01/12/2026).

[159] J. K. Ward. *Design of composite and non-composite cellular beams*. Publication Number P100. Steel Construction Institute. Silwood Park, Ascot, Berkshire, UK, 1990. URL: https://www.steel-sci.com/.

[160] Pauli Virtanen et al. "Author Correction: SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (2020), pp. 352–352. URL: http://dx.doi.org/10.1038/s41592-020-0772-5.

[161] M. S. Smith et al. "Reducing embodied carbon with material optimization in structural engineering practice: Perceived barriers and opportunities". In: *Journal of Building Engineering* (2024).

[162] M. Lundy and A. Mees. "Convergence of an annealing algorithm". In: *Mathematical Programming* 34.1 (Jan. 1986), pp. 111–124. URL: http://dx.doi.org/10.1007/bf01582166.

[163] M. S. Ghalambaz et al. "Physics-informed neural networks (P INNs): application categories, trends and impact". In: *International Journal of Numerical Methods for Heat & Fluid Flow* 34.8 (July 2024), pp. 3131–3165.

[164] I. Thawon et al. "Physics-informed neural networks in the energy sector: Progress, trends, and future directions". In: *Energy Reports* 15 (2026), p. 109013. URL: https://www.sciencedirect.com/science/article/pii/S2352484725008893.

[165] G. Montavon, G. Orr, and K.-R. Müller, eds. *Neural Networks: Tricks of the Trade*. 2nd ed. Vol. 7700. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012.

[166] P. Liashchynskyi and P. Liashchynskyi. *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*. 2019. arXiv: 1912.06059 [cs.LG]. URL: https://arxiv.org/abs/1912.06059.

[167] J. Snoek, H. Larochelle, and R. P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Advances in Neural Information Processing Systems*. Vol. 25. 2012, pp. 2951–2959.

[168] F. Nogueira. *Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python*. https://github.com/fmfn/BayesianOptimization. GitHub repository. 2014.

[169] X. Glorot and Y. Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 249–256.

[170] A. A. Heydari, C. A. Thompson, and K. Mehmood. "SoftAdapt: Techniques for Adaptive Loss Weighting of Neural Networks with Multi-Part Loss Functions". In: *arXiv preprint arXiv:1912.12355* (2019).

[171] Z. Chen et al. "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 2018.

[172] S. Wang, Y. Teng, and P. Perdikaris. *Understanding and mitigating gradient pathologies in physics-informed neural networks*. 2020. arXiv: 2001.04536 [cs.LG]. URL: https://arxiv.org/abs/2001.04536.

[173] F. A. C. Viana et al. "Estimating model inadequacy in ordinary differential equations with physics-informed neural networks". In: *Computers & Structures* 245 (2021), p. 106458. URL: https://www.sciencedirect.com/science/article/pii/S0045794920302613.

[174] M. Islam et al. "Extraction of material properties through multi-fidelity deep learning from molecular dynamics simulation". In: *Computational Materials Science* 188 (2021), p. 110187. URL: https://www.sciencedirect.com/science/article/pii/S0927025620306789.

[175] Q. He et al. "Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport". In: *Advances in Water Resources* 141 (2020), p. 103610. URL: https://www.sciencedirect.com/science/article/pii/S0309170819311649.

[176] M. Bazmara, M. Mianroodi, and M. Silani. "Application of Physics-Informed Neural Networks for Nonlinear Buckling Analysis of Beams". In: *Acta Mechanica Sinica* 39.6 (2023), p. 422438. URL: https://doi.org/10.1007/s10409-023-22438-x.

[177] G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314.

[178] S. Cuomo et al. *Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next*. 2022. arXiv: 2201.05624 [cs.LG]. URL: https://arxiv.org/abs/2201.05624.

[179] PyTorch. *Autograd mechanics*. https://docs.pytorch.org/docs/main/notes/autograd.html. PyTorch main documentation, last updated May 07, 2026. 2026.

[180] D. P. Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[181] J. Nocedal. "Updating Quasi-Newton Matrices with Limited Storage". In: *Mathematics of Computation* 35.151 (1980), pp. 773–782.

[182] D. C. Liu and J. Nocedal. "On the Limited Memory BFGS Method for Large Scale Optimization". In: *Mathematical Programming* 45 (1989), pp. 503–528.

[183] A. Mathews et al. "Uncovering turbulent plasma dynamics via deep learning from partial observations". In: *Phys. Rev. E* 104 (2 Aug. 2021), p. 025205. URL: https://link.aps.org/doi/10.1103/PhysRevE.104.025205.

[184] C. Wu et al. "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 403 (2023), p. 115671. URL: https://www.sciencedirect.com/science/article/pii/S0045782522006260.

[185] M. E. O'Neill. "PCG : A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation". In: 2014. URL: https://api.semanticscholar.org/CorpusID:3489282.

[186] M. D. McKay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code". In: *Technometrics* 42 (2000), pp. 55–61. URL: https://api.semanticscholar.org/CorpusID:40429700.

[187] M. Stein. "Large Sample Properties of Simulations Using Latin Hypercube Sampling". In: *Technometrics* 29.2 (1987), pp. 143–151. URL: http://www.jstor.org/stable/1269769 (visited on 01/28/2026).

[188] J. H. Halton. "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals". In: *Numerische Mathematik* 2.1 (1960), pp. 84–90. URL: https://doi.org/10.1007/BF01386213.

[189] J. M. Hammersley and D. C. Handscomb. "Monte Carlo methods, methuen & co". In: *Ltd., London* 40 (1964), p. 32.

[190] I. M. Sobol. "On the distribution of points in a cube and the approximate evaluation of integrals". In: *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7.4 (1967), pp. 784–802.

[191] A. Celaya, D. Fuentes, and B. Riviere. *An Adaptive Collocation Point Strategy for Physics Informed Neural Networks via QR Discrete Empirical Interpolation Method*. arXiv:2501.07700. 2025.

[192] S. Subramanian et al. *Adaptive Self-supervision Algorithms for Physics-informed Neural Networks*. 2022. arXiv: 2207.04084 [cs.LG]. URL: https://arxiv.org/abs/2207.04084.

[193] Z. Mao and X. Meng. "Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving partial differential equations with sharp solutions". In: *Applied Mathematics and Mechanics (English Edition)* 44.7 (2023), pp. 1069–1084.

[194] S. Goraya, N. Sobh, and A. Masud. "Error estimates and physics informed augmentation of neural networks for thermally coupled incompressible Navier–Stokes equations". In: *Computational Mechanics* 72 (2023).

[195] R. Leiteritz and D. Pflüger. *How to Avoid Trivial Solutions in Physics-Informed Neural Networks*. 2021. arXiv: 2112.05620 [cs.LG]. URL: https://arxiv.org/abs/2112.05620.

[196] G. Kirsch. "Die Theorie der Elastizität und die Bedürfnisse der Festigkeitslehre". German. In: *Zeitschrift des Vereins Deutscher Ingenieure* 42 (1898), pp. 797–807.

[197] R. A. Soares Junior et al. "Effect OF Curvatures In The Buckling Analysis Of Perforated Plates Using The Boundary Element Method". In: *WIT Transactions on Engineering Sciences* (2022). URL: https://api.semanticscholar.org/CorpusID:250292059.

[198] E. Naraidoo. *Abalone for Grasshopper: Video Demonstrations*. Vimeo channel showcasing plugin demonstrations. Accessed: 2026-03-20. 2026. URL: https://vimeo.com/abalone4grasshopper.