



UNIVERSITY OF SHEFFIELD

Rationalization of trusses and yield-line patterns identified using layout optimization

by

Linwei He

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Department of Civil and Structural Engineering

July 2015

Declaration of Authorship

I, Linwei He, declare that this thesis entitled, 'Rationalization of trusses and yield-line patterns identified using layout optimization' and the work presented in it are my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed: _____

Date: _____

ABSTRACT

To help engineers to design and analyse structures, various tools exist. However, many of them are complicated and difficult for engineers to master. In industry simple, accurate, and rapid tools are potentially very useful. The development of such tools has thus been the main focus of this thesis.

One application is the design of lightweight truss structures. Although techniques have been available to identify efficient truss designs for more than half a century, these are not widely used in industry. A major problem is that the structures generated are often complex in form, so that manufacturing becomes problematic. To address this, the current research explores two rationalization techniques: (i) introducing joint lengths to control the number of joints that exist in the resulting structure; and (ii) utilising geometry optimization to adjust the locations of joints in a truss. The former involves a minor modification to the standard process such that it retains the linear nature of the original problem, while the latter solves a more challenging non-linear optimization problem that can simultaneously simplify (make less complicated) and improve (make lighter) a given truss layout. To ensure a rapid and reliable process for the latter, analytical expressions of functions and their derivatives are supplied to a general purpose non-linear optimizer and various practical issues are also considered. A number of benchmark problems are solved to show the efficacy of the two rationalization techniques.

Another application is yield-line analysis of reinforced concrete slabs. Even in the modern computer age, with many engineering analysis procedures successfully computerized, a fully automated means of undertaking a yield-line analysis has been lacking, forcing engineers in industry to use hand-calculations in order to benefit from the power of the yield-line method. This thesis is therefore concerned with the development of techniques that automate this method. By utilising the novel discontinuity layout optimization (DLO) method, the process of yield-line analysis has been truly automated at last. In addition, motivated by the outcomes of the rationalization procedure developed for trusses, research has been conducted to rationalize yield-line patterns generated via DLO. Similar to the technique used in trusses, analytical expressions of functions and their derivatives are deduced and then supplied to a non-linear optimizer, leading to a rapid and reliable computational process. To make DLO and the rationalization ready for use in industry, various slab configurations found in practice are also considered, permitting challenging slab problems to be tackled using the method. A number of examples from the literature and industry are analysed to demonstrate the efficacy of DLO and the rationalization technique.

Acknowledgements

First, I would like to thank all who have helped me undertake the research presented in this thesis. I would particularly like to thank my supervisors, Matthew Gilbert and Colin Smith, for their invaluable guidance, support and encouragement throughout the various stages of this work.

I would also like to thank members of the Computational Mechanics & Design research group. Without their friendship, my PhD life would not have been so enjoyable and colourful. Appreciations also to colleagues at LimitState Ltd; without their companionship in my final year, writing my thesis would have been dull.

I would like to thank my friends who shared happiness, sadness, boredom and excitement with me. Their support and encouragement over these years have been invaluable.

And most importantly, I am grateful to my family. In particular my parents who give me their endless love and support, allowing me to pursue my dream.

I would also like to acknowledge the financial support of the University of Sheffield, and that of LimitState Ltd in my final year.

Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 Motivation and objectives	1
1.2 Structure of the thesis	2
2 A brief review of mathematical optimization	4
2.1 Concepts used in optimization	5
2.2 Linear programming	6
2.2.1 Formulation	6
2.2.2 Simplex method	7
2.2.3 Interior point method for LP problems	8
2.2.4 Dual problem	8
2.3 Unconstrained non-linear optimization	9
2.4 Constrained non-linear optimization	9
2.4.1 Optimality conditions	10
2.4.2 Interior point method	11
2.4.3 Sequential quadratic programming and other methods	13
2.4.4 Penalty methods	14
2.4.5 Heuristic methods	14

2.5	Derivative calculation	15
2.5.1	Quasi-Newton methods	15
2.5.2	Automatic Differentiation	15
2.6	Conclusion	16
3	Literature review	17
3.1	Plasticity theory and limit analysis	17
3.1.1	Plastic vs. elastic	17
3.1.2	Idealizations of plasticity	18
3.1.3	Yield criteria and the associated flow rule	18
3.1.4	Limit analysis	19
3.1.5	Lower and upper bound theorems	19
3.2	Limit analysis of truss structures	19
3.3	Structural optimization	20
3.3.1	Methodologies in layout optimization	21
3.3.2	Methodologies for topology optimization	24
3.3.3	Geometry/shape optimization	24
3.4	Plane plasticity problems and discontinuity layout optimization	26
3.5	Limit analysis methods of reinforced concrete slabs - historical methods using hand analysis	26
3.5.1	Hillerborg's strip method	28
3.5.2	Yield-line theory	28
3.6	Computer-aided methods in limit analysis of slabs	31
3.6.1	Lower bound analysis using numerical methods	32
3.6.2	Upper bound analysis using numerical methods	32
3.7	Conclusions	34
4	Rationalization of trusses generated via layout optimization	36
4.1	Introduction	37
4.2	Rationalization of layout optimization solutions using joint lengths	38
4.3	Post-processing rationalization using geometry optimization	39
4.3.1	Basic geometry optimization formulation	39
4.3.2	Geometry optimization formulation with self-weight	43

4.3.3	Practical issues	44
4.3.4	Overall procedure	49
4.4	Numerical examples	50
4.4.1	Hemp cantilever	50
4.4.2	Centrally loaded Michell beam	53
4.4.3	Hemp arch with distributed load	54
4.4.4	Hemp cantilever with self-weight	56
4.4.5	Chan cantilever with two load cases	56
4.4.6	Flower truss with two load cases	56
4.4.7	Michell sphere	60
4.5	Conclusions	60
5	Automatic rationalization of yield-line patterns identified using discontinuity layout optimization	62
5.1	Introduction	63
5.2	Automated yield-line analysis using DLO	65
5.2.1	Overall problem formulation	65
5.2.2	Terms for a single yield-line	66
5.3	Geometry optimization: basic formulation	68
5.3.1	First derivative terms	69
5.3.2	Second derivative terms	70
5.3.3	Assembling the entire problem	70
5.4	Geometry optimization: practical issues	71
5.4.1	Modelling complex slab geometries	71
5.4.2	Inherited issues	72
5.5	Geometry optimization: full formulation	74
5.6	Numerical examples	74
5.6.1	Gilbert et al. (2014) examples	74
5.6.2	Irregular slabs with corner fans	75
5.6.3	Cruciform slab	79
5.7	Conclusions	81
5.8	Acknowledgements	81

6	Automatic yield-line analysis of practical slab problems via discontinuity layout optimization	82
6.1	Introduction	83
6.2	Discontinuity Layout Optimization (DLO) formulation	84
6.3	Modelling features of practical slabs	87
6.4	Application of the automated method	89
6.5	Conclusions	102
6.6	Acknowledgements	102
7	Discussion	104
7.1	Introduction	104
7.2	Issues in layout optimization	104
7.2.1	Use of a user-defined filter when extracting a layout	104
7.2.2	Implementation issues in DLO	105
7.2.3	The effectiveness of uniform nodal grids in DLO	106
7.3	Issues in geometry optimization	108
7.3.1	Trusses in engineering practice	109
7.3.2	Restricting nodes and line connections in non-convex domains	109
7.3.3	Non-smooth issues in geometry optimization for slabs	112
7.4	Computational aspects in layout optimization	114
7.4.1	Mathematical aspects of the member adding scheme	114
7.4.2	On potential improvements to the member adding scheme	115
7.4.3	Estimating duality gap to bracket solution	115
7.4.4	Investigating the heuristics	116
7.4.5	Parallel processing and decomposition	116
7.5	Computational aspects in geometry optimization	117
7.5.1	Steps ready for parallel processing	119
7.5.2	Taking information from previous iterations	119
7.5.3	Alternative non-linear optimization methods	119
8	Conclusions	120
9	Recommendations for future work	122

References	123
A Automatic yield-line analysis of slabs using discontinuity layout optimization	134
B The yield-line method for concrete slabs: automated at last	158
C Second derivative terms in geometry optimization for slab	164
D Domain decomposition techniques for slab	167
E Extra considerations in geometry optimization	169
F Geometric data of the real-world slab	170
G Sign convention used in DLO (for the yield-line analysis of slabs)	172
G.1 Relative displacements	172
G.2 Local coordinate system	173
G.3 Yield-lines on slab boundary	174
G.4 Conclusion	175

List of Figures

1.1	An ancient arch bridge (built in 618 A.D.)	1
2.1	Global and local optimums in a minimization problem	5
2.2	Optimality conditions of a simple example	10
3.1	Stress-strain relations for idealized (a) elastic-plastic, and (b) rigid-plastic materials	18
3.2	Limit analysis for a truss structure	20
3.3	Types of structural optimization problem	21
3.4	Steps in layout optimization	22
3.5	Moments at a point in a slab as generalized stresses	26
3.6	Square yielding criterion	27
3.7	Strip method: load to be carried on X and Y strips	29
3.8	Yield-line pattern for a simply supported rectangular slab under distributed load .	29
3.9	Kinematically admissible failure pattern required for yield-lines	30
3.10	FE mesh for a simply supported rectangular slab under distributed load	33
4.1	Four-bar truss illustrating non-convex nature of geometry optimization	41
4.2	Notation used for a truss bar	41
4.3	Node move limit zone	45
4.4	Merging a group of nodes	47
4.5	Flow chart of the ‘two phase’ geometry optimization procedure	49
4.6	Hemp cantilever	51
4.7	Hemp cantilever: influence of joint length on numerical solution and layout . . .	51
4.8	Hemp cantilever (150 \times 75 nodal divisions)	52
4.9	Hemp cantilever: geometry optimization solutions obtained during the iterative solution procedure	53

4.10	Centrally loaded Michell beam: problem definition	54
4.11	Hemp arch with distributed load	57
4.12	Hemp cantilever with self-weight	58
4.13	Chan cantilever with two load cases	58
4.14	Flower truss with two load cases	59
4.15	Flower truss with two load cases: equivalent single load case problems using superposition principle	59
4.16	Michell sphere	61
5.1	Steps in the DLO procedure	65
5.2	Relative displacements at yield-line AB	66
5.3	Notation used for a yield-line i connecting points A and B	67
5.4	Computing the effect of loads ‘above’ yield-line AB	67
5.5	Node move limit constraints introduced by a non-smooth top edge	71
5.6	Fixed square slab: comparison of DLO and rationalized yield-line patterns	77
5.7	Slab with alcoves: coarse resolution DLO solutions suitable for hand checking	77
5.8	Fixed square slab: CPU time vs. percentage error when using DLO alone and DLO with geometry optimization rationalization	78
5.9	Islam and Park’s slab	78
5.10	Kwan’s five sided slab	79
5.11	Cruciform slab: problem specification	79
6.1	Steps in DLO process: a simple example (after He & Gilbert 2015 <i>b</i>)	85
6.2	Displacement variables for a yield-line AB	85
6.3	Compatibility requirements for yield-lines meeting at a node	86
6.4	Yield moments m_{px}^+ , m_{py}^+ , m_{px}^- and m_{py}^- in an orthotropically reinforced concrete slab	87
6.5	Six-sided plate (configuration 1)	90
6.6	Six-sided plate (configuration 2)	91
6.7	Trapezoidal orthotropic slab	92
6.8	Roof slab	93
6.9	Johansen’s slab with columns	93
6.10	Point loaded slabs - DLO yield-line patterns	95

6.11	Slabs loaded with line loads - DLO yield-line patterns	96
6.12	Slab with central patch load	96
6.13	Real-world slab: a seven-storey block of flats in London	97
6.14	Real-world slab: rationalized yield-line patterns using DLO	98
6.15	Determining characteristic features of a yield-line pattern in an orthotropic slab using an affine transformation	99
6.16	Rectangular slab simply supported on two adjacent edges	100
6.17	Rectangular slab simply supported on two edges: parametric study for the yield- line pattern proposed by Kennedy & Goodchild (2004)	101
6.18	Ramsay’s L-shaped slab simply-supported on three sides	102
7.1	Ambiguous load effect by point load at point C	106
7.2	Grid influence on the size of a failure circle of a point load	107
7.3	Slab with point (live) and pressure (dead) loads acting in different directions: para- metric study by varying pressure load	107
7.4	Circular grid to address infinitely small failure circle	108
7.5	Slab with point (live) and pressure (dead) loads acting in different directions: using an extra circular grid with 65 nodes along the perimeter	108
7.6	A truss design problem studied by Smith et al. (2015)	109
7.7	Domain constraints for (a) convex domain and (b) concave domain	110
7.8	Line collides with concave domain	111
7.9	Preventing line collision with concave boundary	111
7.10	Regions divided by non-smooth point of the top edge	112
7.11	Approximating a non-smooth edge using a curved line	112
7.12	Situation when point load at C is taken on yield line AB	113
7.13	Potential parallel truss layout optimization solution strategy	118
D.1	Domain decomposition for a slab with hole, leading to three sub-domains Ω_1 , Ω_2 , and Ω_3 connected by six coupling boundaries (i.e., e_{12} , e_{21} , etc.)	167
E.1	Extra move limits imposed on nodes due to the presence of point and line loads	169
F.1	Real-world slab: geometry and column ID	170
G.1	Two blocks contacting at yield-line AB	172

G.2	Relative displacements between two contacting blocks	173
G.3	Local coordinate systems established on the two contacting blocks	173
G.4	Evaluating the load effect of a point load P on a yield-line that reads AB or BA .	173
G.5	Alternative local coordinate system for yield line that reads BA	174
G.6	Numbering order of boundary yield-lines	175

List of Tables

4.1	Hemp cantilever: solution and non-linear optimization CPU cost for varying layout optimization nodal densities	52
4.2	Hemp cantilever: solution and non-linear optimization CPU cost for varying nodal merging radii (30×15 nodal divisions)	53
4.3	Centrally loaded Michell beam	55
5.1	Gilbert et al. (2014) examples: DLO and rationalized yield-line patterns	76
5.2	Fixed square slab: influence of number of DLO nodal divisions	77
5.3	Cruciform slab: rationalized yield-line patterns for various x/L ratios	80
6.1	Evaluating the effects of loads lying in a strip ‘above’ yield-line i	86
6.2	Summary of line support types	88
6.3	Summary of column support types	89
6.4	Johansen’s slab with column supports	94
6.5	Summary of DLO solutions against those found in literature	103
7.1	Hemp cantilever in Fig. 4.6(a): comparing no. of bars in full, primal, and extracted problems	115
7.2	Hemp cantilever in Fig. 4.8(a): iterative solutions via member adding scheme	116
F.1	Real-world slab: vertices of the polygonal slab domain (units in metres)	170
F.2	Real-world slab: locations of holes (units in metres)	171
F.3	Real-world slab: information of column supports (units in metres)	171

Chapter 1

Introduction

The activity of designing structures has a long history. In ancient times people relied on trial and error to realise effective designs. This meant our ancestors 1400 years ago were able to design and build elegant arch bridges, some of which are still standing today (e.g., Fig. 1.1), all without systematic design methods or modern computing techniques.



Figure 1.1: An ancient arch bridge (built in 618 A.D.) (Wikipedia 2012)

Generally, structural design involves the use of structural analysis, required to evaluate safety and serviceability. In this research, efficient structural analysis methods were investigated by using *direct methods*, which directly assess load carrying capacity, without considering detailed structural behaviour prior to collapse. Direct methods can be embedded in the design process to obtain economical, safe and serviceable structures.

1.1 Motivation and objectives

The advent of computational techniques in the mid-twentieth century opened up a new era for structural design, freeing engineers from time-consuming hand calculations, and providing them with greater design freedom.

Methods of structural design have been studied extensively in recent decades. Some methods have been developed to help engineers accurately model the response of structures, taking into account the ‘true’ behaviour of materials, and the dynamic characteristics of real-world structures. Other methods have been developed to improve the efficiency of the design process, for example

guiding engineers towards efficient structural topologies, based on simplified representations of material and/or structural behaviour. Precision and speed are often not achieved simultaneously; engineers often rely on approximate solutions to save time. Taking advantage of modern theory and computing power, this research has focused on the development of an accurate, and most importantly, a rapid design tool for use in industry.

Sophisticated software exists that can be readily used in industry; however, some are based on complicated theories making them only suitable for expert users. In addition, to accurately model the complex behaviours of structures a large number of parameters typically need to be introduced. These create a barrier for practitioners and engineers who have just entered the industry. More importantly, incorrect program parameters can cause unexpected behaviour and erroneous solutions, leading to unsafe design and analysis. For this reason this thesis has focused on simple theories and produced software that utilises few parameters. Some of the solutions presented can even be validated through hand analysis.

Many real-world structures include truss elements (i.e., elements that primarily carry either tension or compression only). Trusses can be used to form large scale engineering structures, such as canopies and long-span roofs. They can be highly efficient forms with good strength-to-weight characteristics; for this reason, trusses can also be used in some light-weight mechanical components, for example racing cars and aeroplanes. In recent years, new technology in additive manufacturing permits relatively complex structural components to be made, and a close cooperation has been established at the University of Sheffield between the structural design and additive manufacturing groups. Given this opportunity, a key focus of this thesis was on the design of trusses, which involved the process of finding optimal structures that are highly efficient and light-weight. Although engineering methods and tools are available for designing such structures, one issue has been noted: while solutions can be optimal in theory, their layouts are very often too complex to be accurately fabricated even with modern manufacturing technology. Therefore, a method for generating more rational designs is of particular interest.

On the other hand, it is of interest to note that an analogy exists between an optimum truss layout and the layout of yield-lines in a reinforced concrete slab at failure. Therefore, methods developed for truss design problems can potentially also be used in slab analysis. Given the fact that engineers still perform hand calculations to analyse slabs, this research sought to develop engineering software for automating these analyses.

1.2 Structure of the thesis

This thesis contains nine core chapters and additional sections in the Appendix. This chapter introduces the background, motivation, and objectives of the current research, and outlines the structure of this thesis.

- Chapter 2 primarily reviews mathematical optimization methods involved in the research, including linear programming (LP) and non-linear programming (NLP).
- Chapter 3 provides an overview of theories used in this thesis. A brief review will be given

of plastic limit analysis, followed by one application - optimal truss design using layout optimization. Then the limit analysis of reinforced concrete slabs is introduced, noting the limitation of existing methods and thus the need for a new method.

- Chapter 4 comprises an academic paper which introduces two techniques that can be used for rationalization, developed to address the complexity found in truss structures generated using layout optimization. Mathematical formulations, implementation issues and numerical examples are provided.
- A new method of performing slab yield-line analysis using discontinuity layout optimization (DLO) is introduced in Appendix A; Chapter 5 comprises a further academic paper which describes how rationalization techniques can be transferred to slab analysis. Similarly, mathematical formulae and implementation issues are provided and numerical examples are solved.
- Chapter 6 comprises a third academic paper which focuses on more practical aspects of slab yield-line analysis. Modifications to the original DLO formulation are made to enable the analysis of more complicated slabs (e.g., slabs with orthotropic reinforcement, knife-edge supports, etc). Case studies containing various numerical examples taken from the literature and engineering practice are provided to further demonstrate the efficacy of DLO.
- Chapter 7 discusses various issues relevant to the current research.
- Chapter 8 presents the conclusions of the thesis.
- Chapter 9 presents details of proposed future work.

Chapter 2

A brief review of mathematical optimization

This thesis is primarily concerned with applying numerical approaches to engineering problems. Mathematical optimization is of significant interest because it provides effective numerical methods that can facilitate rapid engineering processes. There are a wide variety types of different optimization problems. Nevertheless, a single objective optimization problem can be written in general as:

$$\min_x f(\mathbf{x}) \quad (2.1a)$$

$$\text{s.t. } h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, l \quad (2.1b)$$

$$g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, m, \quad (2.1c)$$

where \mathbf{x} is a vector of unknown variables typically named the optimization variable; $f(\mathbf{x})$ is the objective function to be minimized (or maximized if given \max_x). Meanwhile, \mathbf{x} must satisfy equality constraint $h_i(\mathbf{x})$ in (2.1b) and inequality constraint $g_j(\mathbf{x})$ in (2.1c). In general, variable \mathbf{x} can be discrete, e.g., admit only integer values; however, in this thesis variables are continuous ($\mathbf{x} \in \mathbb{R}^n$), which normally yields relatively efficient mathematical approaches. According to the mathematical properties of the objective function $f(\mathbf{x})$, constraints $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$, the optimization problem (2.1) can be categorized as:

Linear programming problem When the objective function (2.1a), equality constraint (2.1b) and inequality constraint (2.1c) are linear with respect to the optimization variable \mathbf{x} , the optimization problem is called a linear programming (LP) problem.

Non-linear programming problem When any of the functions in (2.1a), (2.1b) and (2.1c) are non-linear with respect to optimization variable \mathbf{x} , the optimization problem is a non-linear programming (NLP) problem. This category can be further subdivided into two subcategories: constrained and unconstrained NLP problems, depending on whether constraints are included.

The current research involves both LP and NLP problems. Many of the theoretical aspects are used

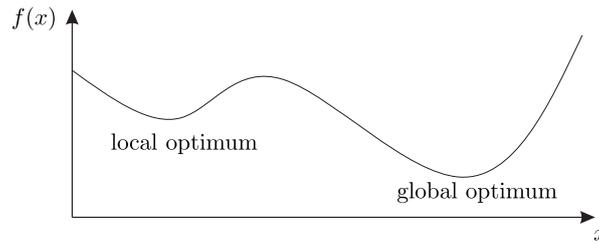


Figure 2.1: Global and local optimums in a minimization problem

to aid the development of solution strategies for engineering problems. It is therefore important to introduce some key theories in mathematical optimization.

2.1 Concepts used in optimization

Before describing mathematical optimization theories, it is necessary to introduce some commonly used concepts.

Minimization and maximization The concepts of *minimization* and *maximization* are straightforward: they directly describe the objectives of the concerned problems. A maximization problem can normally be converted to an equivalent minimization problem, e.g., a simple conversion can be made using: $\max_x f(\mathbf{x}) \Rightarrow \min_x -f(\mathbf{x})$. In the following content, only a minimization problem (i.e., (2.1)) is described; a point \mathbf{x}^* at which the objective function $f(\mathbf{x})$ is minimized is often called a *minimizer*.

Global and local optimums A *global optimum* is strictly the minimum value the objective function (2.1a) can have in the entire variable space, while a local optimum means that the calculated objective function is the minimum value in a neighbourhood of the minimizer (Fig. 2.1). In many engineering problems, a global optimum cannot be assured, so a locally optimized solution is normally acceptable.

Constrained and unconstrained If either or both constraints (2.1b) and (2.1c) are included, the optimization problem is *constrained*, otherwise it is *unconstrained*. The former case is considered in this work. In this category, a point \mathbf{x} satisfying all constraints is called a *feasible point*, the set containing all feasible points is called *feasible set*, denoted as \mathbb{X} , and from a geometric point of view, is also called the *feasible region*.

Convexity Convexity is significant because it affects many aspects of optimization problems. It involves the feasible set \mathbb{X} as well as the objective function $f(\mathbf{x})$. Now consider *any* two points $\mathbf{x}_1 \in \mathbb{X}$, $\mathbf{x}_2 \in \mathbb{X}$ and an arbitrary number $\alpha \in [0, 1]$. Set \mathbb{X} is convex if the following condition is met:

$$\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2 \in \mathbb{X}, \text{ for all } \alpha \in [0, 1]. \quad (2.2)$$

The objective function $f(\mathbf{x})$ is convex if the following condition is satisfied:

$$f(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2), \text{ for all } \alpha \in [0, 1]. \quad (2.3)$$

An important statement regarding convexity is that, when both $f(\mathbf{x})$ and \mathbb{X} are convex, any local optimum of (2.1) is a global optimum.

Smoothness Many mathematical optimization methods examine the derivatives of the objective function (2.4a) and constraints (2.1b) and (2.1c) with respect to the optimization variable \mathbf{x} . Accordingly, it is required that those functions are first- or second-order differentiable at the minimizer. It is worth noting that: (i) a problem with a non-smooth objective function can sometimes be transformed to a smooth problem by reformulating the objective function and constraints, and (ii) a non-smooth constraint can normally be rewritten as several separate smooth constraints.

Within the scope of this thesis, the smoothness condition is always satisfied in the objective function; potential non-smooth constraints are presented and steps taken to ensure this condition is satisfied.

Duality Duality theory (e.g., Vanderbei 2001, Nocedal et al. 2006) shows how an alternative problem can be formulated, by using the functions and coefficients from the original optimization problem (2.1), to be highly relevant to the original problem. The new problem is called the *dual* of the *primal*, i.e., the original problem. While the primal problem (2.1) is stated as a minimization problem, its dual is a maximization problem. The dual problem gives the lower bound solution of the primal and vice versa; the gap between them is called the *duality gap*. In certain cases, the duality gap is zero, so simultaneously investigating both the primal and dual problems can lead to very efficient optimization methods.

2.2 Linear programming

2.2.1 Formulation

Assume linear functions in (2.1a), (2.1b) and (2.1c), problem (2.1) can now be written in matrix notation as:

$$\min_x \quad \mathbf{c}^T \mathbf{x} \quad (2.4a)$$

$$\text{s.t.} \quad \mathbf{A}_h \mathbf{x} = \mathbf{b}_h \quad (2.4b)$$

$$\mathbf{A}_g \mathbf{x} \geq \mathbf{b}_g, \quad (2.4c)$$

where \mathbf{c} is a coefficient vector of the objective function; \mathbf{x} is the optimization variable; \mathbf{A}_h and \mathbf{A}_g are the equality and inequality matrices, respectively, and \mathbf{b}_h and \mathbf{b}_g are constant vectors.

Though problem (2.4) can be considered a special case of (2.1), LP methods have developed in a different way from the NLP methods. LP and NLP problems were considered completely separate problems for decades, and the methods used for solving them varied significantly.

An important mathematical aspect of a LP problem is that the objective function (2.4a) and the feasible set (defined by constraints (2.4b) and (2.4c)) satisfy conditions (2.3) and (2.2), respectively; this implies that LP problems are convex, so any local optimum is also a global optimum.

Instead of investigating the general form (2.4), a so-called *standard form* is considered by many (e.g., Vanderbei 2001). The standard form excludes the inequality constraint (2.4c); in addition, it contains only non-negative optimization variables. It can be transformed from the general form by eliminating inequality constraint (2.4c) using non-negative slack variables \mathbf{s} ($\mathbf{s} \geq \mathbf{0}$), replacing (2.4c) with $\mathbf{A}_g \mathbf{x} - \mathbf{s} = \mathbf{b}_g$. Furthermore, the optimization variable \mathbf{x} in (2.4) is replaced by non-negative variables \mathbf{x}^+ and \mathbf{x}^- (also imposing $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$). The resulting standard form can be written:

$$\min_{\tilde{\mathbf{x}}} \quad \tilde{\mathbf{c}}^T \tilde{\mathbf{x}} \quad (2.5a)$$

$$\text{s.t.} \quad \tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad (2.5b)$$

$$\tilde{\mathbf{x}} \geq \mathbf{0}, \quad (2.5c)$$

where $\tilde{\mathbf{x}} = [\mathbf{x}^{+T}, \mathbf{x}^{-T}, \mathbf{s}^T]^T$ contains the new optimization variables, and $\tilde{\mathbf{c}}$, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ are constructed using the coefficients of (2.4). There exists two major methods for solving LP problems: the simplex method and the interior point method ^[i].

2.2.2 Simplex method

Invented in the late 1940s by Dantzig (1949), the *simplex method* has been continuously improved since its conception. It currently has many variations such as the primal and the primal-dual methods, and various software packages (e.g., GUROBI 2014) have been made available. Nonetheless, the main features of the original simplex method remain, and can be explained using mathematical and geometric interpretations.

In a mathematical interpretation, assume matrix $\tilde{\mathbf{A}}$ has full row rank. The column space of $\tilde{\mathbf{A}}$ is significant. According to theories in linear algebra, a basis of the column space is formed by a set of columns of $\tilde{\mathbf{A}}$; $\tilde{\mathbf{b}}$ is a linear combination of columns taken from this set and the corresponding coefficients (i.e., values in $\tilde{\mathbf{x}}$ corresponding to these columns) can be calculated. Conversely, the coefficients corresponding to other non-basis columns are zero. Given this column set, the objective function can be calculated. Typically, this set is not unique, so the resulting objective value varies. The simplex method first identifies one set, and then in each iteration creates a new set by replacing one column with a non-basis column, aiming to reduce the objective value.

^[i]An interchangeable name is the ‘barrier method’. In this thesis, the term ‘interior point method’ refers to the modern barrier methods developed since the mid 1980s, i.e., the ‘primal-dual interior point methods’. On the other hand, the early ‘barrier methods’, i.e., ‘primal interior point methods’ developed in 1955 (Frisch 1955), are not considered.

In a geometric interpretation, the objective function $\tilde{\mathbf{c}}^T \tilde{\mathbf{x}}$ defines a hyperplane; the objective value increases along its normal direction, and vice versa. The feasible set $\{\tilde{\mathbf{x}} \in \mathbb{R}^n \mid \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tilde{\mathbf{x}} \geq \mathbf{0}\}$ in (2.5) defines a polytope. If the optimization problem is feasible (i.e., feasible set \mathbb{X} is not empty) and bounded, a minimizer can always be found at a vertex of the polytope. The simplex method seeks a minimizer by browsing the vertices of the polytope, starting from one vertex and moving to an adjacent one. Since a polytope is convex, a non-ascending (i.e., not increasing objective function) ‘movement’ is guaranteed unless the minimizer is found.

By browsing vertices of the polytope, every point in its search path is always feasible, and a minimizer can certainly be found. However, in the worst scenario the simplex method can visit every vertex, leading to an inefficient process. For large-scale problems, the polytope has a considerable number of vertices; for this reason, the simplex method is potentially inefficient.

2.2.3 Interior point method for LP problems

Research on solving LP problems using the interior point method dates back to the mid 1980s (Karmarkar 1984, Gill et al. 1986). Realising the issues in using the simplex method for solving large-scale problems, researchers began to seek more efficient approaches. In Wright (2004), an overview of the developments of the primal-dual interior point method was given, showing a profound influence brought by the method, which soon became a strong competitor to the then state-of-the-art simplex method, particularly for solving large-scale problems. Software packages have also been developed, e.g., MOSEK (2011). It is also worth noting that the interior point method, perhaps for the first time, bridged the methodologies developed for LP and NLP problems; it is now widely recognized as a general means of tackling both LP and NLP problems. Nonetheless, highly efficient solution strategies have been developed for LP problems by utilising their linear nature. Key theoretical aspects of the interior point method will be reviewed with NLP methodologies in Section 2.4.2.

2.2.4 Dual problem

According to duality theory, the dual problem of (2.5) can be derived as (e.g., see Vanderbei 2001):

$$\max_{\tilde{\boldsymbol{\lambda}}} \quad \tilde{\mathbf{b}}^T \tilde{\boldsymbol{\lambda}} \tag{2.6a}$$

$$\text{s.t.} \quad \tilde{\mathbf{A}}^T \tilde{\boldsymbol{\lambda}} \leq \tilde{\mathbf{c}}. \tag{2.6b}$$

Problem (2.6) is a maximization problem, thus giving the lower bound of primal problem (2.5). The dual problem (2.6) is of particular interest, due to the so-called strong duality (e.g., Vanderbei 2001, Nocedal et al. 2006, Boyd & Vandenberghe 2004): *‘If either (2.5) or (2.6) has a finite optimal solution, so does the other, and the corresponding optimum values of the objective functions are equal. If either problem has an unbounded objective, the other problem has no feasible solution.’* Therefore, the duality gap between the two problems is zero.

2.3 Unconstrained non-linear optimization

As mentioned, before the advent of interior point method, there was almost no connection between LP and NLP methods. Most NLP methods involve investigating the so-called *optimality conditions* that are satisfied at the minimizer.

Assume a non-linear objective function (2.1a), conjecture a local optimum point \mathbf{x}^* and approximate (2.1a) using a Taylor series (to second-order) at this point: $f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \nabla f^T(\mathbf{x}^*)\mathbf{d} + \frac{1}{2}\mathbf{d}^T \nabla_{xx}^2 f(\mathbf{x}^*)\mathbf{d}$, where $\mathbf{d} \in \mathbb{R}^n$ is any increment of \mathbf{x} in a neighbourhood of \mathbf{x}^* , and $\nabla_{xx}^2 f(\mathbf{x}^*)$ is the Hessian matrix of $f(\mathbf{x}^*)$. To ensure \mathbf{x}^* is a local minimizer, the following optimality conditions can be derived:

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad \text{and} \quad \nabla_{xx}^2 f(\mathbf{x}^*) \succ 0, \quad (2.7)$$

where the sign ‘ \succ ’ means positive definite. The first condition leads to a stationary point, while the second condition examines the local curvature of (2.1a) to ensure the identified point is a minimizer rather than a maximizer or a saddle point.

Taking (2.7), various numerical methods were developed. Given any starting point \mathbf{x}_k , a common strategy is to iteratively move \mathbf{x}_k towards a minimizer. This movement normally requires two sub-problems to be solved: (i) calculate a so-called *descent direction* \mathbf{d}_k - moving \mathbf{x}_k in this direction potentially reduces the objective value and, (ii) determine a step-length α_k so \mathbf{x}_k will be moved to $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.

Methods have been developed governing how \mathbf{d}_k and α_k are obtained. Perhaps the simplest method is the *steepest descent method*, which calculates the descent direction using $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, and the step-length α_k by performing a line search: $\min_{\alpha_k} f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$ [ii]. This method does not require the second derivative (i.e., curvature) of the objective function; for this reason, it has a so-called linearly convergent rate towards a minimizer that is deemed a relatively slow speed. Some methods take into account second derivatives, leading to a super-linear rate-of-convergence (e.g., Newton and Quasi-Newton methods).

2.4 Constrained non-linear optimization

NLP problems are typically more challenging when constrained. There are many strategies for solving NLP problems, and some are introduced here: (i) investigating the optimality conditions, (ii) transforming a constrained problem to an unconstrained problem via a parametrized objective function, which incorporates constraints, and (iii) using heuristics so mathematical derivations are not required.

[ii] Note that the optimization variable in the line search is α_k , if the current point \mathbf{x}_k and search direction \mathbf{d}_k are known.

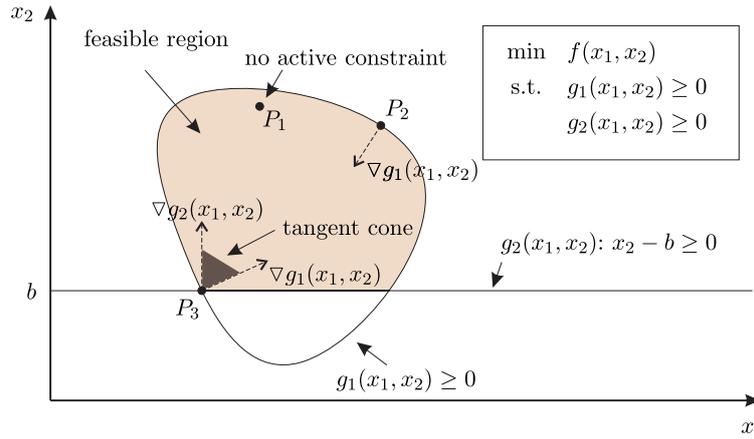


Figure 2.2: Optimalty conditions of a simple example: at P_1 , no constraint is active, a minimizer requires $\nabla f(x_1, x_2) = \mathbf{0}$; if P_2 is a minimizer, then either $\nabla f(x_1, x_2) = \mathbf{0}$ or $\nabla f(x_1, x_2) = \nabla g_1(x_1, x_2)$, while $g_2(x_1, x_2)$ is inactive; if P_3 is a minimizer, $\nabla f(x_1, x_2)$ must lie in the tangent cone.

2.4.1 Optimality conditions

Due to the presence of constraints, $\nabla f(\mathbf{x}^*) = \mathbf{0}$ is now no longer a necessary condition - to investigate the optimality conditions, the objective function (2.1a) needs to be considered in conjunction with constraints (2.1b) and (2.1c). At a local minimizer \mathbf{x}^* , the objective function (2.1a) is approximated using the first-order Taylor series: $f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \nabla f^T(\mathbf{x}^*)\mathbf{d}$, where \mathbf{d} is now a direction that potentially moves \mathbf{x}^* within the feasible region. Therefore, a local optimum satisfies:

$$\nabla f^T(\mathbf{x}^*)\mathbf{d} \geq 0. \quad (2.8)$$

It can be observed that when $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$, \mathbf{d} must be affected by constraints at point \mathbf{x}^* ; the neighbourhood of \mathbf{x}^* is therefore of particular interest. From a geometric point of view, any ‘descent direction’ must move \mathbf{x}^* out of the feasible region (because \mathbf{x}^* is a local minimizer), implying that the direction of $\nabla f(\mathbf{x}^*)$ must lie in a so-called tangent cone (origin at \mathbf{x}^*) formed using constraints. Since \mathbf{x}^* must lie on the surface defined by equality constraints, the cone must involve (2.1b); however, for inequality constraints (2.1c), there are two scenarios. If $g(\mathbf{x}^*) = 0$ is satisfied, this constraint turns out to be an equality constraint that affects the tangent cone, and is called an *active constraint*. Otherwise, there exists a finite step-length that \mathbf{x}^* can be moved in any direction without violating $g(\mathbf{x}) > 0$, which means the constraint has no contribution to the cone. This constraint is called an *inactive constraint* (also refer to P_1, P_2 in Fig. 2.2). The tangent cone can be viewed as a set of non-descending directions (also refer to P_3 in Fig. 2.2), constructed using directions taken from gradients of equality constraints: $\nabla h_i(\mathbf{x}^*)$, $i = 1, 2, \dots, l$, and those of active inequality constraints: $\nabla g_j(\mathbf{x}^*)$, for all $j \in \{g_j(\mathbf{x}^*) = 0 | j = 1, 2, \dots, m\}$. As mentioned, this set must include the direction of $\nabla f(\mathbf{x}^*)$ at a minimizer \mathbf{x}^* , so $\nabla f(\mathbf{x}^*)$ can be expressed as a linear combination of the gradient vectors above. To describe these conditions mathematically, the so-called *Lagrangian function* is used:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) - \sum_{i=1}^l \lambda_i h_i(\mathbf{x}) - \sum_{j=1}^m \mu_j g_j(\mathbf{x}), \quad (2.9)$$

where $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is the Lagrangian function, and $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are vectors of Lagrange multipliers for equality (2.1b) and inequality (2.1c) constraints, respectively. To ensure that $\nabla f(\mathbf{x}^*)$ lies in the tangent cone, the following condition needs to be satisfied:

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla f(\mathbf{x}^*) - \sum_{i=1}^l \lambda_i^* \nabla h_i(\mathbf{x}^*) - \sum_{j=1}^m \mu_j^* \nabla g_j(\mathbf{x}^*) = 0. \quad (2.10)$$

Given the fact that inequality constraints have two ‘sides’, the sign of their gradients matters. For this reason μ_j^* , ($j = 1, 2, \dots, m$), must be non-negative.

Furthermore, to determine the inequality constraints that are active, the so-called *complementary condition* is required: $\mu_j^* g_j(\mathbf{x}^*) = 0$, ($j = 1, 2, \dots, m$). It can be observed that, given $g_j(\mathbf{x}^*) > 0$, its corresponding Lagrange multiplier μ_j^* must be zero, so that its gradient will not be taken in (2.10).

By collecting the above conditions, the so-called Karush-Kuhn-Tucker (KKT) conditions at a local minimizer \mathbf{x}^* can be written (e.g., see Nocedal et al. 2006):

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla f(\mathbf{x}^*) - \sum_{i=1}^l \lambda_i^* \nabla h_i(\mathbf{x}^*) - \sum_{j=1}^m \mu_j^* \nabla g_j(\mathbf{x}^*) = 0, \quad (2.11a)$$

$$h_i(\mathbf{x}^*) = 0, \quad i = 1, 2, \dots, l, \quad (2.11b)$$

$$g_j(\mathbf{x}^*) \geq 0, \quad j = 1, 2, \dots, m, \quad (2.11c)$$

$$\mu_j^* \geq 0, \quad j = 1, 2, \dots, m, \quad (2.11d)$$

$$\mu_j^* g_j(\mathbf{x}^*) = 0, \quad j = 1, 2, \dots, m. \quad (2.11e)$$

The KKT conditions are first-order necessary conditions, and points satisfying KKT conditions are called KKT points. At a KKT point, any feasible direction \mathbf{d} is not a descent direction. However, when $\nabla f^T(\mathbf{x})\mathbf{d} = 0$, the KKT conditions cannot distinguish between a minimizer and a maximizer. In this scenario the second-order conditions need to be investigated, taking into account the curvature of the objective function.

It is worth noting that, due to the complementary condition (2.11e), the use of the gradient of the Lagrange function can be problematic: given a non-optimum point, it is typically difficult to predict which inequality constraints will be active at the potential minimizer. For this reason it becomes a major challenge in developing optimization methods.

2.4.2 Interior point method

The *barrier method* was invented in 1955, and is now viewed as a subclass of the *interior point method*: the ‘primal interior point method’. It led to ill conditioned problems, and as a consequence was almost forgotten later (e.g., Frisch 1955, Fiacco & McCormick 1968). The modern interior point method often refers to the ‘primal-dual interior point method’ that was pioneered in the mid 1980s by Karmarkar (1984) and Gill et al. (1986), when researchers were seeking efficient methods for solving LP problems. In the 1990s, this method received extensive research interest

and was successfully utilised to solve NLP problems. Instead of directly attacking problem (2.1), it first considers a so-called barrier problem with the following formulation:

$$\min_{\mathbf{x}, \mathbf{s}} \quad f(\mathbf{x}) - u \sum_{j=1}^m \ln s_j \quad (2.12a)$$

$$\text{s.t.} \quad h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, l \quad (2.12b)$$

$$g_j(\mathbf{x}) - s_j = 0, \quad j = 1, 2, \dots, m, \quad (2.12c)$$

where u , ($u \geq 0$), is a barrier parameter, and s_j ($j = 1, 2, \dots, m$) are slack variables. The use of the logarithmic function $\ln s_j$ requires the slack variables to be positive. It is worth noting that (2.12) has no inequality constraints, overcoming the difficulty of determining active and inactive constraints. However, it is an approximation of (2.1), and the barrier parameter determines the level of approximation: when u reaches zero, problems (2.12) and (2.1) become equivalent. A typical solution strategy is to use a sufficiently (but not significantly) large u and solve (2.12), and then reduce u and repeat the process until u becomes sufficiently small such that (2.12) is deemed equivalent to (2.1).

Problem (2.12) is tackled by using KKT conditions in (2.11). Define $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ as the vectors containing equality and inequality constraint functions, respectively; \mathbf{s} a vector comprising slack variables s_j ($j = 1, 2, \dots, m$). The KKT conditions of (2.12) can be written as:

$$\nabla f(\mathbf{x}) - \mathbf{J}_h^T(\mathbf{x})\boldsymbol{\lambda} - \mathbf{J}_g^T(\mathbf{x})\boldsymbol{\mu} = \mathbf{0}, \quad (2.13a)$$

$$\mathbf{S}\boldsymbol{\mu} - u\mathbf{e} = \mathbf{0}, \quad (2.13b)$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad (2.13c)$$

$$\mathbf{g}(\mathbf{x}) - \mathbf{s} = \mathbf{0}, \quad (2.13d)$$

where $\mathbf{J}_h(\mathbf{x})$ and $\mathbf{J}_g(\mathbf{x})$ are the Jacobian matrices of $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ respectively, \mathbf{S} and \mathbf{Z} are diagonal matrices whose diagonal entries are given by \mathbf{s} and $\boldsymbol{\mu}$, respectively, and \mathbf{e} is an identical vector. Equations in (2.13) form a non-linear system with respect to variables \mathbf{x} , \mathbf{s} , $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. It can be solved using Newton's method (e.g., Sauer 2011, p.131), which uses the following iterative format:

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) & 0 & -\mathbf{J}_h^T(\mathbf{x}) & -\mathbf{J}_g^T(\mathbf{x}) \\ 0 & \mathbf{Z} & 0 & \mathbf{S} \\ \mathbf{J}_h^T(\mathbf{x}) & 0 & 0 & 0 \\ \mathbf{J}_g^T(\mathbf{x}) & \mathbf{I} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \mathbf{s} \\ \delta \boldsymbol{\lambda} \\ \delta \boldsymbol{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}) - \mathbf{J}_h^T(\mathbf{x})\boldsymbol{\lambda} - \mathbf{J}_g^T(\mathbf{x})\boldsymbol{\mu} \\ \mathbf{S}\boldsymbol{\mu} - u\mathbf{e} \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) - \mathbf{s} \end{bmatrix}, \quad (2.14)$$

where $\nabla_{xx}^2 \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is the Hessian matrix of the Lagrangian function (2.9) with respect to \mathbf{x} , and $\delta \mathbf{x}$, $\delta \mathbf{s}$, $\delta \boldsymbol{\lambda}$, and $\delta \boldsymbol{\mu}$ are the so-called Newton steps for the variables \mathbf{x} , \mathbf{s} , $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$, respectively. With respect to Newton steps, equation (2.14) is *linear* and it can readily be solved^[iii]; its solution indicates a search direction leading to a KKT point of (2.12). Similar to the methods for

^[iii]Note that (2.14) may not always have a solution, preconditioning is often required.

unconstrained problems, after obtaining a search direction, a step-length α is calculated, moving the current point towards a KKT point. There exist various methods calculating this step-length; for example, a simple approach is to use the maximum step-length satisfying $s > 0$.

Given the fact that advanced techniques in linear algebra have been available, (2.14) can be solved efficiently; hence, the interior point method leads to a rapid approach to solving (2.1), particularly for problems having a large number of variables. Since this method was invented, extensive research interest has resulted in various modifications. For example, there are concerns about how the barrier parameter u is updated; techniques also have been developed to incorporate infeasible starting points, of particular interest when solving non-convex problems. Certain heuristic schemes have also been developed to increase efficiency (e.g., Wächter & Biegler 2006).

In this thesis, NLP problems are solved using the interior point method. From (2.13) it can be seen that the method requires first derivatives of objective function $f(\mathbf{x})$ and constraints $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ to be provided. In addition, (2.14) requires their second derivatives to be supplied.

2.4.3 Sequential quadratic programming and other methods

Before the modern interior point method, the most advanced approach to solving general constrained NLP problems was to use sequential quadratic programming (SQP), which involved solving a sequence of quadratic programming (QP) problems.

The QP problem is a sub-class of NLP problems; it has a quadratic objective function: $\min_x f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{G}\mathbf{x} + \mathbf{c}^T \mathbf{x}$, where \mathbf{G} is a symmetric $n \times n$ matrix and \mathbf{c} is a coefficient vector in \mathbb{R}^n ; the constraints are linear. QP problems can be solved efficiently under these specific conditions. Given a positive semi-definite or definite coefficient matrix \mathbf{G} , the objective function satisfies convexity condition (2.3), so that the QP problem becomes convex (note that linear constraints satisfy (2.2)) and a global optimum is guaranteed. Under this condition it has been discovered (e.g., Nocedal et al. 2006) that the difficulty of solving QP problems is similar to that for LP problems - this permits highly efficient numerical methods to be developed.

In SQP, a typical solution strategy involves an inner loop and an outer loop: the inner loop solves a QP sub-problem constructed by approximating the objective function (2.1a) at the current point using the second-order Taylor series, and constraints using the first-order terms; after each successful solution in the inner loop, the outer loop moves the current point forward unless a KKT point of (2.1) is found.

It is widely accepted that SQP and the interior point method are the two most efficient methods for solving general NLP problems today. Similar to the interior point method, SQP tackles (2.1) using approximated problems that can be easily solved. Methods other than SQP exist which follow a similar solution strategy. For example, sequential linear programming (SLP) approximates the objective function using first-order terms. However, this approximation ignores the curvature of the objective function, becoming less effective for solving highly non-linear problems. Another example is the method of moving asymptotes (MMA) invented by Svanberg (1987). In MMA the sub-problem is approximated using asymptotic functions constructed by supplying the value of the objective function (2.1), and the first derivatives of the objective and constraint functions

at the current point. The asymptotic functions are infinitely differentiable such that the curvature of the original function can be approximated. The level of approximation is determined by some prescribed parameters that specify the lower and upper bounds of variable \mathbf{x} . Unfortunately, the choice of these parameters can have a significant influence, so global convergence ^[iv] cannot be guaranteed in general. For this reason, further developments have been made in Svanberg (1995) and Svanberg (2002), addressing the issue of global convergence, though with higher computational cost. The main advantage of MMA is that calculating second derivatives of certain functions can be computationally expensive for some engineering problems. A means of approximating curvatures of those particular functions often leads to potentially efficient approaches, and numerous case studies have demonstrated its efficacy (e.g., Bendsøe & Sigmund 2003).

2.4.4 Penalty methods

Instead of investigating KKT conditions, an alternative approach is to eliminate constraints by constructing a parametrized objective function where constraints can be incorporated. This leads to an unconstrained problem that can readily be solved. One way to construct this parametrized objective function is by using penalized terms. Violated constraints add penalty values to the objective function, enforcing the optimization to reduce these violations. Regarding problem (2.1), a commonly used quadratic penalty function gives:

$$f^P(\mathbf{x}) = f(\mathbf{x}) + \frac{u^P}{2} \sum_{i=1}^l h_i^2(\mathbf{x}) + \frac{u^P}{2} \sum_{j=1}^m \max\{-g_j(\mathbf{x}), 0\}^2, \quad (2.15)$$

where $f^P(\mathbf{x})$ is the penalized objective function and u^P ($u^P > 0$) is a prescribed penalty parameter. Given a significantly large penalty parameter u^P , an unconstrained optimization process will also minimize the penalized terms, so constraints in the original problem are nearly satisfied (with negligible violation). When $u^P \rightarrow \infty$ problems (2.15) and (2.1) are equivalent. However, the use of large u^P value can lead to an ill-conditioned problem, potentially rendering the numerical method ineffective.

2.4.5 Heuristic methods

A major drawback of conventional gradient-based mathematical methods is that, given a non-convex problem, a point cannot normally ‘jump’ from one local minimizer to another; hence the global optimum is not generally guaranteed. For this reason research has focused on investigating new methods that can potentially visit every point in the feasible region. Perhaps more importantly, obtaining analytical expressions of derivatives of the objective function and constraints can be cumbersome and/or extremely difficult in engineering practice. Given this fact, a method for finding the optimum solution is to browse every feasible point. Depending on the number of variables, the feasible region can become enormously large such that browsing every feasible point becomes impractical. To make this process more efficient, researchers invented ‘intelligent’ algorithms, which were often inspired by observing nature. For example, genetic algorithms (GA) were

^[iv]Convergent from any starting point.

invented by adopting Darwin's 'natural selection' principle, where finding an optimum is deemed an evolutionary process. Instead of taking descent directions using derivatives of functions, the mechanism of natural evolution controls how an optimum solution is generated. It only takes objective values of the optimization problem, permitting complicated problems to be tackled without investigating the mathematical properties of the supplied data and functions. However, simulating nature also requires the objective function to be evaluated frequently, leading to a potentially inefficient process. Furthermore, certain mechanisms in GA involve generating randomized numbers, potentially allowing the global optimum to be found; however, it also indicates a lack of reliability, and a poor solution could be generated.

2.5 Derivative calculation

Using gradient-based optimization methods, first and even second derivatives are normally required. In engineering practice, derivation of exact expressions of derivatives can be cumbersome (e.g. deriving the second derivatives has been repeatedly reported as being quite difficult). For this reason alternative approaches have been proposed by researchers.

2.5.1 Quasi-Newton methods

The so-called *Quasi-Newton methods* were developed to address the complexity in calculating the second derivatives (i.e., the Hessian matrix). They approximate the Hessian matrix using a positive semi-definite or definite matrix constructed using gradient information obtained in the current and previous iterations. Various methods have been developed to construct this matrix; one of the most commonly used methods is the Broyden, Fletcher, Goldfarb and Shanno (BFGS) method. For many non-linear problems, constructing an approximate Hessian matrix is more efficient than using its exact solution, so calculating a search direction is computationally inexpensive in general; however, the approximate direction may not be equally efficient, leading to a relatively slow convergence speed (i.e. requires more steps to move a point to a minimizer). Today, many software packages provide options to use Quasi-Newton approximations as an alternative means of supplying second derivatives (e.g., IPOPT by Vigerske & Wachter 2013).

2.5.2 Automatic Differentiation

Sometimes even the first derivatives can be quite complicated to derive by an analytical approach. In this case the well-known *finite-difference* method can be adopted. However, the use of the finite-difference method causes many problems: first, the calculated first derivatives and hence the second-order terms can be inaccurate; second, it is likely to be computationally expensive.

Alternatively, derivatives can be evaluated using the so-called *automatic differentiation* (AD) method (e.g., Griewank & Walther 2008), which obtains highly accurate derivatives of functions without deducing their analytical expressions. The AD algorithm is based on the well-known 'chain rule'; it involves a sequence of simple elementary operations that can be performed automatically under the 'object oriented programming' framework (e.g., discussed in Neidinger 2010).

For example, to calculate $\frac{d}{dx} \sin(x^2)$ at x_0 , $\frac{d}{dx} \sin(x^2) = \cos(x_0^2) \frac{dx^2}{dx}$ is calculated and then $\frac{dx^2}{dx} = 2x_0$. By overwriting operators (e.g., let $\sin(*) = \cos(*)$ and $*^2 = 2 \times *$, where $*$ is any mathematical expression), the above sequence of calculation can be automated. Unlike the finite-difference method, AD accurately evaluates derivatives. Furthermore, some research has shown that AD can be as efficient as using analytical expressions (e.g., Forth et al. 2004). For these reasons AD has been adopted in many engineering problems (Corliss et al. 2002). Software packages have been made available such as MAD, a package in TOMLAB, which permits AD to be used in MATLAB. Though analytical expressions are deduced in this thesis, AD provides a means of verifying these expressions.

2.6 Conclusion

Mathematical optimization provides a means of solving engineering problems in this thesis.

- Linear programming (LP) problems involve solely linear functions for which highly efficient methods have been developed. In addition, LP problems are convex, permitting the global optimum to be obtained.
- Non-linear programming (NLP) problems are typically more difficult to solve, particularly constrained NLP problems. The Karush-Kuhn-Tucker (KKT) conditions provide a method for locating a local minimizer using information provided by first derivatives of the functions involved. Using KKT conditions, the interior point method seeks a local optimum by iteratively solving a system of linear equations that can be tackled very efficiently using modern linear solvers, allowing relatively large-scale problems to be solved efficiently.

Chapter 3

Literature review

This thesis covers two major areas: the design of light-weight truss structures and yield-line analysis of reinforced concrete slabs. However, the approaches applied to these two areas have purposely not been separated, since both problems can be tackled using mathematical optimization methods introduced in Chapter 2. They also involve the use of plastic methods.

3.1 Plasticity theory and limit analysis

It is recognized that ductility exists in many concrete structures (i.e., strength is dominated primarily by reinforcement). Stress and strain behaviours due to material ductility are addressed in plasticity theory, which is built on certain experimental observations and then idealized using an appropriate mathematical formulation (Chakrabarty 2006). A comprehensive review of plastic theory for concrete structures is given by Nielsen & Hoang (2011), which discusses the history of plastic theory and its applications for concrete structures (e.g., beams, slabs, etc.). In this thesis, plastic behaviour of material is assumed; the reason for this assumption will now be explained.

3.1.1 Plastic vs. elastic

Structural analysis based on plasticity and elasticity principles are quite different. When using elastic methods it is usual to assume that a structure will fail when a critical stress is reached at any point in the structure. However, structural design theories based on plastic methods account for the fact that the structure will continue to carry load when stresses reach a plastic limit, until a failure mechanism is formed. Consequently, elastic design methods result in conservative, potentially uneconomic and wasteful, solutions. Furthermore, certain plastic models permit relatively simple mathematical formulations to be derived for structural analysis and design, leading to efficient approaches that can be readily used with confidence in industry. Therefore, plasticity is adopted here, with two idealized plastic models are explored.

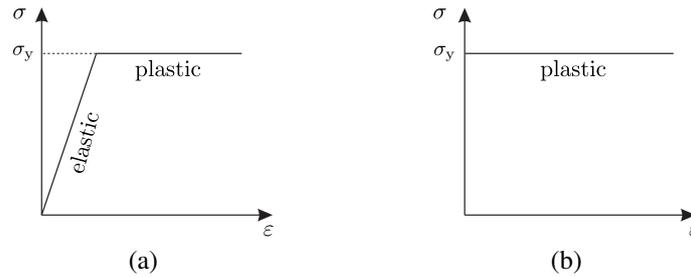


Figure 3.1: Stress-strain relations for idealized (a) elastic-plastic, and (b) rigid-plastic materials

3.1.2 Idealizations of plasticity

The real stress-strain behaviour of ductile materials (e.g., mild steel) may include strain hardening or softening. However, two idealized models are often used in engineering practice to describe the stress-strain relations in the plastic phase: elastic-plastic and rigid-plastic, as shown in Fig. 3.1(a) and (b) respectively (Rees 2006).

For the elastic-plastic material model, stress and strain are linearly coupled initially; however this coupling effect is eliminated in the plastic phase. For the rigid-plastic model, elastic strain is completely ignored; no deformation exists unless stress is sufficiently large, leading to a so-called perfectly plastic deformation, where deformation is increased steadily without changing the stress. Though no real rigid-plastic material exists, this model is applicable when plastic strains are much larger than elastic strains (Nielsen & Hoang 2011).

3.1.3 Yield criteria and the associated flow rule

It is clear from Fig. 3.1 that the yield stress σ_y determines the load carrying capacity of a material. This yield stress can be described in terms of either normal or shear stress, or a combination of both depending on the material involved. Various failure criteria exist to address this, where hypotheses are made according to characteristics observed from the failure mode. A general failure criterion can be written as:

$$g(\sigma_1, \sigma_2, \sigma_3) \leq 0, \quad (3.1)$$

where g is the function describing a failure criterion with respect to the principal stresses σ_1 , σ_2 , and σ_3 . Specifically, the point set where equality is satisfied in (3.1), i.e., $g(\sigma_1, \sigma_2, \sigma_3) = 0$, is often called the yield surface. Perfect plasticity allows unlimited deformation to occur when the equality condition is satisfied. Deformation can have more than one dimension and the relationship between individual deformed directions is governed by rules of plastic flow, which can also be affected by the yield surface (Calladine 1969).

The use of yield criteria and the associated flow rule allows the load carrying capacity of a ductile structure to be estimated, for example, using limit analysis.

3.1.4 Limit analysis

Limit analysis is an engineering approach used to estimate the load capacity of a ductile structure. In the case of a brittle material, stresses can no longer be sustained once the failure stress has been reached; in contrast stresses can continue to be transmitted in the case of a ductile material. In a structure formed using a ductile material, yielding will spread to adjoining sections such that the external load can be increased until the yielding can spread no further. Prior to the collapse of a ductile structure, a statically admissible stress field can be found that is in equilibrium with applied loads without exceeding the yield criterion anywhere in the structure. In contrast, when the structure is collapsing, a kinematically admissible velocity field is formed, permitting plastic deformation to occur. Approaches can be developed to estimate the collapse load by assuming these two states.

3.1.5 Lower and upper bound theorems

The so-called *lower bound* and *upper bound* theorems are two of the most important theorems in limit analysis. In a lower-bound analysis a statically admissible stress field is obtained using equilibrium relations and checking that the yield condition is not violated at any location. This leads to a safe estimate of the load capacity of a structure - a collapse load calculated from any possible statically admissible stress field cannot be higher than the true collapse load. In contrast, in an upper bound analysis collapse is assumed to have occurred. Given a failure mechanism, the work method can be used to evaluate the external work, and hence the collapse load. However, the true failure mechanism must have a collapse load which is the lowest of all failure modes; this implies that an upper bound analysis may lead to an overestimate of the load capacity of a structure.

Designs using the lower bound method are often conservative and hence lead to uneconomic design solutions; in addition, the failure mechanism is not immediately apparent by interpreting the stress field. The velocity field in upper bound analysis provides a direct visual interpretation of the associated failure mechanism. However, given the fact that the upper bound method is potentially non-conservative, and that an erroneous solution may lead to severe consequences, its usefulness has been questioned (e.g., Johnson 2006).

3.2 Limit analysis of truss structures

Limit analysis can be applied to many ductile structures. One of its application is to estimate the load carrying capacity of truss structures. Since truss bars take only axial forces, the yield criterion (3.1) is written as: $-\sigma_y^- \leq \sigma_1 \leq \sigma_y^+$, where σ_y^- and σ_y^+ are compressive and tensile yield stresses, respectively. Given a truss structure under prescribed load and support conditions, the axial force in every truss bar can be calculated using equilibrium, by resolving forces at every node (also assuming material stiffness is known in an elastic analysis). Its load carrying capacity can then be estimated using the above yield criterion.

Figure 3.2 shows a three-bar truss as an example. In an elastic analysis, the load carrying capacity

is defined as the maximum load without causing yielding at any part of the structure. So this capacity is $P = 0.914A\sigma_y$ when bar 1 is yielding with a stress equal to $\frac{0.5A\sigma_y}{0.5A} = \sigma_y$. Nonetheless, the external load P can still be increased without collapsing the whole structure, despite bar 1 experiencing plastic deformation. In plastic limit analysis, the loading capacity is defined as the maximum load the whole structure can sustain. When further increasing load P , the axial forces continue increasing in bars 2 and 3, until bar 3 yields and the whole structure collapses. Thus, $P = 1.061A\sigma_y$ is the maximum load capacity of the structure obtained via a plastic limit analysis.

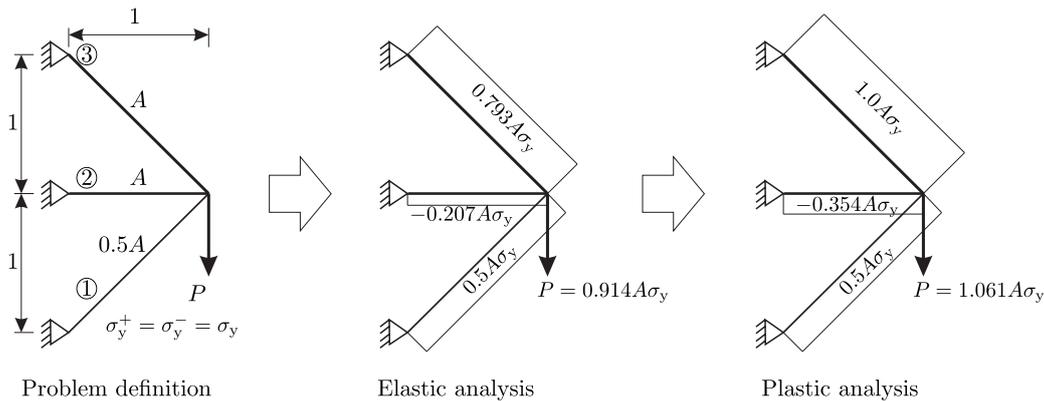


Figure 3.2: Limit analysis for a truss structure

An interesting topic is the reverse problem of limit analysis. It raises the question of how to determine the best (e.g., lightest) structure under prescribed loading and support conditions. The topic turns out to be rather challenging, and is introduced in the following section.

3.3 Structural optimization

One of the most challenging engineering topics is the allocation of materials for structures such that every part of the structures functions at its maximum capacity. A structure is often designed to meet some given criteria (e.g., to sustain a certain applied load) while one or more measured structural costs (e.g., structural weight) are minimized. In computational design approaches, structures should be ‘optimized’ for this purpose (hence the term *structural optimization*).

Depending on the structural parameters to be optimized, structural optimization can be categorized as one, or even a combination, of the following three categories: size, geometry/shape and layout/topology optimizations; The terms ‘geometry’ and ‘layout’ are often used when trusses are concerned, while ‘shape’ and ‘topology’ are used for continuum structures. The three categories are illustrated in Fig. 3.3.

Of these three categories, layout/topology optimizations are usually considered the most challenging (e.g., Cheng 1995). Two branches of research have developed to tackle layout/topology optimization problems: (i) heuristic approaches (e.g., Xie & Steven 1993, Querin 1997, Kane & Schoenauer 1996, Wang & Tai 2005, Tai & Akhtar 2005), and (ii) gradient-based approaches (e.g., Bendsøe & Sigmund 2003, Rozvany & Lewiński 2014). There has been significant activity in both branches (Sigmund 2011). The former branch is normally described as inefficient because of its lack of mathematical framework; however, it can be ‘versatile’ as a wide range of problems can

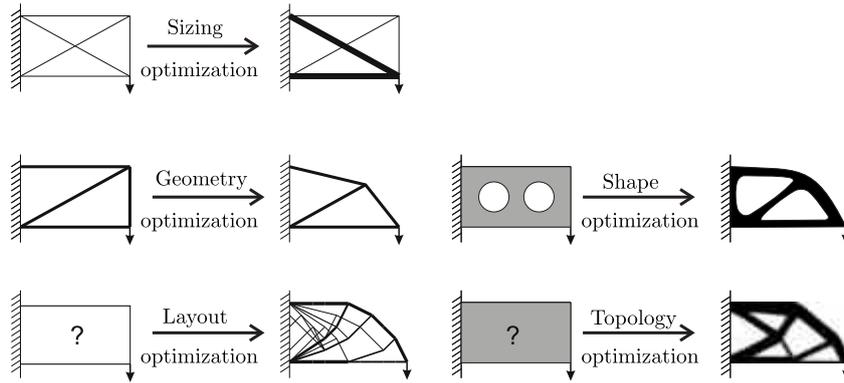


Figure 3.3: Types of structural optimization problem

be tackled and because many optimization methods used in this branch (e.g., GA in Section 2.4.5) do not involve mathematical derivations of the particular problems being investigated. In contrast, the latter branch often requires some mathematically derived information (e.g., first derivatives) to be supplied, and generally results in a relatively rapid optimization process. For a particular type of optimization problem, its mathematical properties need to be investigated. Since efficiency and reliability are important in industry, this thesis focuses on the latter branch. The nature of trusses and continuum structures are quite different; consequently the approaches used in the fields of layout and topology optimization are also different.

3.3.1 Methodologies in layout optimization

For a truss structure under prescribed loading and support conditions, layout optimization seeks to find the optimum arrangement of bars by minimizing the cost function (e.g., volume). Approaches to solving this problem fall into two main categories: numerical and analytical approaches.

Numerical approaches

Numerical approaches take advantage of well-developed mathematical optimization methods, with a view to finding the optimal connectivity of nodes in a design domain that contains nodes interconnected by potential members (leading to a ‘ground structure’, as shown in Fig. 3.4). In these problems, the loads, support conditions and material strengths are prescribed. Considering a rigid-plastic material response (as shown in Fig. 3.1b), equilibrium equations are established on nodes using the axial forces of truss bars; the optimization formulation is as follows (Dorn et al. 1964):

$$\min_{\mathbf{q}} \quad V = \mathbf{c}^T \mathbf{q} \quad (3.2a)$$

$$\text{s.t.} \quad \mathbf{B}\mathbf{q} = \mathbf{f} \quad (3.2b)$$

$$\mathbf{q} \geq 0, \quad (3.2c)$$

where:

V is the total volume of the truss structure, and \mathbf{c} contains coefficients constructed using lengths, tensile and compressive yielding stresses of bars. \mathbf{B} is an equilibrium matrix, while \mathbf{q} and \mathbf{f}

represent internal and external forces, respectively. Problem (3.2) takes the form of the standard LP problem (2.5) and can be readily solved using the methods introduced in Section 2.2.

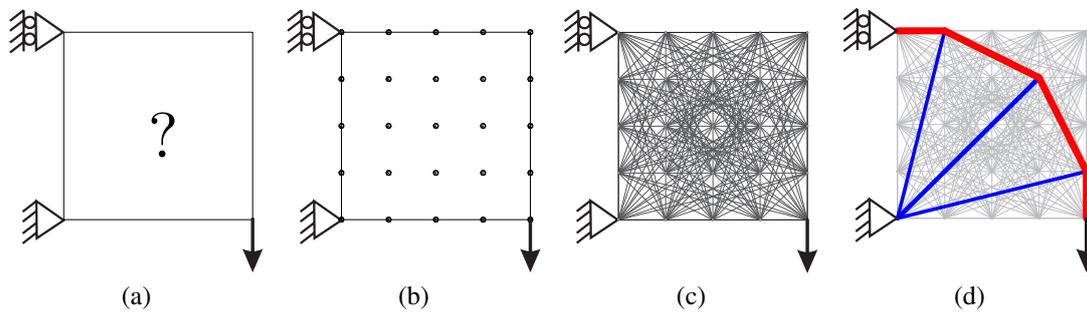


Figure 3.4: Steps in layout optimization: (a) problem specification; (b) nodal discretization; (c) forming the ‘ground structure’; (d) identifying the most efficient arrangement of bars

Although it was developed half a century ago, this approach has not been widely used in engineering practice. First, computational implementations of LP were not well-developed to cope with the large-scale problems associated with truss layout optimization. Second, computer-aided design was not popular until recently. Finally, traditional fabrication techniques were unable to produce truss structures of complex geometries. These obstacles were recently overcome using modern technology (e.g., MOSEK 2011, LimitState 2008, Excell & Nathan 2010), so that layout optimization could become a productive area of research that can directly benefit industry.

One of the most serious drawbacks in layout optimization using a ground structure approach (e.g., Fig. 3.4) is that problem size grows extremely quickly as the number of nodes employed increases (e.g., a total of $\frac{m(m-1)}{2}$ bars when every pair of nodes are connected, where m is the number of nodes). Typically, more efficient structures can be obtained by using more nodes. However, even modern computer software and optimization tools cannot solve the resulting LP problems which become enormously large. Thus, significant effort has been put into addressing this issue. In Gilbert & Tias (2003), a so-called *member adding scheme* was proposed so ground structures with more than 100,000,000 bars could be tackled effectively. Later work involving the use of the member adding scheme can be seen in Pritchard (2004), Pritchard et al. (2005), Tias et al. (2006) and Sokół (2011).

While structurally efficient trusses can now be generated by using a dense nodal grid in the ground structure, the resulting layouts often become very complex and cannot be fabricated, even with modern additive manufacturing techniques (Smith et al. 2015). This highlights the need for more rational layouts that can be used in engineering practice. Another concern about layout optimization is the fact that a uniformly distributed nodal grid, as shown in Fig. 3.4, has difficulty representing curved structures. To address this, researchers have employed the so-called *unstructured nodes* (e.g., Zegard & Paulino 2014). However, results are not always satisfactory and a potentially superior means of generating rationalized structures is to perform a geometry optimization step, allowing the positions of nodes to be adjusted.

Analytical approaches

Analytical approaches have been developed in parallel with numerical approaches. Instead of deriving numerical solutions that typically have errors, analytical approaches consider theoretical aspects, aiming to obtain necessary and sufficient conditions for the optimum structures in order to deduce exact layouts that have the least structural volume. Research in this field was pioneered by Michell (1904); his optimality criteria was later made more general by Hemp (1973):

A pin-jointed framework has the least volume of material, if it can carry its given forces, with stresses in its tension members equal to σ_T and stresses in its compression members equal to $-\sigma_C$ and if a virtual deformation of a region of space, in which the competing frameworks must lie, satisfies the kinematic conditions imposed on the framework and gives strains of σ_ϵ/σ_T in its tension members, strains of $-\sigma_\epsilon/\sigma_C$ in its compression members and has no direct strain lying outside these limits.

Under this criterion, optimum truss structures contain an infinite number of bars with infinitesimal areas, which are also called *Michell structures* or *Michell continua*. Research on Michell's theorem has been the subject of intense interest. Early work can be found in Chan (1962) and Hemp (1974), where Michell structures were sketched for a class of cantilever trusses and that of pin-supported trusses under uniformly distributed load (UDL), respectively. In Hemp (1973) and Cox (1965), comprehensive theoretical studies of Michell structures were given; in Prager (1978*b,a*), concerns were expressed regarding derivation of optimum layouts having a finite number of nodes. More recently, limitations in Michell's theory were explained in Rozvany (1996). Some researchers have devoted themselves to deriving analytical solutions for various problems (e.g., Lewiński et al. 1994*a,b*, Sokół & Rozvany 2012, Rozvany 1998, Lewiński & Rozvany (2007, 2008*b,a*), Lewiński et al. (2013), Graczykowski & Lewiński 2005, 2006*a,b*, 2007*a,b*).

Interaction between numerical and analytical approaches

Numerical layout optimization approaches provide a means of checking the validity of analytical solutions and vice versa. Due to limitations in computing power and a lack of highly efficient optimization methods, approximations were made, leading to less accurate solutions. For instance, numerical results in Darwich et al. (2010) suggested that, to carry a uniform load between pinned supports, the then widely-accepted parabolic funicular was non-optimum. Later, Tyas et al. (2010) used analytical methods to confirm the correctness of the proposed new layout and the associated volume. This is not the only case, to design the optimal structure to carry a uniformly distributed load between two pinned supports, Hemp (1974) derived its analytical solution with certain limitations. After conducting both theoretical and numerical studies, Pichugin et al. (2012) investigated its limitation and extended the scope of Hemp's original work.

Today, it is quite common in the field of structural optimization for researchers to present theoretical outcomes in conjunction with numerical validations (e.g., Sokół 2011, Sokół & Rozvany 2012, Pichugin et al. 2015), and vice versa (e.g., Martínez et al. 2007, Mazurek et al. 2011, Mazurek 2012).

3.3.2 Methodologies for topology optimization

The boom in topology optimization for continuum structures started in 1988, after the advent of the so-called homogenization method by Bendsøe & Kikuchi (1988). This approach was intensively studied by several researchers and the simplified isotropic material with penalization (SIMP) method (i.e., Bendsøe 1989, Zhou & Rozvany 1991, Mlejnek 1992) was developed. Those contributions, along with the subsequent comprehensive studies of Bendsøe & Sigmund (1999) and Bendsøe & Sigmund (2003), inspired many researchers in the field, increasing research interest in topology optimization. According to recent review papers (e.g., Rozvany 2009, Sigmund & Maute 2013), there are numerous means of attacking topology optimization problems. For example, methods developed other than SIMP include the level set (Allaire et al. 2002, 2004), topology derivative (Sokolowski & Zochowski 1999), phase field (Bourdin & Chambolle 2003) and evolutionary (Huang & Xie 2010*b,a*) approaches. Another factor that extends the community of topology optimization is the common sharing of software source code (e.g., Sigmund 2001, Andreassen et al. 2011), attracting new researchers into the field.

SIMP remains the most popular among all methods available today. It is based on a ‘density approach’: to determine the material distribution of the structures, the material ‘density’ of every structural element (i.e., a finite element mesh) is viewed as optimization variables that are assumed to be varying from ‘0’ to ‘1’^[i]. In addition, certain mechanical properties (e.g., Young’s modulus) have been considered as functions with respect to material density, attempting to establishing relationships between the distribution of material and the resulting structural behaviour. Therefore, the objective function (e.g., compliance of a structure) can be expressed using material densities. An optimization is thus performed to calculate the value of optimization variables (i.e., densities in every structural element) that directly indicate the distribution of materials. However, since it assumes density can vary continuously, it has been reported that the solutions may contain ‘ersatz material’ (i.e., density between 0 and 1) that do not exist. To address this issue, methods using filter functions have been adopted to eliminate the ersatz materials (e.g., Sigmund 2007).

Among mathematical optimization tools, the most commonly used method in this field is MMA (Section 2.4.3), which was initially designed for solving topology optimization problems. For a typical topology optimization problem, calculating derivatives can be computationally expensive; thus, using MMA normally leads to very efficient processes. In contrast, the use of conventional mathematical optimization methods (e.g., the interior point method considered in Section 2.4.2) has not been widely explored in this field.

3.3.3 Geometry/shape optimization

From a computational perspective, geometry/shape optimization often involves seeking the optimum locations of nodes that have been employed. With topology optimization, the need for subsequent shape optimization of continuum structures is not always required, since shape optimization is often already incorporated (e.g., in the level set and topology derivative approaches). Given a dense nodal grid and a proper filter method, the shapes of the resulting structures can

^[i]Conventionally, a density of ‘0’ (or near ‘0’) means no material, and ‘1’ for solid material.

readily be identified without moving nodal positions in the mesh.

In contrast, for discrete structures (i.e., trusses), geometry optimization is not truly embedded in the layout optimization process. Given a dense nodal grid, the resulting layouts and geometries often become very complex. Therefore, geometry optimization is of particular interest. Although layout optimization is viewed as the most challenging category, the use of a ground structure transforms the problem into a less challenging sizing optimization, which is relatively easy to solve from a mathematical perspective (e.g., an LP problem in (3.2)). However, a geometry optimization problem often comes with the sizing problem - moving nodes inevitably changes the load path and sizes of bars are changing accordingly. The resulting optimization problem can be written, for example, as:

$$\min_{x,y,q} V = \mathbf{c}(\mathbf{x}, \mathbf{y})^T \mathbf{q} \quad (3.3a)$$

$$\text{s.t.} \quad \mathbf{B}(\mathbf{x}, \mathbf{y})\mathbf{q} = \mathbf{f} \quad (3.3b)$$

$$\mathbf{q} \geq 0, \quad (3.3c)$$

where x and y denote x - and y - coordinates of nodes. With respect to optimization variables (i.e., x , y , and q), (3.3) is no longer an LP, but is instead a constrained NLP problem. In this sense, geometry optimization is more challenging than layout optimization so satisfactory approaches have not been developed. McConnell (1974) was one of the early researchers who used geometry optimization to improve truss designs. In Gil & Andreu (2001), the sizing and geometry problems are separated and tackled using different strategies. However, decoupling sizing and geometry variables potentially affects the solutions produced; in Czarnecki (2003) and Martínez et al. (2007), trusses are restricted to be statically determinate, taking the equilibrium condition (3.2b) in the objective function; thus, statically indeterminate structures and those in unstable equilibria cannot be tackled. Instead of using gradient-based approaches, some researchers employ heuristic methods to solve the problem (e.g., Azid et al. 2002, Rahami et al. 2008).

One issue in geometry optimization is that the resulting layout depends highly on the initial layout employed; for this reason, it cannot normally ensure a global optimum is found. To address this problem, some researchers suggest a so-called *combined layout and geometry optimization*, seeking the optimum nodal distribution in the design domain and leading to the least volume. It can be written as a two-level (min-min) hierarchical formulation:

$$\min_{x,y} \min_q V = \mathbf{c}^T \mathbf{q} \quad (3.4a)$$

$$\text{s.t.} \quad \mathbf{B}\mathbf{q} = \mathbf{f} \quad (3.4b)$$

$$\mathbf{q} \geq 0. \quad (3.4c)$$

Optimization of this type is typically non-smooth (e.g., Bendsøe & Sigmund 2003) and solving this problem can be computationally expensive. For example Achtziger (2007) and Descamps & Filomeno Coelho (2013) only tackle relatively small-scale problems, even without employing a fully connected ground structure.

3.4 Plane plasticity problems and discontinuity layout optimization

The theoretical study of optimum truss layouts in the last century was also influenced by research in plane plasticity problems. It was found that the Michell continua (Michell 1904) were remarkably similar to the so-called slip-line fields in plane plasticity problems - both involve a special geometric form known as ‘Hencky-Prandtl nets’. Theories developed in plane plasticity problems were transferred to truss design problems (e.g., Hemp 1958, Prager 1958, Johnson 1961, Strang & Kohn 1983). However, the numerical solution methodologies developed in truss design problems (e.g., layout optimization) were not transferred to plane plasticity problems until this century, when ‘discontinuity layout optimization’ (DLO) was developed (Smith & Gilbert 2007).

Instead of seeking the best arrangement of bars in a truss structure, DLO identifies a velocity field (i.e., failure mechanism) consisting of truss-like discontinuities, in which plastic deformation occurs. By performing a standard layout optimization (e.g., solving problem (2.5)), the critical failure mode can be found. The use of DLO permits highly efficient methods developed in the field of truss optimization to be effectively employed in identifying plastic failure patterns in limit analysis. Some extensions of DLO can be seen in Gilbert et al. (2010) and Hawksbee et al. (2013). Similarly, Gilbert and the author of this thesis, et al., have recently developed a DLO approach that can be applied to the limit analysis of reinforced concrete slabs (Gilbert et al. 2014); while full details of this contribution are given in Appendix A, the historical developments involved in slab analysis are introduced here.

3.5 Limit analysis methods of reinforced concrete slabs - historical methods using hand analysis

A specific application of interest in this thesis is the analysis of reinforced concrete slabs (henceforth called slabs), considered assuming the following: (a) a rigid-plastic material behaviour is assumed; (b) the thickness of the slab is small compared to its other dimensions; (c) loads are applied normally to the slab surface; (d) the slab is in a state of bending, and membrane stresses are negligible. Yielding is considered as a result of moments. The bending and torsional moments are often described as generalized stress (e.g., Nielsen & Hoang 2011, Park & Gamble 2000, Chakrabarty 2010); therefore, the stress state at any point in the slab is expressed in Fig. 3.5, where m_x , m_y are the bending moments written in Cartesian coordinates, and m_{xy} and m_{yx} are the torsional moments.

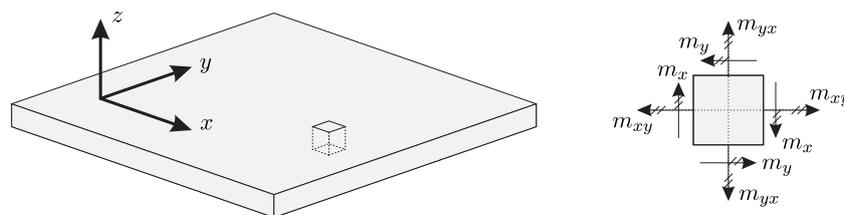


Figure 3.5: Moments at a point in a slab as generalized stresses

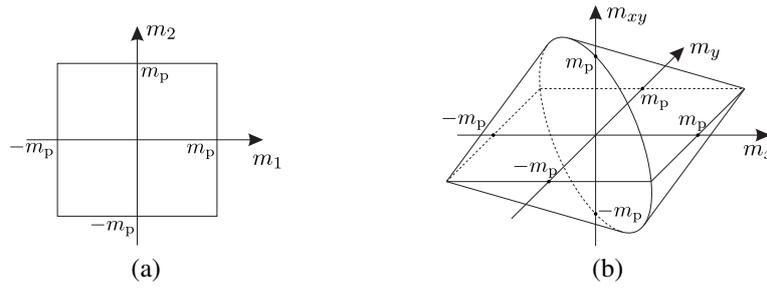


Figure 3.6: Square yield criterion using: (a) principal moments; (b) moments in a Cartesian coordinate system

The principal moments m_1 and m_2 can be derived as:

$$m_{1,2} = \frac{m_x + m_y}{2} \pm \sqrt{\left(\frac{m_x - m_y}{2}\right)^2 + m_{xy}^2}. \quad (3.5)$$

Various yield criteria can be formulated, by using the principal moments m_1 and m_2 . One of the most commonly used criteria is the *square yield criterion* (Johansen 1943); for slabs which are isotropically reinforced in the x and y directions, and in the top and bottom layers, this is given as:

$$-m_p \leq m_1 \leq m_p, \quad (3.6a)$$

$$-m_p \leq m_2 \leq m_p, \quad (3.6b)$$

where m_p is the plastic moment of resistance. A graphical interpretation of this criterion is depicted in Fig. 3.6(a) using principal moments. It has a square form, hence the name square yield criterion. Alternatively, using moments in a Cartesian coordinate system (Fig. 3.5), a graphical interpretation of the criterion which consists of two cones is given in Fig. 3.6(b).

In the case of orthotropic reinforcement, Johansen (1943) proposed a *stepped yielding criterion*:

$$m_p^+ = m_{px}^+ \cos^2 \phi + m_{py}^+ \sin^2 \phi, \quad (3.7a)$$

$$m_p^- = m_{px}^- \cos^2 \phi + m_{py}^- \sin^2 \phi, \quad (3.7b)$$

where, m_p^+ and m_p^- denote the positive and negative plastic moments of resistance per unit length along a yield-line that has an angle of ϕ to the x - axis of the Cartesian coordinate system. m_{px}^+ and m_{py}^+ are the positive plastic moments of resistance per unit length in the \vec{X} and \vec{Y} directions, respectively; m_{px}^- and m_{py}^- are the negative moments. Note that sometimes the yield-line angle ϕ is unknown, implying that criteria (3.7) cannot readily be used in these cases (Nielsen & Hoang

2011). Alternatively, by eliminating ϕ , (3.7) can be reformulated in the Cartesian coordinate as:

$$m_x \leq m_{px}^+, \quad (3.8a)$$

$$m_y \leq m_{py}^+, \quad (3.8b)$$

$$m_x \geq -m_{px}^-, \quad (3.8c)$$

$$m_y \geq -m_{py}^-, \quad (3.8d)$$

$$(m_{px}^+ - m_x)(m_{py}^+ - m_y) \geq m_{xy}^2, \quad (3.8e)$$

$$(m_{px}^- + m_x)(m_{py}^- + m_y) \geq m_{xy}^2. \quad (3.8f)$$

Various analysis methods have been developed which use these failure criteria. Methods using hand analysis will be introduced in this section, while computer-based approaches will be described in Section 3.6.

3.5.1 Hillerborg's strip method

When concerned with conservative designs (i.e., lower bound analysis), the *strip method* (Hillerborg 1956) is often used. A slab is divided into horizontal and vertical strips, and external load is assumed to be carried by slab strips individually in the X (horizontal) and Y (vertical) directions. Let q denote the external load and assume $q = q_x + q_y$, where q_x is carried by X strips, and q_y by Y strips. Considering the equilibrium condition, each strip is determined to be a beam element whose load carrying capacity can be calculated using beam theory. Then, the strength of the slab is determined by assessing the load effects on all beam elements. It is worth noting that, the distribution of q_x and q_y is normally chosen subjectively by a designer's experience; for a simply supported rectangular slab under a uniformly distributed load q , a distribution of q_x and q_y has been assumed in Fig. 3.7, where two strips are analysed: 1-1 in the X direction and 2-2 in Y . Similarly, other strips in the X and Y directions need to be examined to find the collapse load.

Since calculating the load capacity of each strip is simple, the strip method is easy to use in practice. To incorporate column supports, the advanced strip method was proposed by Hillerborg (1959) and further discussed in Hillerborg (1982). However, a major drawback of using Hillerborg's strip method is the fact that it often leads to very conservative designs.

3.5.2 Yield-line theory

The development of yield-line was pioneered by Ingerslev (1923), who made the assumption that 'the bending moment is distributed across the main lines along which rupture takes place'. This approach was further investigated by Johansen (1943), who proposed the remarkable *yield-line theory*. The theory assumes that yielding occurs along certain lines (i.e., yield-lines) across the slab, while the other parts remain rigid. To perform an analysis, a system of yield-lines is required, forming a collapse mechanism and known as a yield-line pattern (e.g., Fig. 3.8), to be prescribed for any given slab problem. Since it assumes a collapse state, it is an upper bound analysis; the associated collapse load can be calculated via two approaches: (i) the virtual work method and (ii) the equilibrium method.

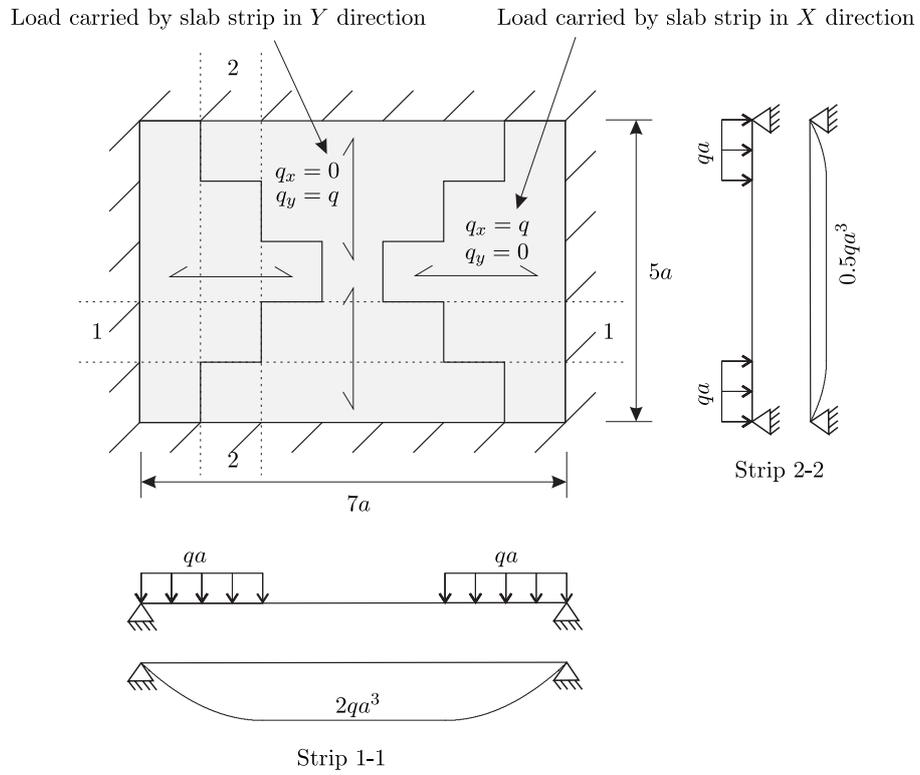


Figure 3.7: Strip method: load to be carried on X and Y strips

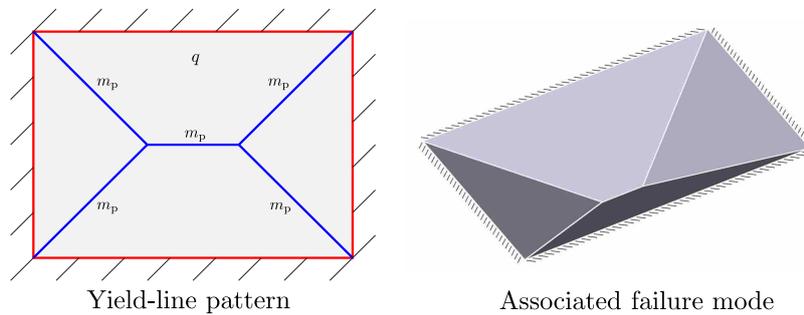


Figure 3.8: Yield-line pattern for a simply supported rectangular slab under distributed load (blue lines represent sagging; red lines represent hogging)

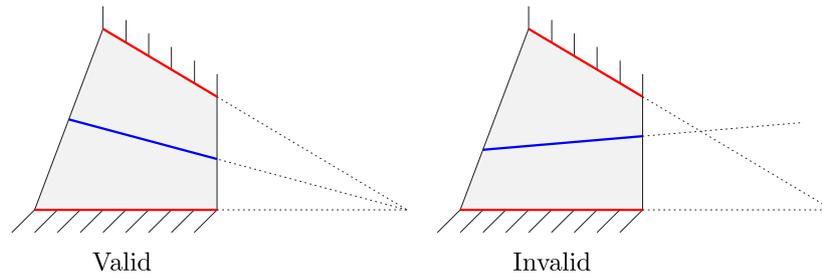


Figure 3.9: Kinematically admissible failure pattern required for yield-lines

(i) The virtual work method. The yield-lines become plastic hinges when the slab is collapsing; the work done by external load must equal the dissipation of internal energy among all yield-lines. It is worth noting that yielding must form a kinematically admissible failure pattern (as shown in Fig. 3.9), where the slab can potentially deform along yield-lines without using additional plastic hinges that have not been prescribed.

The prescribed yield-line pattern is vitally important; to find the most critical case in a given pattern, the geometric parameters (e.g., position and angle of lines) are typically adjustable. A differentiation process (e.g., in Johansen 1943, Park & Gamble 2000) may then be performed. Nevertheless, a prescribed pattern has only limited geometric variations; for this reason, engineers often postulate a few potential yield-line patterns in practice, and then calculate the associated collapse load for all of them to discover the most critical scenario. However, these repetitive calculations can become tedious (Kennedy & Goodchild 2004). To avoid the use of differentiation, and thus to make hand analysis easier to perform, the equilibrium method was proposed.

(ii) The equilibrium method (also named ‘nodal force’, or ‘formulae’ method). For a slab that has been ‘divided’ into rigid segments using yield-lines, equilibrium conditions need to be satisfied in each of the segments, taking into account the moments and forces distributed along its transverse sections (i.e., prescribed yield-lines) and the external load. Given any transverse section, besides the bending moment, torque and shear forces may exist. Johansen (1943) suggested that the distribution of torque and shear force was not of significant interest, so the load effect could effectively be taken into account using equivalent out-of-plane forces acting on the two end nodes of the section (the yield-line). This can be calculated by resolving the distributed forces on end nodes. The equivalent forces acting on nodes are therefore named ‘nodal forces’, which can be written as functions with respect to bending moments and geometrical parameters of the yield-lines. Taking nodal forces, problems can be formulated using equilibrium conditions; formulae can be deduced, giving an explicit expression of the collapse load with respect to known parameters (e.g., geometry, material strength) of the slab problem. A differentiation process is no longer required, leading to very efficient approaches.

It is worth noting that, the ‘equilibrium’ method is *not* a lower bound analysis, despite its name. It requires that the yield-line bending moment, nodal force and external load are in equilibrium, but does not ensure moments within the slab segments satisfy the yield criteria. Some researchers (e.g., Wood & Jones 1967, Nielsen & Hoang 2011) have pointed out that the ‘equilibrium’ method is in fact another form of the work method. Under Johansen’s square failure criterion, yielding

occurs when principal moments (i.e., m_1 , m_2) equal the yielding moment; this implies moments taken at any point of a yield-line must lie on the yielding surface (i.e., equality is satisfied in (3.6)). Therefore, if a transverse section is truly a yield-line, torque along this line must be zero; otherwise there must exist another direction in which the principal moment is larger, leading to constraint (3.6) being violated. Similarly, since the bending moment is at its maximum along a yield-line, the shear force must be zero. Nielsen & Hoang (2011) note that, when torque and shear forces are zero along every internal yield-line, the equilibrium and the virtual work methods lead to the same predicted collapse load.

Since the yield-line method is an upper bound analysis, errors incurred in the calculation will lead to unsafe designs. Though extensively studied (e.g., Johansen 1968, Wood & Jones 1967, Park & Gamble 2000, Nielsen & Hoang 2011, etc.) and adopted in engineering practice (e.g., Kennedy & Goodchild 2004), the method cannot normally be used by engineers who have insufficient experience. To analyse real slab problems that are potentially complex (e.g., in shape), it is difficult to postulate (near-)correct yield-line patterns; even an expert engineer may not have strong confidence in their solution (e.g., Johnson 2006).

Computational methods have been developed to generate reasonable yield-line patterns without depending on the experience of the designer. This also frees engineers from tedious calculations. These approaches often require mathematical optimization methods to identify the critical failure mechanism.

3.6 Computer-aided methods in limit analysis of slabs

The limit analysis methods introduced in Section 3.1.4 are based on hand calculations; various numerical methods have been developed that take advantage of mathematical optimization methods. Some early research was undertaken by Hodge & Belytschko (1968) and Chan (1972), who attempted to obtain lower and upper bound solutions, bracketing the exact solution. In their approaches, a slab was discretized using computational elements (e.g., triangular finite element meshes), and the state variables (e.g., moments) were calculated in each element. A slab is often statically indeterminate, in which case the most critical distribution of moments (or failure mode in an upper bound analysis) needs to be identified using mathematical methods such as optimization. The load carrying capacity of the slab can normally be estimated at the same time. The efficacy of some early numerical experiments was affected by the limited computing power available at the time, and by the fact that highly efficient optimization methods (e.g., the interior point method) had not yet been developed. Today, computing power has grown significantly, and mathematical optimization methods have also improved similarly. Researchers have therefore attempted to produce more accurate numerical results via different approaches, such as introducing new computational element types (e.g., Krenk et al. 1994), adopting meshless finite element methods (e.g., Le et al. 2009), directly employing more efficient optimization methods (e.g., Thavalingam et al. 1999, Krabbenhøft et al. 2007), etc.

3.6.1 Lower bound analysis using numerical methods

In lower bound analysis, the external load needs to be maximized to reduce the gap between the numerical and exact solutions. It is common to adopt the well-known finite element (FE) method: first, moments at any point of an element can be calculated, permitting the yield criterion to be strictly satisfied at any point throughout a slab; second, for a given geometry, the FE meshes can be generated automatically, leading to a fully automated numerical process; third, employing finer meshes will produce more accurate analysis results in general, similar to when using standard FE methods; lastly, yielding points identified in an analysis can be used as a basis to automatically refine the FE mesh, potentially leading to efficiency gains. Research in this area can be seen in Krabbenhoft & Damkilde (2003), Le, Nguyen-Xuan & Nguyen-Dang (2010), Jackson (2010), and Maunder & Ramsay (2012). Relatively accurate results have been reported when using modern computing power and efficient optimization methods (e.g., less than 1% error in Jackson 2010). However, the statically admissible moment field in lower bound analysis provides no direct indication of the failure mechanism.

3.6.2 Upper bound analysis using numerical methods

Various attempts have been made to automate the hand calculations used in yield-line analysis, to obtain more accurate (and less non-conservative) design solutions. Surprisingly, ever since the 1960s, when yield-line theory was widely known in western countries, a method capable of systematically automating the yield-line method has been lacking. The major issue, which was pointed out by Hodge & Belytschko (1968) at a very early stage, is the sensitivity of the solution to the mesh pattern being used in the calculation.

Methods using FE meshes

Chan (1972) and Munro & Da Fonseca (1978) were among the first researchers to propose an FE-based approach to yield-line analysis. In later contributions, an ‘automatic yield-line analysis’ method was developed. This method employs rigid triangular FE-like elements to discretize a slab; plastic deformations, and thus yield-line rotations, were permitted along the boundaries of the elements. An LP problem was then established and solved to identify the most critical yield-line pattern.

However, the method is affected by an issue known as mesh-pattern dependency. Using the slab problem shown in Fig. 3.8 as an example: given a FE mesh pattern shown in Fig. 3.10, the true failure mechanism cannot be accurately formed. It is now well-understood that, when FE meshes are utilised, refining the mesh does not necessarily ensure that the numerical solution will converge to the exact value. Researchers tried to address this issue for decades. Some studies involve migrating nodes using a geometry optimization process, attempting to reduce the influence of the mesh pattern. Contributions can be seen in Johnson (1994) and Johnson (1995), where sequential linear programming was utilized; in Thavalingam et al. (1999), where the conjugate gradient method was employed; and in Ramsay & Johnson (1997), Ramsay & Johnson (1998), where a direct search method was adopted. Despite these various attempts, the mesh-pattern dependency has

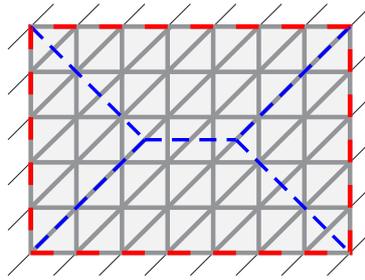


Figure 3.10: FE mesh for a simply supported rectangular slab under distributed load (blue and red dashed lines indicate the correct yield-line pattern): yield-lines from the top left to bottom right cannot be accurately represented using the given FE mesh.

not been truly solved because the geometry optimization step requires that the starting yield-line pattern is close to the true pattern; this cannot be guaranteed when starting with a solution derived from a rigid FE analysis. For example, Ramsay et al. (2015) reports that an L-shaped slab problem analysed in Ramsay & Johnson (1997) has a margin of error of 40%. Furthermore, the yield-line patterns generated by ‘automatic yield-line analysis’ can be very complex and potentially obscure; consequently the process of identifying clear yield-line patterns has been repeatedly stated to be ‘difficult’ (Johnson 1994, Jackson 2010). Thavalingam et al. (1999) even claims it is ‘up to the user to check whether alternative topologies or starting geometries need to be investigated’. The reliability of the geometry optimization processes described in the literature is therefore questionable.

Dip and strike angles method

The dip and strike angles method is an alternative means of defining the key geometric parameters of a yield-line pattern. The method was proposed by Kwan (2004) and further improved in Kwan (2013). Instead of discretizing the slab using a regular FE mesh, the method involves directly investigating the requisite kinematically admissible failure patterns (e.g., Fig. 3.9) formed at collapse, and thus describing the failure mode using dip and strike angles of the deflected rigid blocks. For this reason, the resulting failure mechanism is guaranteed to be kinematically admissible. However, it involves the use of a heuristic approach to generate yield-line patterns, and thus cannot identify certain patterns (e.g., fan-type mechanisms). Furthermore, the employed heuristic is potentially ineffective given slabs with non-convex shape, leading to incorrect solutions.

Library method

Another means of automating yield-line analysis is to examine every known yield-line pattern using computer software, without the need for potentially tedious hand calculation. It requires the use of a ‘library’ of potential yield-line patterns for given types of slab problems. The software package COncrete BRidge ASsessment (COBRAS), developed by the group led by Middleton (1997), is a case where the library method has been successfully adopted; its built-in library continuously grows to include new slab problems, leading to increasingly accurate analysis over time. The resulting patterns are relatively simple and reasonable, and can be understood by engineers.

However, given the fact that its collection of potential yield-line patterns is still very limited, particularly for slabs with irregular shapes, the method can lead to non-conservative results when the true failure mode has not been included in the library.

Jackson's user-interacted method

In Jackson (2010) and Jackson & Middleton (2013), a semi-automated yield-line analysis method using geometry optimization was proposed. Instead of generating the initial pattern using 'automatic yield-line analysis', the method involves performing a lower bound analysis and then using this as a basis to produce a yield-line pattern via user-interaction, allowing it to be semi-automated. To solve the geometry optimization problem, a Tabu search algorithm is used, a heuristic method within the category described in Section 2.4.5; consequently it is not very computationally efficient. Since it involves user-interaction and a heuristic optimization method, it may not always generate accurate upper bound solutions, and can become tedious when tackling large numbers of slab problems.

Slab analysis using DLO

As mentioned in Section 3.4, a means of automating yield-line analysis of slabs has recently been developed by a research group which includes the present author (Gilbert et al. 2014). While full details of this method are provided in Appendix A, some general remarks are now made here.

Firstly, the layout optimization technique developed for trusses (see Section 3.3.1) has been transferred to the problem of yield-line analysis, leading to a fully automated process. However, issues reported in the former have also been noticed in the later. For example, the resulting yield-line pattern is relatively complex and can be difficult to understand by engineers. Similar to the former, a means of rationalizing the resulting pattern is of particular interest.

Secondly, many slab problems in engineering practice involve complicated configurations, for example orthotropic reinforcement, knife-edge supports, point, line and patch loads; these have not been incorporated in Gilbert et al. (2014). It is therefore of significant interest to further develop the DLO method, to enable analysis of real-world slabs.

Note that, though automated, the use of the yield-line method does not provide a full solution to slab analysis problems. It considers only flexural failure; additional checks (e.g. for punching shear failure and/or serviceability limit state deflections) will however still be required. In addition, the assumption of rigid-plastic material (Fig. 3.1(b)) requires that the concrete is lightly reinforced, so that its ductility enables 'moments in indeterminate structures to redistribute themselves' (Beeby 1997).

3.7 Conclusions

This thesis involves application of plastic methods to structural design and analysis problems. Although engineering methods and tools are widely used in industry, areas for improvement remain.

- Layout optimization provides a method for designing light-weight truss structures. However, the resulting structural layouts can be complex, highlighting the need for rationalization methods.
- Taking techniques originally developed for truss layout optimization, the discontinuity layout optimization (DLO) method was conceived. With DLO the process of traditional limit analysis techniques can be automated efficiently. Considering yield-line analysis of reinforced concrete slabs, it has been shown that previously developed automated approaches are not satisfactory. DLO addresses this, permitting truly systematic automation of the yield-line method.
- However, the yield-line patterns derived from DLO can be complex in form, and therefore it is of interest to develop a means of generating rationalized yield-line patterns, which are easier to interpret. Also, it is of interest to explore the full range of applicability of the DLO method, particularly with reference to practical slab analysis and design problems.

Chapter 4

Rationalization of trusses generated via layout optimization^[i]

Preface

With the goal of developing rapid and reliable tools for the use in industry, in this thesis potential applications of numerical layout optimization methods are explored. One application involves the design of lightweight truss structures. A numerical layout optimization approach for trusses was established half a century ago by Dorn et al. (1964), and a highly efficient solution strategy was designed by Gilbert & Tyas (2003); however, it has not seen widespread adoption in industry. One reason for this is that the trusses produced using this approach are often complex in form, and hence are challenging to fabricate. Therefore, an effective means of generating more rational truss forms is of potential interest.

^[i]The content of this chapter was originally prepared for a journal paper: He, L., Gilbert, M. (2015), 'Rationalization of trusses generated via layout optimization', *Structural and Multidisciplinary Optimization* **52**, 677-694

Abstract Numerical layout optimization provides a computationally efficient and generally applicable means of identifying the optimal arrangement of bars in a truss. When the plastic layout optimization formulation is used, a wide variety of problem types can be solved using linear programming. However, the solutions obtained are frequently quite complex, particularly when fine numerical discretizations are employed. To address this, the efficacy of two rationalization techniques are explored in this paper: (i) introduction of ‘joint lengths’, and (ii) application of geometry optimization. In the former case this involves the use of a modified layout optimization formulation, which remains linear, whilst in the latter case a non-linear optimization post-processing step, involving adjusting the locations of nodes in the layout optimized solution, is undertaken. The two rationalization techniques are applied to example problems involving both point and distributed loads, self-weight and multiple load cases. It is demonstrated that the introduction of joint lengths reduces structural complexity at negligible computational cost, though generally leads to increased volumes. Conversely, the use of geometry optimization carries a computational cost but is effective in reducing both structural complexity and the computed volume.

Keywords Truss, layout optimization, geometry optimization, multiple load cases

4.1 Introduction

Numerical layout optimization provides an efficient means of identifying (near-)optimal truss layouts. The ‘ground structure’ layout optimization procedure was first proposed by Dorn et al. (1964) and more recently was made more efficient for single and multiple load case problems respectively by Gilbert & Tyas (2003) and Pritchard et al. (2005). In the latter contributions an adaptive ‘member adding’ algorithm was proposed which meant that much larger scale layout optimization problems could be solved; this and similar techniques are helping to provide new insights on a wide range of problems (e.g. Darwich et al. 2010, Sokół & Rozvany 2012, Pichugin et al. 2012). However, whilst fine numerical discretizations are needed in order to obtain highly accurate numerical solutions, the associated truss bar layouts can become very complex. Therefore identifying means of rationalizing such layouts is potentially of significant interest. Various rationalization approaches are possible, for example: (i) the problem formulation can be modified to ensure solution complexity is addressed directly from the outset; or (ii) a standard layout optimization solution can be subsequently modified in a post-processing step.

In the case of (i), directly addressing complexity within the formulation, a range of optimization methods can be applied (e.g. mixed integer linear programming, MILP, or non-classical optimization methods such as genetic algorithms); the downside of such procedures is that they are generally comparatively computationally expensive, so that only relatively small problems can be tackled. However, simple formulations are also available, and here the efficacy of the simple ‘joint length’ method proposed by Parkes (1975) will be explored. A key benefit of this method is that the linear character of the standard linear programming (LP) based layout optimization formulation is retained.

In the case (ii), addressing complexity via a post-processing step, it can be observed that the

solutions obtained from the layout optimization procedure will generally comprise far fewer bars than are present in the initial ‘ground structure’. This is significant as it means that any post-processing step need only deal with a comparatively small number of bars.

One option is to use the truss layout derived from the layout optimization process as the starting point for a geometry optimization post-processing step. Integrating layout optimization with geometry optimization has been examined before (e.g. Bendsøe et al. 1994, Bendsøe & Sigmund 2003, who pose the problem as one of non-smooth optimization). Gil & Andreu (2001) combined size and geometry optimization, obtaining solutions to small-scale problems by using optimality criteria and conjugate gradient methods in succession. Martínez et al. (2007) proposed a ‘growth’ method, in which geometry optimization was carried out in conjunction with a heuristic ‘node adding’ algorithm, allowing an increasingly complex truss structure to evolve from a relatively simple initial layout. Although not of specific interest in the present study, their ‘growth’ method allowed a limit to be placed on the number of joints in the resulting optimized truss to be controlled, thereby ensuring that the resulting optimized trusses could be rationalized as desired. (Limiting the number of joints was also of specific interest to Prager (1978a) and, more recently, Mazurek et al. (2011), Mazurek (2012).) However, the focus of most work in this field has been on single load case problems, and most of the aforementioned methods cannot easily be extended to treat multiple load cases. An exception is the combined topology/layout and geometry optimization procedure put forward by Aichtziger (2007), which was recently extended by Descamps & Filomeno Coelho (2013) to allow small-scale multiple load case problems to be considered. However, in general, geometry optimization requires the starting layout to quite closely resemble the true optimal solution in order for it to work effectively. Furthermore, the geometry optimization process can be computationally expensive. Here the efficacy of a geometry optimization post-processing step will be explored, which involves starting with a layout optimization solution comprising a reduced number of nodes and bars, and then using a highly efficient interior point method to solve the resulting non-linear optimization problem. This approach is general, and can be applied to a wide variety of problems, including those involving multiple load cases and self-weight.

The format of the paper is as follows: firstly the general layout optimization problem is considered and then revised to incorporate ‘joint lengths’; secondly, the geometry optimization problem is mathematically defined and extensions and implementation issues discussed; finally a number of numerical examples are solved to demonstrate the efficacy of the rationalization methods considered, and conclusions are drawn.

4.2 Rationalization of layout optimization solutions using joint lengths

The first rationalization technique considered is one proposed by Parkes (1975). According to his formulation, a notional joint length, s , is added to the length of each bar. Thus, the computed volume of the truss structure under consideration becomes: $V = \tilde{\mathbf{I}}^T \mathbf{a}$, where V is the total computed volume of the truss structure; $\tilde{\mathbf{I}}$ is a vector containing modified truss bar lengths (i.e.

$\{l_1 + s, l_2 + s, \dots, l_m + s\}$, for a problem involving m bars), and \mathbf{a} is a vector containing the bar cross-sectional areas.

Note that though this can simplify the truss layout, the calculated structural volume will clearly always *increase* because of the inclusion of additional joint lengths. However, after the optimization has been completed, the ‘standard’ volume can be calculated by summing up the volumes of all bars, excluding the joint lengths from this calculation (all volumes reported herein were calculated in this way).

The updated layout optimization problem, now including joint lengths, can therefore be stated as:

$$\min_{\mathbf{a}, \mathbf{q}} V = \tilde{\mathbf{l}}^T \mathbf{a} \quad (4.1a)$$

s.t.

$$\left. \begin{aligned} \mathbf{B}\mathbf{q}^\alpha + \mathbf{W}\mathbf{a} &= \mathbf{f}^\alpha \\ -\sigma^- \mathbf{a} &\leq \mathbf{q}^\alpha \leq \sigma^+ \mathbf{a} \end{aligned} \right\} \text{for all } \alpha \in \mathbb{F} \quad (4.1b)$$

$$\mathbf{a} \geq \mathbf{0}, \quad (4.1c)$$

where \mathbf{W} contains self-weight coefficients, here assuming self-weight to be lumped at the nodes; \mathbf{B} is an equilibrium matrix comprising direction cosines; \mathbf{q} is a vector containing the internal bar forces and \mathbf{f} is a vector containing the external forces. Also σ^+ and σ^- are limiting tensile and compressive stresses respectively, $\mathbb{F} = \{1, 2, \dots, p\}$ is a load case set, where α is the load case identifier and p represents the total number of load cases.

The optimization variables are the cross-sectional areas in \mathbf{a} and the internal forces in \mathbf{q} . It can be observed that the coefficient matrices are determined by the positions of the nodes and the connectivity of the truss bars; therefore the optimization formulation (4.1) is an LP problem.

4.3 Post-processing rationalization using geometry optimization

The second technique considered involves the use of geometry optimization as a post-processing step to rationalize solutions obtained using layout optimization. Initially the geometry optimization process will be considered in isolation; subsequently practical issues related to combining geometry optimization with layout optimization will be considered.

4.3.1 Basic geometry optimization formulation

Initially consider an unbounded 2D design domain, where the x, y positions of the nodes in a truss are considered as optimization variables. (For sake of simplicity, formulae for 3D trusses are not explicitly provided in the paper; however, the relevant formulae can be derived similarly.)

Considering first a problem involving a single load case, without self-weight, gives:

$$\min_{x,y,a,q} V = \mathbf{l}(\mathbf{x}, \mathbf{y})^T \mathbf{a} \quad (4.2a)$$

$$\text{s.t.} \quad \mathbf{B}(\mathbf{x}, \mathbf{y})\mathbf{q} = \mathbf{f} \quad (4.2b)$$

$$-\sigma^- \mathbf{a} \leq \mathbf{q} \leq \sigma^+ \mathbf{a} \quad (4.2c)$$

$$\mathbf{a} \geq \mathbf{0}, \quad (4.2d)$$

where \mathbf{l} is a vector containing the lengths of the truss bars. The optimization variables in this case are x , y , a and q ; it is evident that the objective function (4.2a) and equality constraint (4.2b) are both now non-linear. Without loss of generality, problem (4.2) can be categorized as a non-linear, non-convex optimization problem.

Also, although problem (4.2) can be considered as a combined layout and geometry problem, similar to the approach put forward by Achtziger (2007), and developed further by Descamps & Filomeno Coelho (2013), in this paper geometry optimization is considered as a *separate process*, which is carried out only after an initial layout optimization solution has been performed, and active bars in the optimum truss have been identified. Advantages of this approach stem from the fact that the layout optimization formulation: (i) allows a globally optimal solution to be obtained for a given ground structure, typically very close to the true optimal solution; (ii) can be solved extremely rapidly. Thus the layout optimization solution provides an excellent starting point for a subsequent geometry optimization, which, although capable of rationalizing the structure, is fundamentally non-convex and may be computationally expensive.

Figure 4.1 illustrates the non-convex nature of a simple four-bar truss problem. Suppose that the truss shown in Fig. 4.1 has only one free (movable) node C, whose position can be optimized in the x - y plane. In this case there exists two zones Ω_1 and Ω_2 in which node C can potentially become trapped, leading to different optimum solutions. In fact node C must be positioned in zone Ω_2 , at (1.00, 0.25), in order to obtain the globally optimal solution.

Assuming that a truss layout is available, various methods of improving the solution via geometry optimization techniques are possible, though some methods appear to have inherent limitations. For example, the geometry optimization step in the ‘growth’ method proposed by Martínez et al. (2007) requires that the truss under consideration is statically determinate. With this stipulation, the state variable \mathbf{q} can be eliminated by taking $\mathbf{q} = \mathbf{B}^{-1}\mathbf{f}$, simplifying the underlying optimization problem. However, for problems with multiple load cases, this stipulation cannot be imposed. As both single and multiple load case problems are considered here, a more general approach is required, with statically indeterminate truss structures allowed. To solve the resulting non-linear problem efficiently, first and second derivatives of the objective function and constraints in (4.2) with respect to optimization variables are obtained analytically.

Note that the entire geometry optimization formulation for a truss structure can be assembled using locally derived formulae for each truss bar. Also, the derivatives can be assembled similarly. In the following section local formulae for a single bar are introduced, permitting the problem for the whole structure to be constructed.

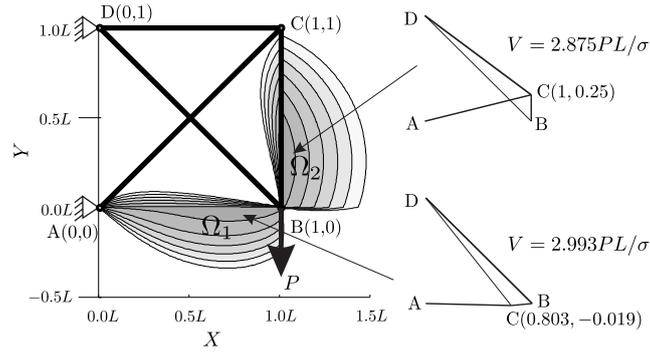


Figure 4.1: Four-bar truss illustrating non-convex nature of geometry optimization. The optimum position of node C is sought; contours show the variation of the structural volume for differing positions of node C.

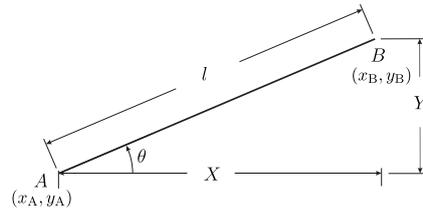


Figure 4.2: Notation used for a truss bar

Mathematical expressions for a single truss bar

For the truss bar connecting nodes $A(x_A, y_A)$ and $B(x_B, y_B)$ shown in Fig. 4.2, let $X = x_B - x_A$ and $Y = y_B - y_A$. The length of this bar is $l = \sqrt{X^2 + Y^2}$ and volume $V^{AB} = la$.

The contribution to the equilibrium matrix of this single bar can be stated as:

$$\mathbf{B}^{AB} = \begin{bmatrix} -\cos \theta & -\sin \theta & \cos \theta & \sin \theta \end{bmatrix}^T. \quad (4.3)$$

Assuming the optimization variables are defined as $[x_A, y_A, x_B, y_B, a, q]$, the gradient of the objective function is written as:

$$\nabla V^{AB} = \begin{bmatrix} -\frac{Xa}{l} & -\frac{Ya}{l} & \frac{Xa}{l} & \frac{Ya}{l} & l & 0 \end{bmatrix}^T. \quad (4.4)$$

The Jacobian matrix of the equality constraint (4.2b) can be derived as:

$$\mathbf{J}_{Bq}^{AB} = \begin{bmatrix} \frac{qY^2}{l^3} & -\frac{qXY}{l^3} & -\frac{qY^2}{l^3} & \frac{qXY}{l^3} & 0 & -\frac{X}{l} \\ -\frac{qXY}{l^3} & \frac{qX^2}{l^3} & \frac{qXY}{l^3} & -\frac{qX^2}{l^3} & 0 & -\frac{Y}{l} \\ -\frac{qY^2}{l^3} & \frac{qXY}{l^3} & \frac{qY^2}{l^3} & -\frac{qXY}{l^3} & 0 & \frac{X}{l} \\ \frac{qXY}{l^3} & -\frac{qX^2}{l^3} & -\frac{qXY}{l^3} & \frac{qX^2}{l^3} & 0 & \frac{Y}{l} \end{bmatrix}. \quad (4.5)$$

The stress inequality constraint (4.2c) is linear; therefore the coefficients directly form the Jacobian

matrix.

For a single truss bar, first derivatives of the objective function and associated constraints can also be obtained. To ensure rapid convergence of the non-linear optimization process, second-order terms are also derived analytically; the Hessian matrix of the objective function, $V^{AB} = la$ can be derived as:

$$\nabla^2 V^{AB} = \begin{bmatrix} \frac{aY^2}{l^3} & -\frac{aXY}{l^3} & -\frac{aY^2}{l^3} & \frac{aXY}{l^3} & -\frac{X}{l} & 0 \\ -\frac{aXY}{l^3} & \frac{aX^2}{l^3} & \frac{aXY}{l^3} & -\frac{aX^2}{l^3} & -\frac{Y}{l} & 0 \\ -\frac{aY^2}{l^3} & \frac{aXY}{l^3} & \frac{aY^2}{l^3} & -\frac{aXY}{l^3} & \frac{X}{l} & 0 \\ \frac{aXY}{l^3} & -\frac{aX^2}{l^3} & -\frac{aXY}{l^3} & \frac{aX^2}{l^3} & \frac{Y}{l} & 0 \\ -\frac{X}{l} & -\frac{Y}{l} & \frac{X}{l} & \frac{Y}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.6)$$

For equality constraint $\mathbf{B}^{AB}q - \mathbf{f}^{AB} = \mathbf{0}$, note that this comprises four constraints: $-q \cos \theta - f_{x_A} = 0$, $-q \sin \theta - f_{y_A} = 0$, $q \cos \theta - f_{x_B} = 0$ and $q \sin \theta - f_{y_B} = 0$, where f_{x_A} , f_{y_A} , f_{x_B} and f_{y_B} are external loads applied at nodes A and B. Also note that the magnitude of external loads are assumed not to change during the optimization process, so that $\nabla^2 f_{x_A} = \nabla^2 f_{y_A} = \nabla^2 f_{x_B} = \nabla^2 f_{y_B} = \mathbf{0}$. The Hessian matrix of each of the constraints can readily be derived. For instance, $\nabla^2(q \cos \theta)$ is shown in (4.7), and the mathematical expression for $\nabla^2(q \sin \theta)$ can be obtained in a similar manner. Also, as the inequality constraint (4.2c) is linear, its second-order derivative term is zero.

$$\nabla^2(q \cos \theta) = \begin{bmatrix} -\frac{3qXY^2}{l^5} & -\frac{qY(-2X^2+Y^2)}{l^5} & \frac{3qXY^2}{l^5} & \frac{qY(-2X^2+Y^2)}{l^5} & 0 & -\frac{Y^2}{l^3} \\ -\frac{qY(-2X^2+Y^2)}{l^5} & \frac{qX(-X^2+2Y^2)}{l^5} & \frac{qY(-2X^2+Y^2)}{l^5} & -\frac{qX(-X^2+2Y^2)}{l^5} & 0 & \frac{XY}{l^3} \\ \frac{3qXY^2}{l^5} & \frac{qY(-2X^2+Y^2)}{l^5} & -\frac{3qXY^2}{l^5} & -\frac{qY(-2X^2+Y^2)}{l^5} & 0 & \frac{Y^2}{l^3} \\ \frac{qY(-2X^2+Y^2)}{l^5} & -\frac{qX(-X^2+2Y^2)}{l^5} & -\frac{qY(-2X^2+Y^2)}{l^5} & \frac{qX(-X^2+2Y^2)}{l^5} & 0 & -\frac{XY}{l^3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{Y}{l^3} & \frac{XY}{l^3} & \frac{Y^2}{l^3} & -\frac{XY}{l^3} & 0 & 0 \end{bmatrix}. \quad (4.7)$$

4.3.2 Geometry optimization formulation with self-weight

The basic formulation can be extended to account for self-weight. For a single truss bar AB, the corresponding self-weight coefficient matrix \mathbf{W}^{AB} is given as:

$$\mathbf{W}^{AB} = \frac{\rho g l}{2} \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T. \quad (4.8)$$

In which ρ and g are respectively the material density and acceleration due to gravity. The Jacobian matrix $\mathbf{J}_{W_a}^{AB}$ can be derived as:

$$\mathbf{J}_{W_a}^{AB} = \frac{\rho g}{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{aX}{l} & -\frac{aY}{l} & \frac{aX}{l} & \frac{aY}{l} & l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{aX}{l} & -\frac{aY}{l} & \frac{aX}{l} & \frac{aY}{l} & l & 0 \end{bmatrix}. \quad (4.9)$$

Also, the Hessian matrix can be obtained by considering only the second and fourth (i.e. non-zero) terms of \mathbf{W}^{AB} . Note that the relevant terms in both cases are:

$$\nabla^2\left(\frac{\rho g}{2}ql\right) = \frac{\rho g}{2} \begin{bmatrix} \frac{qY^2}{l^3} & -\frac{qXY}{l^3} & -\frac{qY^2}{l^3} & \frac{qXY}{l^3} & 0 & -\frac{X}{l} \\ -\frac{qXY}{l^3} & \frac{qX^2}{l^3} & \frac{qXY}{l^3} & -\frac{qX^2}{l^3} & 0 & -\frac{Y}{l} \\ -\frac{qY^2}{l^3} & \frac{qXY}{l^3} & \frac{qY^2}{l^3} & -\frac{qXY}{l^3} & 0 & \frac{X}{l} \\ \frac{qXY}{l^3} & -\frac{qX^2}{l^3} & -\frac{qXY}{l^3} & \frac{qX^2}{l^3} & 0 & \frac{Y}{l} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{X}{l} & -\frac{Y}{l} & \frac{X}{l} & \frac{Y}{l} & 0 & 0 \end{bmatrix}. \quad (4.10)$$

With respect to the geometry optimization problem (4.2), analytical expressions for the first and second derivatives have been derived. Thus simple problems (e.g. the problem in Fig. 4.1) can now be optimized without difficulty (though without any certainty of obtaining the global optimum). However, when dealing with structures involving large numbers of nodes, various practical issues might prevent the process obtaining a satisfactory solution; these issues are considered in the next section.

4.3.3 Practical issues

A number of practical issues which must be considered in order to develop a robust and flexible geometry optimization procedure are now considered.

Node move limits

It has been shown that geometry optimization will in general lead to a non-convex mathematical optimization problem, which can cause issues when applying convex optimization methods. To try to avoid such issues it is convenient to impose upper and lower limits on nodal positions x and y . However, it is worth pointing out that imposing such limits will mean that only locally optimal solutions will be found. Considering the evolving nature of the geometry of the structure during the optimization process, rules can be applied which ensure that the structure always remains similar in form to the initial structure. Hence the starting point, or initial condition, for the problem is crucial as it directly determines which local optimum zone the solution lies in. For instance, considering the structure shown in Fig. 4.1, as node C lies on the edge of zone Ω_2 , it is likely that imposed move limits will eliminate the possibility of this node being moved to zone Ω_1 . However, whether node C is restricted to lie within zone Ω_1 or Ω_2 depends upon the initial position of C, and upon the imposed move limits.

To describe node move limits concisely, coordinates of a given node in a 2D truss are written in column vector form: $\boldsymbol{\nu} = [x, y, 1]^T$ in \mathbb{R}^3 . (Note that although nodal positions lie in \mathbb{R}^2 , the redundant '1' in $\boldsymbol{\nu}$ is used solely to condense the mathematical expression.)

Now consider the node move limits. Suppose that each node is allowed to move within a circular zone, determined according to the distance from a given node to adjacent nodes. Figure 4.3 shows adjacent nodes A and B, which are originally located at $\boldsymbol{\nu}_A^0$ and $\boldsymbol{\nu}_B^0$ respectively. Two circular zones Ω_A and Ω_B , with radius $r_{AB} = \frac{1}{2} \|\boldsymbol{\nu}_B^0 - \boldsymbol{\nu}_A^0\|_2$, are defined to restrict nodal movements. Let $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_B$ represent the positions of node A and B respectively. When $\boldsymbol{\nu}_A = \boldsymbol{\nu}_B$, a zero length bar may be implied. To prevent this occurring, a gap of length ϵ is created between zones, such that the restriction for node A becomes: $\|\boldsymbol{\nu}_A - \boldsymbol{\nu}_A^0\|_2^2 \leq (r_{AB} - \epsilon)^2$. This is an extra constraint compared with those in the standard formulation (4.2). Its Jacobian matrix \mathbf{J}^A and Hessian matrix \mathbf{H}^A can be obtained as:

$$\mathbf{J}^A = 2 [x_A - x_A^0, y_A - y_A^0], \quad (4.11)$$

$$\mathbf{H}^A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \quad (4.12)$$

Although the restriction shown in Fig. 4.3 is normally sufficient to assure the non-linear, non-convex, optimization process is stable, in some cases additional restrictions need to be imposed. Thus, a program parameter r_s is introduced which defines the maximum node move limit for all nodes; in this case the above restriction is modified to:

$$\|\boldsymbol{\nu}_A - \boldsymbol{\nu}_A^0\|_2^2 \leq (r^*)^2, \quad (4.13)$$

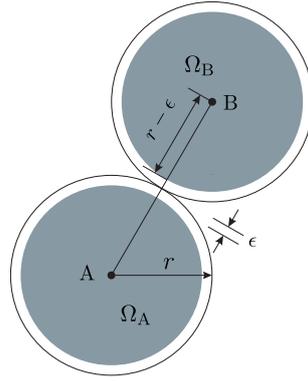


Figure 4.3: Node move limit zone: shaded circular zones indicate node move limits

where $r^* = \min\{r_s, r_{AB}\} - \epsilon$ is the modified node move limit. In this paper r_s is taken as the x - or y -distance between the nodes used in the original layout optimization process. When the non-linear optimization fails to converge rapidly, this parameter can be reduced with a view to stabilizing the non-linear problem. Also, from a computational point of view it is useful to impose relatively tight bounds on the variables x_A and y_A representing movements of a given node A, for simplicity applying these limits in the Cartesian directions:

$$\begin{aligned} x_A^0 - r^* &\leq x_A \leq x_A^0 + r^*, \\ y_A^0 - r^* &\leq y_A \leq y_A^0 + r^*. \end{aligned} \quad (4.14)$$

However, restricting nodal movements means that the final solution will normally not be obtained in a single step, and an iterative solution scheme is therefore required. In this scheme all nodes are moved to optimum positions within the prescribed move limit zones; these zones are then updated based on the new nodal positions. The optimization process proceeds iteratively, until all nodes are stationary (to within a specified tolerance).

Note that the aforementioned constraints are defined using the nodal distances between adjacent nodes. Therefore when a node is quite close to another, each node is restricted from moving a significant distance. This might affect convergence speed, especially when particular nodes lie in an extremely small region, with a radius r not significantly larger than ϵ . As a consequence these nodes can become effectively locked, and cannot be moved further.

Additionally, various design limitations may need to be taken into account. The first is the line constraint, which restricts certain nodes (e.g. nodes on supported boundaries) to move only along given line paths. The second of these is the design domain constraint, which restricts all nodes to lie within the specified design domain. It is only necessary to apply this constraint to nodes which have the potential to move outside the domain (this can conveniently be determined by taking account of the move limit for each node). For sake of simplicity, polygonal design domains and straight line supports are considered in this paper, so that only linear constraints need to be formulated for these two types of design constraint.

A line in \mathbb{R}^2 can be written as: $T^x x + T^y y + T^c = 0$, where T^x , T^y and T^c are coefficients of the line; its vector form is then written as: $\mathbf{T}\boldsymbol{\nu} = 0$, in which, $\mathbf{T} = [T^x, T^y, T^c]$. Thus the line constraint for a given node A can be written as:

$$\mathbf{T}_A^L \boldsymbol{\nu}_A = 0, \quad (4.15)$$

where \mathbf{T}_A^L is the coefficient vector for the line node A is prescribed to lie on. Also, the domain constraint can be written as:

$$\mathbf{T}_A^D \boldsymbol{\nu}_A \geq \mathbf{0}, \quad (4.16)$$

where \mathbf{T}_A^D contains coefficients of all domain boundary lines close to node A (each row in \mathbf{T}_A^D comprises coefficients of a single boundary line):

$$\mathbf{T}_A^D = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} T_1^x & T_1^y & T_1^c \\ T_2^x & T_2^y & T_2^c \\ \vdots & \vdots & \vdots \end{bmatrix}. \quad (4.17)$$

Note that for a domain boundary line, its normal direction (i.e. the sign of \mathbf{T}) matters as it determines which side of the line is ‘inward’ facing.

Modified formulation

Consider a truss comprising $\mathbb{N} = \{1, 2, \dots, n\}$ nodes, with subsets of nodes \mathbb{N}^L and \mathbb{N}^D denoting nodes lying on lines or close to domain boundaries respectively. The full optimization problem, taking account of node move limits and self-weight, can now be written as:

$$\min_{x, y, a, q^\alpha} V = \mathbf{l}^T \mathbf{a} \quad (4.18a)$$

s.t.

$$\left. \begin{array}{l} \mathbf{B}\mathbf{q}^\alpha + \mathbf{W}\mathbf{a} = \mathbf{f}^\alpha \\ \sigma^- \mathbf{a} \leq \mathbf{q}^\alpha \leq \sigma^+ \mathbf{a} \end{array} \right\} \text{for all } \alpha \in \mathbb{F} \quad (4.18b)$$

$$\|\boldsymbol{\nu}_j - \boldsymbol{\nu}_j^0\|_2^2 \leq (r^*)^2 \text{ for all } j \in \mathbb{N} \quad (4.18c)$$

$$\mathbf{T}_{j_L}^L \boldsymbol{\nu}_{j_L} = 0 \text{ for all } j_L \in \mathbb{N}^L \quad (4.18d)$$

$$\mathbf{T}_{j_D}^D \boldsymbol{\nu}_{j_D} \geq \mathbf{0} \text{ for all } j_D \in \mathbb{N}^D \quad (4.18e)$$

$$\mathbf{a} \geq \mathbf{0} \quad (4.18f)$$

$$\mathbf{x}^{\text{lb}} \leq \mathbf{x} \leq \mathbf{x}^{\text{ub}} \quad (4.18g)$$

$$\mathbf{y}^{\text{lb}} \leq \mathbf{y} \leq \mathbf{y}^{\text{ub}}. \quad (4.18h)$$

The new constraints (4.18d) and (4.18e) are linear, so coefficient matrices \mathbf{T}^D and \mathbf{T}^L directly form the Jacobian matrices. (The Hessian matrices are zero matrices in this case.)

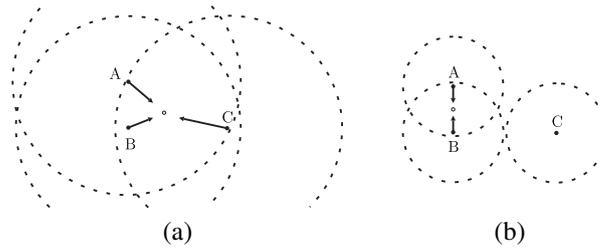


Figure 4.4: Merging a group of nodes: (a) a large merge radius results in a group containing the three nodes, A, B and C, which can then be merged into a single node; (b) a small merge radius results in a group consisting of nodes A and B, which can then be merged, whilst node C remains as-is.

Merging nodes

During the geometry optimization process some nodes may migrate towards one another (this phenomenon was also observed by Achtziger (2007), who addressed this by adding the possibility for nodes to ‘melt’ (i.e. merge together) in his proposed procedure). In this paper, it can be observed that the gap ϵ included in constraint (4.13) will prevent nodes from taking up the same position, and hence merging. Therefore an approach is needed to merge nodes into a concentrated node; here this involves two major steps.

In the first step, nodes to be merged are identified and grouped, based on a program parameter, the node merge radius r_M . A node merge group contains candidate nodes to be merged. For a given node, adjacent nodes lying within radius r_M are added to the same group; an example is shown in Fig. 4.4. When r_M is greater than the distance between nodes A and C (Fig. 4.4a), a single group containing all three nodes is created, and then merged to a single node. When r_M is greater than the distance between nodes A and B, but is less than the distance between A and C (Fig. 4.4b), one group is created, and the two nodes in this group are then merged.

In the second step, all nodes in a given node merge group are merged to the centroid of the nodes in the group. Due to its heuristic nature, the validity of this process needs to be numerically validated; the merging process is deemed to be successful if the resulting structure has the same computed volume as before (within a prescribed error tolerance).

All steps in process to merge nodes are listed below:

Node merge algorithm

1. Select an initial prescribed node merge radius, r_M .
 2. Create node merge groups.
 3. For every group, check whether a valid merge can be undertaken.
 4. If a valid merge can be carried out for all groups go to 6, else 5.
 5. If invalid group can be split, reduce r_M and go to 2, else 6.
 6. End of node merge process.
-

Considering crossovers

In a truss layout derived from layout optimization, bars will very often intersect / crossover one another. However, crossover points do not normally coincide with nodes. A crossover creation process can be carried out to create nodes at these points, thereby splitting the intersecting bars.

With these newly created nodes, there is scope to further reduce structural volume. However, creating new nodes also leads to a growth in problem size, which becomes significant in the first few iterations, when a large number of crossover points are typically observed. To avoid significantly increasing problem size, the crossover creation process is therefore not carried out initially. This is achieved by using inner and outer loops in the main procedure as follows: (i) inner loop: the optimization is progressed without creating crossover nodes; this loop terminates when a prescribed termination criterion has been met; (ii) outer loop: this carries out the process of creating crossover nodes when the inner loop ends. The outer loop terminates when no more crossover nodes need to be created, also terminating the entire optimization procedure.

This approach is based on the assumption that, whenever an inner loop terminates, the form of an optimized layout has been identified, and the number of crossover points has been significantly reduced. Note that when considering 3D structures, bars are less likely to intersect one another, since for this to occur both bars must lie on the same plane. However, often a bar in a 3D structure can pass very close to one or more other bars. This indicates that crossovers should be identified approximately, using a tolerance which is progressively increased from zero to a prescribed value.

Extracting nodes and bars from the layout optimization solution

A viable structural layout, obtained using layout optimization, is the starting point of the geometry optimization-based rationalization process described here. However, ensuring a viable layout is obtained requires various steps to be taken, as described in this section.

Conventionally, when using an interior point-based linear programming solver, an optimum truss layout is ‘extracted’ by identifying bars which have an area above a given filter threshold. Though this typically provides a qualitatively reasonable layout, it can mean that one or more small but structurally important bars are filtered out. To ensure this does not happen, the ‘extracted’ structure can be used as the basis of a new layout optimization, and the volume compared with that obtained originally; if these are not within a prescribed tolerance then the filter threshold should be reduced and the process repeated until a viable layout is obtained.

Finally, chains of in-line bars should be merged into single bars to avoid intermediate nodes from moving freely along their axis without improving the solution (though this is not required in cases when intermediate nodes are either loaded or supported, or when self-weight is being considered).

Dealing with structures which are in unstable equilibrium with the applied loads

Layout optimization may identify structures which are in unstable equilibrium with the applied loads. When dealing with such structures in the geometry optimization rationalization technique, it will normally be observed that the calculated structural volume is very sensitive to the position of certain nodes. This can cause numerical issues in the non-linear optimization solution process. To address this, virtual supports are added and connected with all unsupported nodes by connections which incorporate large joint length penalties to ensure they are not present in the final optimized

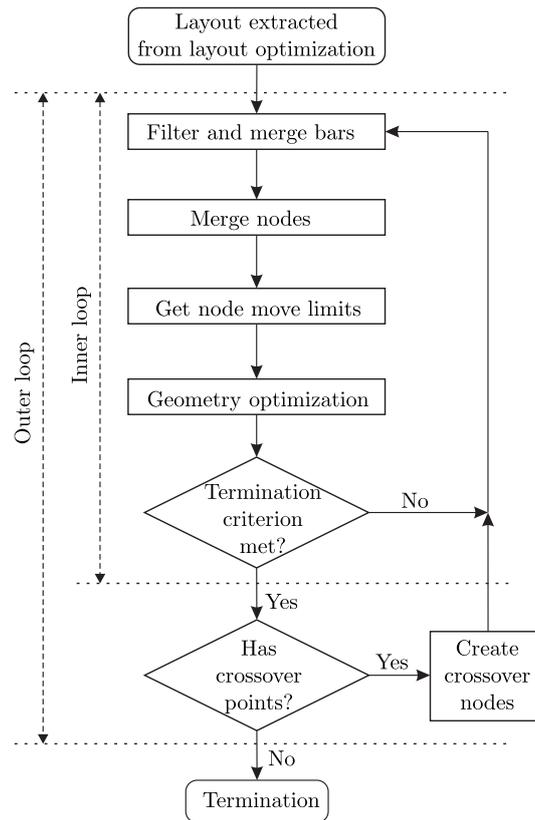


Figure 4.5: Flow chart of the ‘two phase’ geometry optimization procedure

structure. (In the case of 2D trusses two virtual pinned supports^[ii] are required to ensure that every node is adequately constrained, whilst in the case of 3D trusses three virtual supports are required.)

4.3.4 Overall procedure

The overall procedure is shown in Fig. 4.5. As indicated in the figure, initially the geometry optimization steps are performed within an inner loop, starting with the layout derived from layout optimization. Within this loop the form of the structure will gradually change, due to moving and merging of nodes; crossover points, if present, are completely ignored in this loop. The maximum movement of any node is used as the termination criterion (taken as 1×10^{-4} in this paper). Thereafter, crossover points are considered in the outer loop. The process then continues as indicated until no crossover points are identified, with the entire optimization process then terminating.

It is worth pointing out that, when merging nodes, the coordinates of the new merged nodes will be obtained approximately. As a consequence the calculated volume may in some cases be very slightly higher than in the previous step.

^[ii]Before starting an optimization, each virtual support will be connected to every unsupported node using temporary bars. Due to large joint length penalties, these extra bars are forced to have negligible area. After optimization, they will be removed.

4.4 Numerical examples

The efficacy of the two rationalization techniques considered in this paper, i.e. (i) introduction of ‘joint lengths’ and, (ii) application of geometry optimization, are now demonstrated through application to a range of example problems. Unless stated otherwise, a reference length L is used to define the size of a given problem, a load P is applied, and the limiting material stresses are taken as: $\sigma^+ = \sigma^- = \sigma$. Also, in cases where advantage is taken of symmetry (or anti-symmetry), the volume quoted is that of the full structure. With respect to the optimization solvers employed, all LP layout optimization problems were solved using MOSEK (2011) and the non-linear geometry optimization problems solved using IPOPT 3.11.0 (Vigerske & Wachter 2013), with default settings except for the maximum iteration number which was set to 500. All calculations were carried out using MATLAB2013a running on an Intel i5-2310 powered desktop PC with 6G RAM, and running Windows 7 (64bit).

For many of the problems considered a known analytical solution is available. In these cases the errors in the numerical solutions can be quantified, and are denoted ξ_L , ξ_J and ξ_G for the percentage errors of the layout optimization, joint length and geometry optimization rationalized solutions respectively. Also, ξ_M denotes the percentage error in the solution obtained using the software described by Martínez et al. (2007). Also, as the geometry optimization procedure will generally improve on the layout optimization solution, it is also useful to quantify this improvement, here denoted $\eta = \frac{(\xi_L - \xi_G)}{\xi_L} \times 100\%$.

4.4.1 Hemp cantilever

The first example is a cantilever truss considered by Hemp (1974). The problem involves application of a point load at mid-height between two pinned supports, as illustrated in Fig. 4.6(a). (Note that only half of the domain needs to be considered if anti-symmetry is taken into account.) Hemp (1974) quoted the analytical volume to be $4.34PL/\sigma$, but Lewiński (2005) repeated the calculations using greater precision to obtain a more accurate solution, $4.32168PL/\sigma$.

A sample layout optimization solution and corresponding rationalized solutions are also shown in Fig. 4.6. It is evident that both rationalization techniques allow simplified solutions to be obtained. However, whereas the volume associated with the solution obtained using joint length rationalization is 1.49% above the exact value, the solution obtained using geometry optimization rationalization is only 0.23% above the exact value, a significant improvement on the original layout optimization error of 0.75% ($\eta = 69\%$ in this case).

Factors affecting the joint length rationalization technique

With the joint length rationalization technique, adding an additional length s to the real length of each bar has the effect of modifying the solution by effectively penalizing short bars. For the Hemp cantilever shown in Fig. 4.6(a), the influence of the value of s on the layout and corresponding volume is illustrated in Fig. 4.7. It is evident that the volume tends to increase as the joint length is increased, and also that the form of the solution is generally simpler when an increased joint

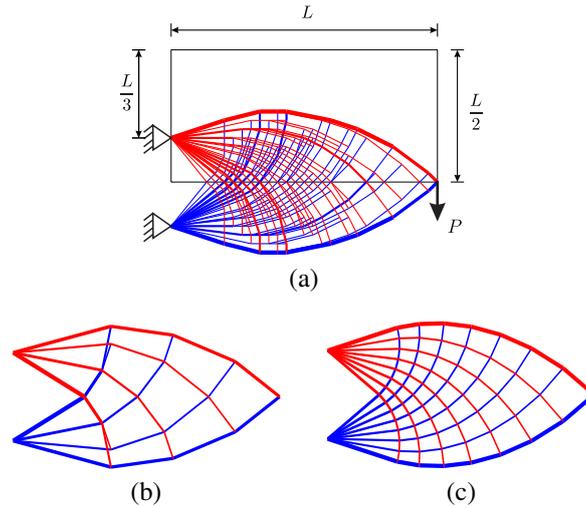


Figure 4.6: Hemp cantilever: (a) problem definition and layout optimization solution obtained using 30×15 nodal divisions, $V = 4.3541PL/\sigma$ ($\xi_L = 0.75\%$); (b) rationalized solution obtained using joint length $s = 0.006L$, $V = 4.3863PL/\sigma$ ($\xi_J = 1.49\%$); (c) rationalized solution obtained using geometry optimization, $V = 4.3318PL/\sigma$ ($\xi_G = 0.23\%$).

length is used. Note that the CPU times were similar for all joint length cases considered.

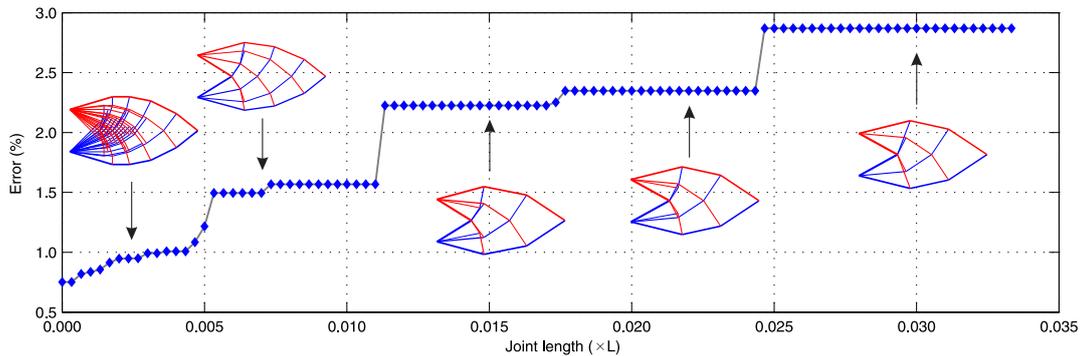


Figure 4.7: Hemp cantilever: influence of joint length on numerical solution and layout (using 30×15 nodal divisions)

Factors affecting the geometry optimization rationalization technique

The geometry optimization rationalization technique is affected by several factors, two of which are now considered: (i) influence of starting structural layout; (ii) influence of node merge radius.

(i) Influence of starting structural layout. Geometry optimization is here viewed as a post-processing technique and a better starting layout, obtained using a finer numerical discretization, will naturally be likely to result in an improved solution, at least in terms of volume. Fig. 4.8 shows the starting layout (obtained using a layout optimization involving 150×75 nodal divisions) and the corresponding rationalized solution obtained using geometry optimization, demonstrating that this rationalization technique can be applied to relatively large-scale problems. However,

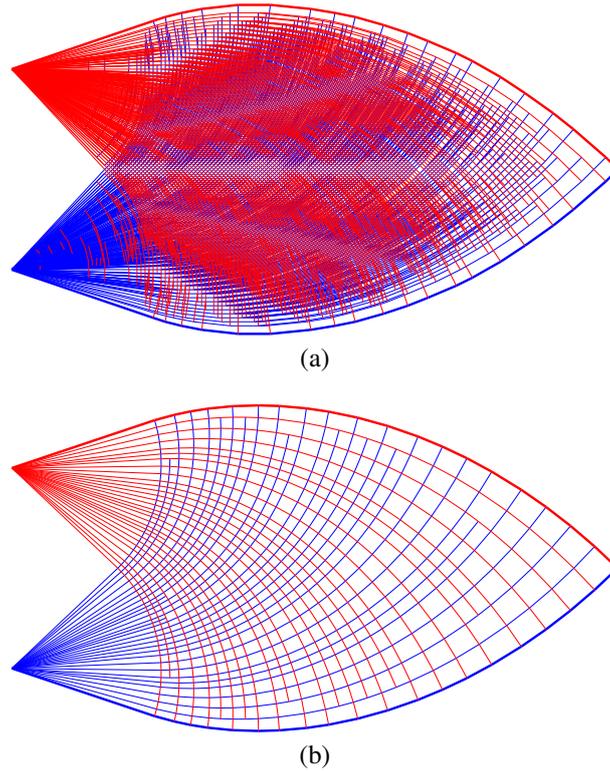


Figure 4.8: Hemp cantilever: (a) layout optimization solution obtained using 150×75 nodal divisions, $V = 4.3258PL/\sigma$ ($\xi_L = 0.09\%$) ; (b) rationalized solution obtained using geometry optimization, $V = 4.3228PL/\sigma$ ($\xi_G = 0.03\%$)

Table 4.1: Hemp cantilever: solution and non-linear optimization CPU cost for varying layout optimization nodal densities

Layout optimization				Geometry optimization rationalization				
Nodal divs	No. of bars	Volume (PL/σ)	Error ξ_L (%)	No. of nodes	No. of bars	Volume (PL/σ)	Error ξ_G (%)	CPU time [†] (sec.)
30×15	74655	4.3541	1.26	92	163	4.3318	0.23	5
60×30	892702	4.3350	0.31	324	605	4.3258	0.09	58
90×45	3149297	4.3296	0.18	774	1480	4.3235	0.04	358
120×60	7004968	4.3274	0.13	1302	2519	4.3232	0.03	1279
150×75	12456601	4.3258	0.09	2192	4244	4.3228	0.03	4875

[†]Total CPU time expended on non-linear optimization, as reported by the solver.

as indicated on Table 4.1, the computational cost associated with the non-linear optimizations employed in the geometry optimization process does increase markedly with problem size (number of nodes). Also, it is evident that the structure shown in Fig. 4.8(b) is still quite complex compared with that shown in Fig. 4.6(c), suggesting that more practically useful solutions will often be obtained when using coarse nodal discretizations.

(ii) Influence of node merge radius. Using a smaller node merge radius r_M can be expected to allow more detail from the original layout optimization solution to be retained, implying also that a better quality solution can be expected to be obtained in this case. However, disabling the merging of nodes altogether can lead to problems (for example some nodes can become effectively

Table 4.2: Hemp cantilever: solution and non-linear optimization CPU cost for varying nodal merging radii (30×15 nodal divisions)

Merge radius*	No. of nodes in resulting structure	No. of bars in resulting structure	Volume (PL/σ)	Error ξ_G (%)	CPU time [†] (sec.)
0.50	37	64	4.3318	0.23	5
0.25	57	102	4.3304	0.20	15
0.10	108	199	4.3283	0.15	76
0	274	489	4.3295	0.18	203

*Expressed as a multiplier of the layout optimization nodal spacing.

[†]Total CPU time expended on non-linear optimization, as reported by the solver.

locked in position when the node move limits are applied). Table 4.2 shows the influence of the choice of node merge radius for the Hemp cantilever problem shown in Fig. 4.6(a). It is clear that the choice of node merge radius has a significant influence on the CPU time, and also does affect the solution slightly (and, for the reason outlined previously, the use of a zero node radius does not lead to the best solution). Thus in this paper a merge radius r_M which equals half the x - or y -distance between the nodes used in the original layout optimization process is pragmatically utilized unless specified otherwise.

Finally, in Fig. 4.9 the progress of the entire iterative solution procedure is shown for the Hemp cantilever example shown in Fig. 4.6(a). The optimization process stays in the inner loop (see Fig. 4.5) until the end of the 7th iteration. Crossover nodes are then created and the inner loop is entered for a second time. The layout of the structure evolves further, until the termination criterion is met.

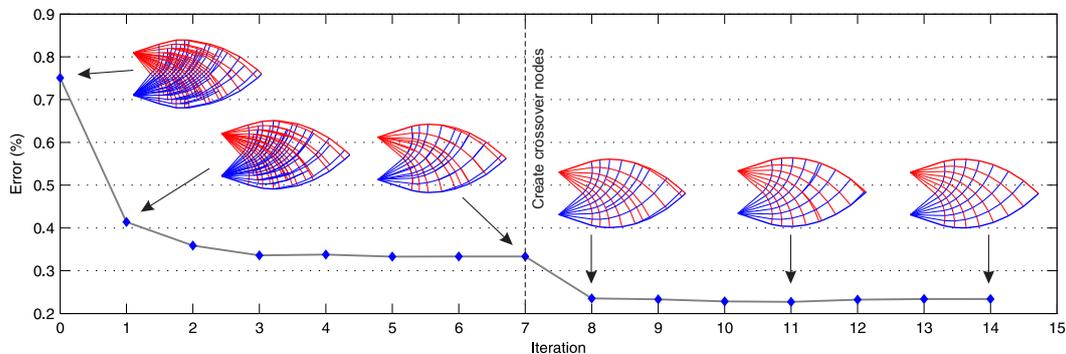


Figure 4.9: Hemp cantilever: geometry optimization solutions obtained during the iterative solution procedure

4.4.2 Centrally loaded Michell beam

The problem shown in Fig. 4.10 is similar to the problem originally considered by Michell (1904), though here the inclination of the midspan point load is allowed to vary. For comparative purposes numerical solutions obtained using the ‘growth’ method described by Martínez et al. (2007) are also provided (using software downloaded using the link given in the paper).

Numerical solutions are shown in Table 4.3. Note that in order to ensure that the geometry optimization rationalization technique produced forms which were anti-symmetric about the line of

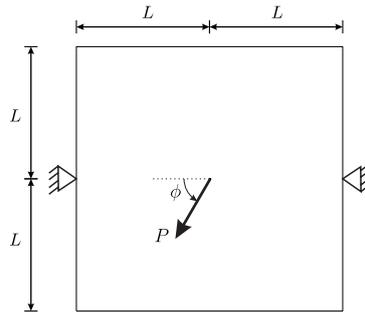


Figure 4.10: Centrally loaded Michell beam: problem definition

load application, similar to those obtained when using layout optimization, it was necessary to prescribe that the horizontal reaction forces at the two pinned supports were equal in magnitude; this was achieved by replacing one of the supported degrees of freedom with an equivalent reaction force, of magnitude $\frac{P \cos(\phi)}{2}$. (However, this approach did not allow sensible solutions to be obtained using the method proposed by Martínez et al. (2007), because Martínez's method appears to 'grow' either the top or the bottom part of the structure, but not both simultaneously.) Also, to avoid nodes being merged in the geometry optimization phase in the vicinity of the singularities at the supports and load position, the node merge radius r_M used was taken as half the standard value (being a quarter of the x - or y -distance between the nodes used in the original layout optimization process).

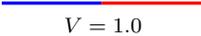
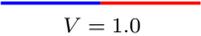
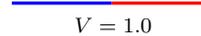
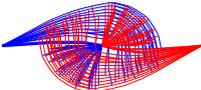
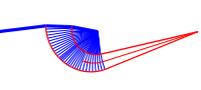
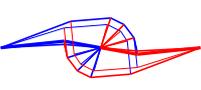
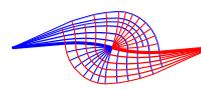
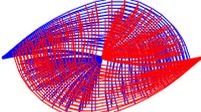
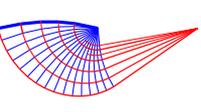
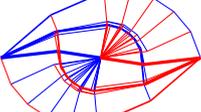
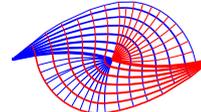
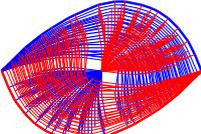
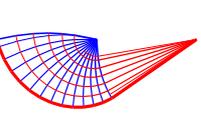
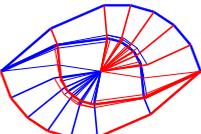
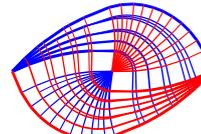
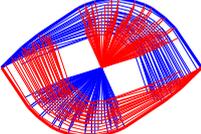
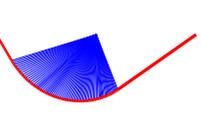
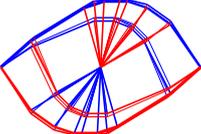
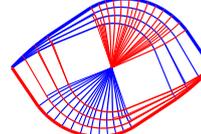
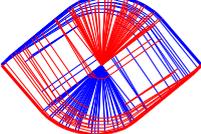
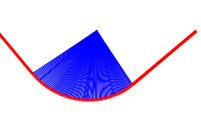
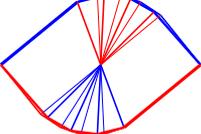
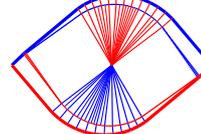
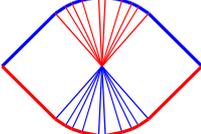
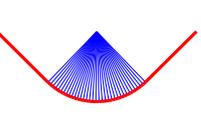
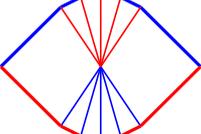
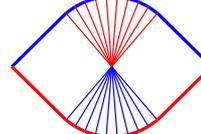
It is apparent from Table 4.3 that for this problem the geometry optimization rationalization technique provides the best all-round solutions, successfully simplifying the standard layout optimization layouts. Also, although the 'growth' method proposed by Martínez et al. (2007) produces the most accurate solution for the $\phi = 90^\circ$ case, in most other cases it fails to capture important detail present in the (near-)optimal layouts, leading to less accurate solutions and to higher computed volumes.

4.4.3 Hemp arch with distributed load

Details of the arch problem investigated by Hemp (1974) are provided in Fig. 4.11(a). Hemp proposed an analytical solution but found that this was in fact non-optimal. However, what is likely to be a very close estimate of the volume of the exact layout ($V = 3.15163 \frac{wL^2}{\sigma}$) was recently put forward by Pichugin et al. (2012). This was obtained using the 'Type III' uniformly distributed loading pattern proposed by Darwich et al. (2010), which is also used here. Additionally, due to the sensitivity of the computed volume to the position of particular nodes, virtual supports are utilized in the geometry optimization rationalization technique. Also, to avoid nodes being merged in the geometry optimization phase in the vicinity of the singularity at the support, the node merge radius r_M used was half the standard value (being a quarter of the x - or y -distance between the nodes used in the original layout optimization process).

Considering the layouts shown in Fig. 4.11, it is clear that only the geometry optimization rationalization technique is capable of simplifying the layout whilst maintaining key features of the original form. The geometry optimization rationalization step also reduces the error from 0.51%

Table 4.3: Centrally loaded Michell beam: volumes ($\times PL/\sigma$) and layouts obtained using various methods vs. inclined load angle ϕ . Minimum volume shown in boldface.

Load angle ϕ	Layout optimization with 60×60 nodal divisions	'Growth' method by Martínez et al. (2007)	Rationalized layout optimization solutions Using joint length: $s = 0.01L$	Using geometry optimization
0°	 $V = 1.0$	 $V = 1.0$	 $V = 1.0$	 $V = 1.0$
10°	 $V = 1.3395$	 $V = 1.3448$	 $V = 1.3478$	 $V = \mathbf{1.3333}$
22.5°	 $V = 1.7450$	 $V = 1.7667$	 $V = 1.7640$	 $V = \mathbf{1.7373}$
45°	 $V = 2.2765$	 $V = 2.3101$	 $V = 2.3001$	 $V = \mathbf{2.2691}$
67.5°	 $V = 2.5263$	 $V = 2.5443$	 $V = 2.5439$	 $V = \mathbf{2.5202}$
80°	 $V = 2.5703$	 $V = 2.5662$	 $V = 2.5860$	 $V = \mathbf{2.5657}$
90°	 $V = 2.5771$	 $V = \mathbf{2.5711}$	 $V = 2.5856$	 $V = 2.5740$

in the original layout optimization solution to 0.10% ($\eta = 81\%$).

4.4.4 Hemp cantilever with self-weight

The Hemp cantilever shown in Fig. 4.6(a) is revisited, though now taking account of self-weight, with $\rho \times g = 1.5\sigma/L$. The solutions are shown in Fig. 4.12.

Although a relatively large joint length has been used in an attempt to derive a suitably simplified structure, it is evident that the resulting layout is significantly more complex than the equivalent layout obtained using the geometry optimization technique.

4.4.5 Chan cantilever with two load cases

The problem shown in Fig. 4.13(a) is a variation on the cantilever truss considered by Chan (1962), though now involving two load cases (and two forces, P and Q , which are each active in only one of the load cases). For the case when $P = Q$, the exact solution can be calculated using superposition principles (e.g. see Nagtegaal & Prager 1973, Spillers & Lev 1971): in this case the ‘sum’ problem clearly gives a volume of $0.5PL/\sigma$; and the ‘difference’ problem takes the form of a ‘Michell’ truss (Lewiński et al. 1994a), whose volume is given by Graczykowski & Lewiński (2010) as $4.729085649PL/\sigma$. Therefore the exact solution can be calculated to be $(4.729085649 + 0.5)PL/\sigma$.

It can be observed from Fig. 4.13 that both rationalization techniques described here successfully simplify the layout, with the geometry optimization rationalization technique also reducing the error in the computed volume, from 0.30% to 0.10% (error reduction $\eta = 66.7\%$).

4.4.6 Flower truss with two load cases

To further demonstrate the capability of the rationalization techniques, another problem involving two load cases will be considered; details of the problem are shown in Fig. 4.14(a). The analytical solution for this problem can again be derived using superposition principles. Thus with given dimension $R = 0.5L$, the optimal volume can be calculated to be: $V = (46.052 + 10.000)PL/\sigma$ (refer to Fig. 4.15 for further details).

Due to the relatively coarse nodal discretization employed in this case, comparatively little rationalization of the initial layout optimization solution is required. However, the geometry optimization rationalization clearly simplifies the layout and also reduces the error ($\eta = 80\%$) in this case. (Also note that for this problem ξ_L and ξ_G are both relatively high, partly because the circular support is modelled with only 18 nodes and, in this paper, these are non-movable in the geometry optimization phase. i.e. a curved nodal movement path is beyond the scope of the present paper).

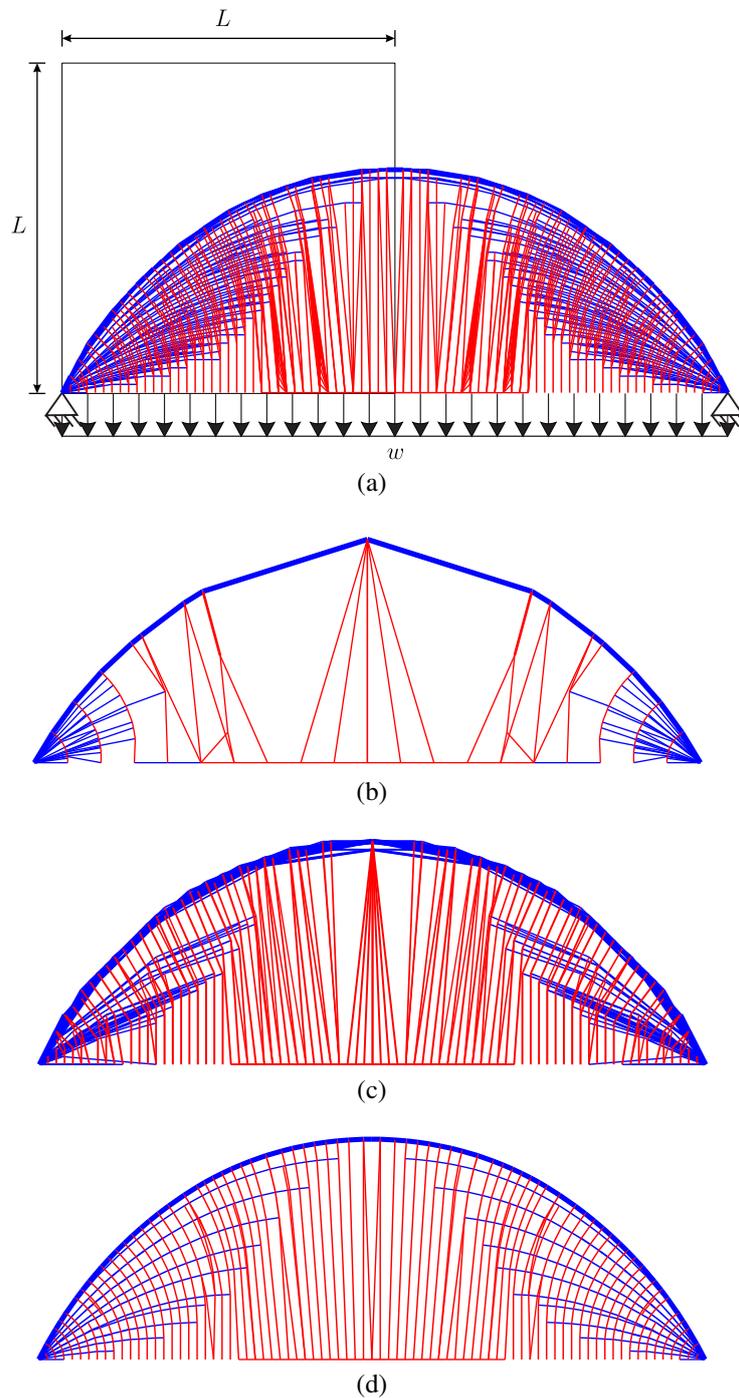


Figure 4.11: Hemp arch with distributed load: (a) problem definition and layout optimization solution obtained using 40×40 nodal divisions, $V = 3.1679wL^2/\sigma$ ($\xi_L = 0.51\%$); (b) method by Martínez et al. (2007), using 20 nodal divs as software failed to yield reasonable results when 40 nodal divs were employed, $V = 3.2736wL^2/\sigma$ ($\xi_M = 3.86\%$); (c) rationalized solution obtained using joint length $s = 0.01L$, $V = 3.2044wL^2/\sigma$ ($\xi_J = 1.66\%$); (d) rationalized solution obtained using geometry optimization, $V = 3.1550wL^2/\sigma$ ($\xi_G = 0.10\%$)

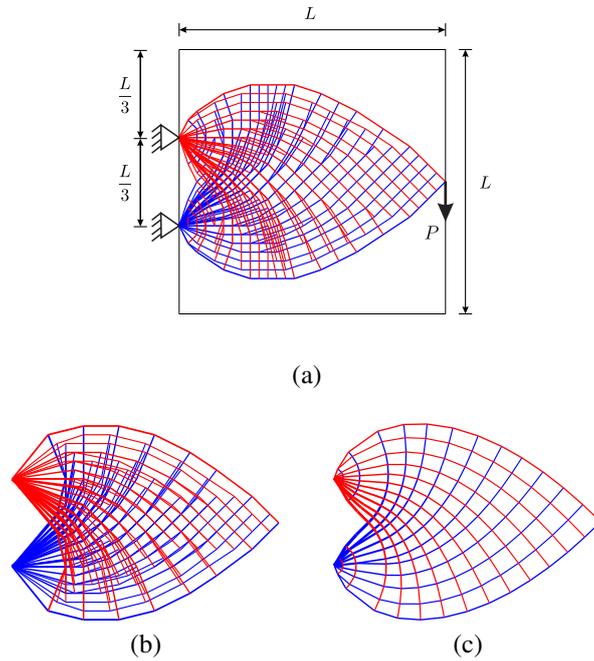


Figure 4.12: Hemp cantilever with self-weight: (a) problem definition and layout optimization solution using 30×30 nodal divisions, $V = 35.894PL/\sigma$; (b) rationalized solution obtained using joint length $s = 0.06L$, $V = 38.150PL/\sigma$; (c) rationalized solution obtained using geometry optimization, $V = 34.608PL/\sigma$

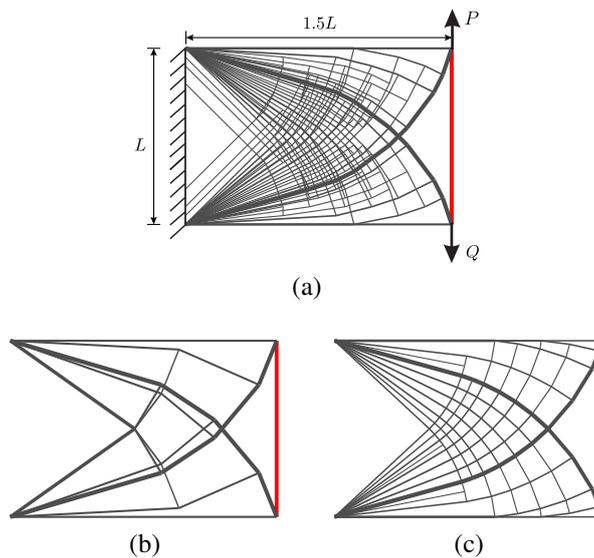


Figure 4.13: Chan cantilever with two load cases: (a) problem definition ($Q = P$) and layout optimization solution obtained using 30×20 nodal divisions, $V = 5.2450PL/\sigma$ ($\xi_L = 0.30\%$); (b) rationalized solution obtained using joint length $s = 0.015L$, $V = 5.2712PL/\sigma$ ($\xi_J = 0.80\%$); (c) rationalized solution obtained using geometry optimization, $V = 5.2344PL/\sigma$ ($\xi_G = 0.10\%$)

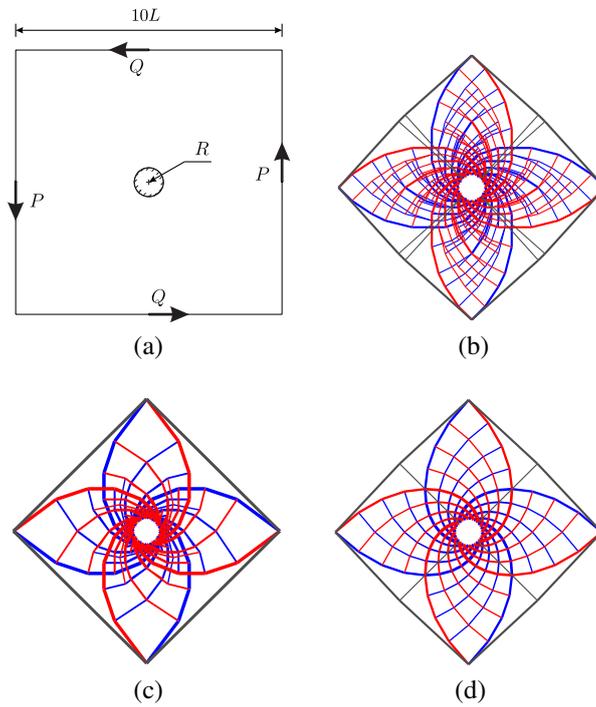


Figure 4.14: Flower truss with two load cases: (a) problem definition ($P = Q$), circular support modelled using 18 nodes; (b) layout optimization solution obtained using 50×50 nodal divisions, $V = 57.387PL/\sigma$ ($\xi_L = 2.38\%$); (c) rationalized solution obtained using using joint length $s = 0.05L$, $V = 57.801PL/\sigma$ ($\xi_J = 3.12\%$); (d) rationalized solution obtained using geometry optimization, $V = 56.324PL/\sigma$ ($\xi_G = 0.49\%$)

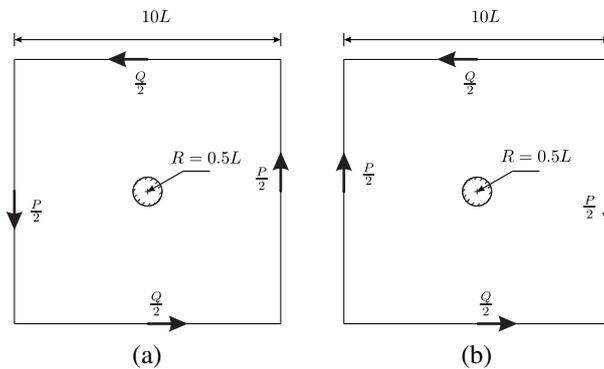


Figure 4.15: Flower truss with two load cases: equivalent single load case problems using superposition principle (a) ‘sum’ problem $V = \frac{1}{2} \times 5 \log\left(\frac{5}{0.5}\right) \times 2 \times 4PL/\sigma = 46.052PL/\sigma$ (Michell 1904); (b) ‘difference’ problem $V = \frac{1}{2} \sin^2\left(\frac{\pi}{4}\right) \times 5 \times 4PL/\sigma = 10.000PL/\sigma$

4.4.7 Michell sphere

The Michell sphere is the minimum volume 3D structure to support a pair of axial torques (Michell 1904). Though the exact solution to this problem has been derived theoretically (e.g. Michell 1904, Hemp 1973, Lewiński 2004), existing numerical solutions are not satisfactory. For example in Czarnecki (2003) the difference between the quoted computed and exact volumes was found to be 40.6% (Lewiński 2004). Here, using anti-symmetric boundary conditions, the problem can be modelled using a reduced domain; in this case one eighth of a cube was used, as shown in Fig. 4.16(a). The torque on one side is modelled by applying point loads to 20 circumferentially positioned nodes in the full problem (i.e. to $20/4 + 1 = 6$ nodes in the reduced problem). The analytical solution is $V = \frac{4T}{\sigma} \log \cot \frac{\phi}{2}$ (after Hemp 1973, Lewiński 2004). For the given dimensions ($R = 50L$, $\phi = 18^\circ$ and $T = 100PL$), the exact volume is therefore $737.09PL/\sigma$.

The results of the geometry optimization rationalization technique are shown in Fig. 4.16(c), (d). It is clear that the rationalization technique does an excellent job of simplifying the complex initial layout optimization solution shown in Fig. 4.16(b), also reducing the error in the volume in this case from 4.24% to 0.43% (error reduction $\eta = 90\%$).

4.5 Conclusions

Numerical layout optimization provides an efficient means of identifying (near-)optimal truss topologies for a variety of problem types. However, the solutions obtained are often complex in form, and effective means of rationalizing the output are often needed. In this paper two rationalization techniques are explored:

- Rationalization by including joint lengths in the layout optimization problem is computationally efficient since it simply requires minor modification of the underlying linear programming (LP) problem. The solutions obtained are often simplified effectively, according to the joint length utilized. However, the solutions are normally *less* efficient (i.e. have a higher structural volume) than solutions obtained using the standard layout optimization procedure. Also, in some cases this method fails to simplify the truss topology effectively.
- Rationalization by performing geometry optimization is a post-processing step which involves the solution of a non-linear optimization problem. This approach has been found to be effective in simplifying the solution obtained via layout optimization for a wide variety of problem types, including those involving distributed loads, self-weight, multiple load-cases and 3D geometries. Starting with a layout optimization solution, which typically comprises relatively few bars, means that the subsequent geometry optimization phase is relatively computationally inexpensive (cf. the integrated layout and geometry optimization strategies proposed by others). Also, the solutions are normally *more* efficient (i.e. have a lower structural volume) than the original layout optimization solutions. However, the non-linear, non-convex, nature of the geometry optimization formulation means that there can be no guarantee as to the proximity of the solution obtained to the global optimum; thus its use primarily as a rationalization technique, as proposed in this paper, appears appropriate.

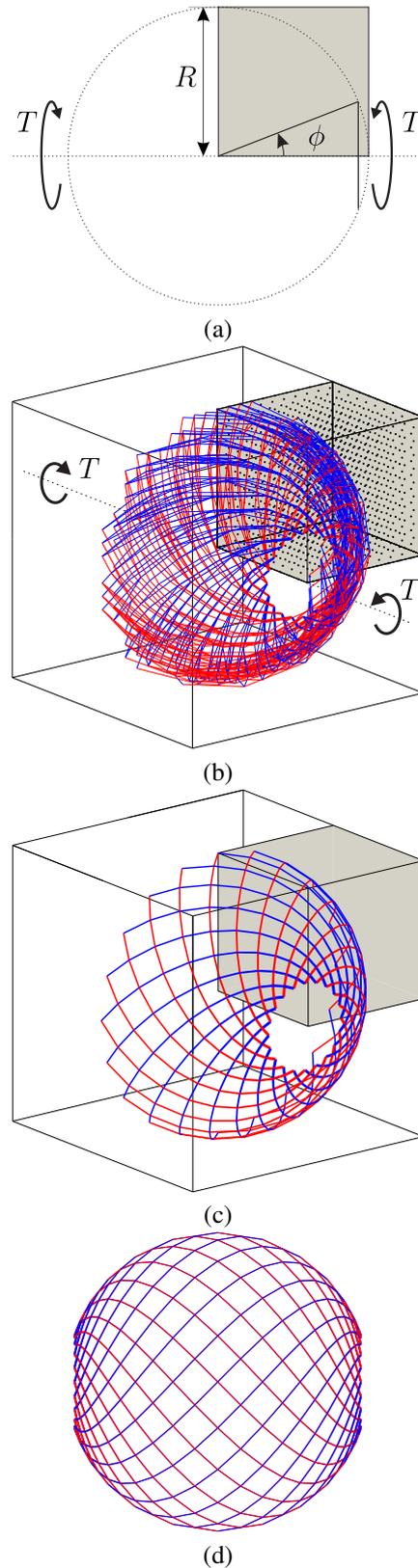


Figure 4.16: Michell sphere: (a) problem definition ($R = 50L, T = 100PL, \phi = 18^\circ$), torsional load modelled using 20 nodes in the full problem; (b) layout optimization solution obtained using $10 \times 10 \times 10$ nodal divisions, $V = 768.34PL/\sigma$ ($\xi_L = 4.24\%$) (showing half of the full structure); (c) rationalized solution obtained using geometry optimization, $V = 740.26PL/\sigma$ ($\xi_G = 0.43\%$) (showing half of the full structure); (d) alternative view of solution shown in (c) (showing full structure)

Chapter 5

Automatic rationalization of yield-line patterns identified using discontinuity layout optimization^[i]

Preface

Another application of layout optimization of interest in this thesis is yield-line analysis of reinforced concrete slabs, undertaken using discontinuity layout optimization (DLO). The method takes into account the analogy between truss layouts and yield-line patterns, leading to a truly systematic automated yield-line analysis procedure, something which had been lacking for more than half a century since the method was invented. While details of this novel method were published in a journal paper co-authored by the author of this thesis (Gilbert et al. 2014, also available in Appendix A), this chapter focuses on enhancing the basic method. Motivated by the outcomes of the rationalization process applied in truss layout optimization, this chapter explores the potential of utilising a similar technique to rationalize the yield-line patterns identified using DLO.

^[i]The content of this chapter was originally prepared for a journal paper: He, L., Gilbert, M. (2015), 'Automatic rationalization of yield-line patterns identified using discontinuity layout optimization', *International Journal of Solids and Structures* (submitted for publication).

Abstract The well-known yield-line analysis procedure for slabs has recently been systematically automated, enabling the critical yield-line pattern to be identified quickly and easily, whatever the slab geometry. This has been achieved by using the discontinuity layout optimization (DLO) procedure, which involves using optimization to identify the critical layout of yield-line discontinuities interconnecting regularly spaced nodes distributed across a slab. However, whilst highly accurate solutions can be obtained, the corresponding yield-line patterns are often quite complex in form, especially when relatively dense nodal grids are employed. Here a method of rationalizing the DLO-derived yield-line patterns via a geometry optimization post-processing step is described. Geometry optimization involves adjusting the positions of the nodes, thereby simultaneously simplifying and improving the accuracy of the solution. The mathematical expressions involved are derived analytically, and various practical issues are highlighted and addressed. Finally, an interior point optimizer is used to obtain rationalized solutions for a variety of sample slab analysis problems, clearly demonstrating the efficacy of the proposed rationalization technique.

Keywords Discontinuity layout optimization, yield-line analysis, geometry optimization

5.1 Introduction

The yield-line method of analysis proposed by Johansen (1943) provides a powerful means of computing the collapse load factor of a reinforced concrete slab. The method requires a kinematically admissible failure mechanism to be prescribed, defined by means of a yield-line pattern. The early focus was on slabs with relatively simple geometries (e.g., Johansen 1943, 1968) because, at the time, systematic means of identifying the critical failure mechanism for irregularly shaped slabs were not available. Subsequently Chan (1972) and Munro & Da Fonseca (1978) proposed a means of automatically identifying the critical yield-line pattern. This involved discretizing a slab using rigid finite-elements, with the critical yield-line pattern then obtained automatically via linear optimization. However, because yield-lines were restricted to forming only at the edges of the finite-elements, the resulting yield-line patterns were significantly influenced by the initial mesh topology. Attempting to address this issue, various workers proposed the use of ‘geometry optimization’ to subsequently adjust the positions of selected nodes in a post-processing phase. For example, Johnson (1994, 1995) proposed that this be achieved via the use of sequential linear programming. Other workers to propose a similar approach included Thavalingam et al. (1999), who employed a conjugate gradient optimizer, and Ramsay & Johnson (1997, 1998), who used a direct search solver. However, as indicated by Ramsay et al. (2015), the outcomes will be affected by the initial mesh topology, and a poor initial solution will render any subsequent geometry optimization phase largely ineffective. Another issue is the need to manually identify yield-lines from the finite-element meshes; any misinterpretation can reduce the efficacy of the geometry optimization phase. This has been described as being ‘difficult’ (e.g., Johnson 1994, Thavalingam et al. 1999).

More recently, Jackson (2010) and Jackson & Middleton (2013) used a lower-bound finite element solution to derive ‘yield-line indicators’, which could be used to infer the likely general form of the critical yield line pattern. This then enabled a more refined yield-line pattern to be identified

via a geometry optimization step. The resulting procedure allowed reasonable yield-line analysis solutions to be obtained for complex slab problems. However, as the procedure involved a manual interpretation step, a truly systematic means of automatically identifying the critical yield-line pattern remained to be found.

Recently, this goal was achieved by Gilbert et al. (2014), who used discontinuity layout optimization (DLO) to automate the process of identifying the most critical yield-line pattern. Instead of discretizing the problem using elements arranged in a finite element mesh, the slab area is populated by nodes, and these are then interconnected with a large set of potential yield-lines, which are free to cross-over one another. A highly efficient optimization process is then used to find the critical subset of yield-lines involved in the critical failure mechanism. An overview of the steps involved in the DLO procedure is shown in Fig. 5.1. However, whilst highly accurate solutions can be obtained using the DLO procedure, the corresponding yield-line patterns are often quite complex in form, especially when relatively dense nodal grids are employed. In an attempt to address this, a modified formulation was also proposed by Gilbert et al. (2014). The modified formulation involved penalizing short yield-lines, leading to solutions that were generally simpler in form than the original. However, these solutions were also less accurate (i.e. the gap between the exact and DLO solution was increased). In the present paper a geometry optimization step will instead be used to rationalize the yield-line patterns, with a view to simultaneously simplifying the yield-line patterns and improving the solutions (i.e. so that the gap between the exact and DLO solution reduces).

The proposed procedure clearly has similarities with the procedure put forward by Johnson (1994, 1995), which also involved the use of a geometry optimization step. However, in the proposed procedure the rationalization process starts from a yield-line pattern obtained using DLO, which is a much better starting point than a yield-line pattern derived from a rigid finite element analysis. Also, here the relevant geometry optimization formulae will be derived analytically, thus permitting a wider variety of optimization methods to be applied. These distinguishing features can be expected to ensure that performance is much improved. Note also that the proposed procedure is similar to the procedure recently proposed for rationalizing trusses identified using layout optimization (He & Gilbert 2015a); also the use of a geometry optimization step to improve very coarse resolution DLO solutions has recently been proposed for in-plane analysis problems by Bauer & Lackner (2015).

The paper is organized as follows: (i) the new DLO-based automated yield-line analysis procedure is first introduced; (ii) the geometry optimization problem is defined and relevant mathematical expressions are given; (iii) implementation issues are considered and addressed; (iv) various numerical examples are used to demonstrate the efficacy of the procedure; (v) conclusions from the study are presented.

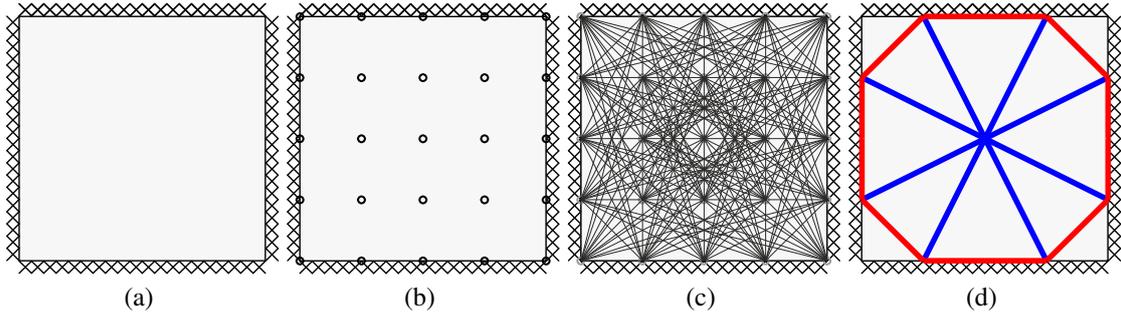


Figure 5.1: Steps in the DLO procedure: (a) define slab geometry and properties; (b) discretize slab using nodes; (c) interconnect nodes with potential yield-lines; (d) use optimization to identify optimal subset of yield-lines, and resulting yield-line pattern

5.2 Automated yield-line analysis using DLO

5.2.1 Overall problem formulation

The kinematic DLO limit analysis formulation for a weightless slab can be written as an optimization problem as follows (after Gilbert et al. 2014):

$$\min_{\mathbf{d}, \mathbf{p}} \lambda \mathbf{f}_L^T \mathbf{d} = \mathbf{g}^T \mathbf{p} \quad (5.1a)$$

$$\text{s.t. } \mathbf{B} \mathbf{d} = \mathbf{0} \quad (5.1b)$$

$$\mathbf{N} \mathbf{p} - \mathbf{d} = \mathbf{0} \quad (5.1c)$$

$$\mathbf{f}_L^T \mathbf{d} = 1 \quad (5.1d)$$

$$\mathbf{p} \geq \mathbf{0}, \quad (5.1e)$$

where the objective is to minimise the internal work done along yield-lines (5.1a), subject to compatibility at nodes (5.1b), plastic flow requirements (5.1c), a unit displacement constraint, defined according to the principle of virtual work, (5.1d), and a constraint that ensures that the internal work done must be positive (5.1e). And where λ is a dimensionless load factor, and \mathbf{p} and \mathbf{g} are vectors containing plastic multipliers and their corresponding work equation coefficients. Also \mathbf{B} is a suitable compatibility matrix containing direction cosines for the yield-lines, and \mathbf{d} contains relative displacements along yield-lines, as shown in Fig. 5.2 (where θ_n , θ_t , and δ are respectively the normal rotation, twisting rotation, and out-of-plane displacement, along a yield-line or at the edge of a slab). Also, \mathbf{N} is a suitable plastic flow matrix and \mathbf{f}_L is a vector that prescribes the effect of live loads ‘above’ each yield-line.

The optimization variables are the yield-line displacements in \mathbf{d} and plastic multipliers in \mathbf{p} . Since all terms are linear, the optimization formulation (5.1) can be solved using linear programming (LP). The entire optimization problem can be assembled using locally derived formulae for each yield-line, which are introduced in the following section.

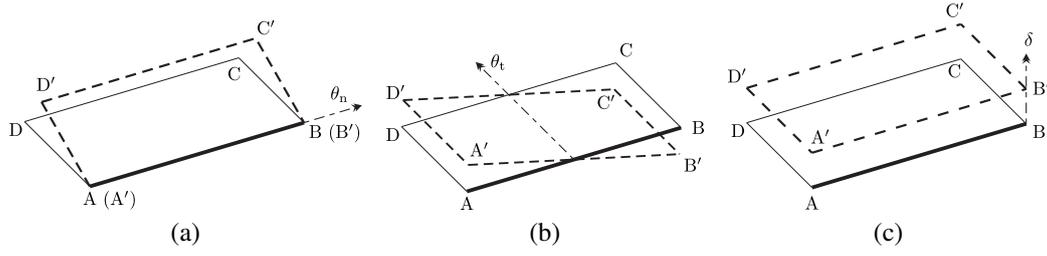


Figure 5.2: Relative displacements at yield-line AB (assuming slab area ABCD moves to A'B'C'D'): (a) normal rotation along yield-line; (b) twisting rotation; (c) out-of-plane translation

5.2.2 Terms for a single yield-line

For a yield-line i that connects two nodes $A(x_A, y_A)$ and $B(x_B, y_B)$, as shown in Fig. 5.3, let $x_l = x_B - x_A$ and $y_l = y_B - y_A$. (Note that in the interests of conciseness, the subscript i has been omitted, i.e. x_l is used rather than x_{li} ; this is repeated for all coefficients defined in this section). Clearly, the length of this yield-line $l = \sqrt{x_l^2 + y_l^2}$. Now assume that the displacement variables in \mathbf{d} for this yield-line are of the form $[\theta_n, \theta_t, \delta]^T$. The contribution to the nodal compatibility constraint (5.1b) for this yield line is given by:

$$\mathbf{B}_i \mathbf{d}_i = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & \frac{l}{2} & 1 \\ -\cos \phi & \sin \phi & 0 \\ -\sin \phi & -\cos \phi & 0 \\ 0 & \frac{l}{2} & -1 \end{bmatrix} \begin{bmatrix} \theta_n \\ \theta_t \\ \delta \end{bmatrix}. \quad (5.2)$$

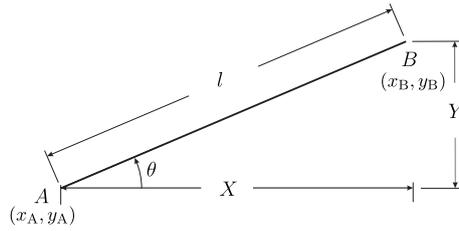
However, in yield-line analysis θ_t and δ will be zero except at free edges and along symmetry planes; also internal work will only be associated with normal rotation θ_n . Hence the plastic flow rule constraint for the yield-line will simply be:

$$\mathbf{N}_i \mathbf{p}_i - \mathbf{d}_i = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} p^+ \\ p^- \end{bmatrix} - [\theta_n] = 0, \quad (5.3)$$

where p^+, p^- are plastic multiplier variables, constrained to take only positive values. Assuming that the slab is isotropically reinforced, and m_p^+ and m_p^- denote the sagging and hogging moment capacity per unit length respectively, the contribution to the objective function (5.1a) for this yield-line can be written as:

$$\mathbf{g}_i^T \mathbf{p}_i = \begin{bmatrix} m_p^+ l & m_p^- l \end{bmatrix} \begin{bmatrix} p^+ \\ p^- \end{bmatrix} = l(m_p^+ p^+ + m_p^- p^-). \quad (5.4)$$

The external work done by live loads is calculated by considering the effect of loads on a strip lying 'above' the yield-line under consideration (Fig. 5.4a). The geometric parameters of the strip are defined by this yield-line and the shape of the top edge of the slab. In the present paper it is


 Figure 5.3: Notation used for a yield-line i connecting points A and B

necessary to define mathematical expressions for \mathbf{f}_{Li} . First, as shown in Fig. 5.4, global $(\vec{X}, \vec{Y}, \vec{Z})$ and local $(\vec{\xi}, \vec{\eta}, \vec{Z})$ Cartesian coordinate systems are defined for yield-line AB. The effects of loads acting on the strip can be prescribed via the local coordinate system: rotational moment along yield-line AB in the $\vec{\xi}$ direction, torsional moment in the $\vec{\eta}$ direction, and shear force in the out-of-plane direction (\vec{Z} direction).

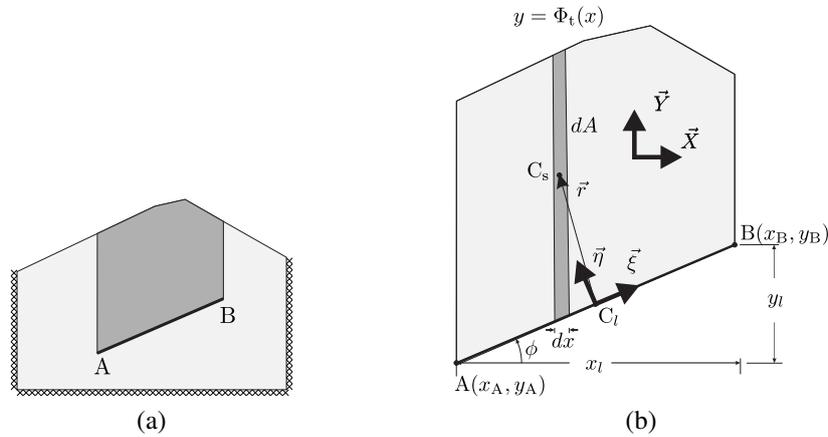


Figure 5.4: Computing the effect of loads 'above' yield-line AB

Consider a uniformly distributed pressure load of intensity q . Now consider an infinitely narrow vertical strip of thickness dx located at horizontal distance x . The area of this strip can be written as $dA = (\Phi_t(x) - \Phi_l(x))dx$, where the shape of the top and bottom edges of the strip are defined by $y = \Phi_t(x)$ and $y = \Phi_l(x)$ respectively. The magnitude of the pressure load on the whole strip can now be written as:

$$\mathbf{f}_i = \begin{bmatrix} 0, & 0, & - \int_{x_A}^{x_B} q dA \end{bmatrix}^T. \quad (5.5)$$

To determine the moment caused by the external load it is necessary to calculate the distance vector \vec{r} from the mid-point of line AB to the centroid of the load, where $\vec{r} : \mathbf{r} = [x_s - x_c, y_s - y_c, 0]^T$, and where the centroid of the infinitely thin strip is located at (x_s, y_s) , and the mid-point of AB is located at (x_c, y_c) . Thus the moment caused by load on the whole strip above AB will be:

$$\mathbf{m}_i = \begin{bmatrix} \int_{AB} q (y_c - y_s) dA, & \int_{AB} q (x_s - x_c) dA, & 0 \end{bmatrix}^T. \quad (5.6)$$

By combining (5.5) and (5.6), the effects of the live load can thus be written as:

$$\mathbf{f}_{Li}^G = q \begin{bmatrix} \int_{AB} (y_c - y_s) dA \\ \int_{AB} (x_s - x_c) dA \\ - \int_{AB} dA \end{bmatrix} = q \begin{bmatrix} \int_{x_A}^{x_B} \Lambda_x(x) dx \\ \int_{x_A}^{x_B} \Lambda_y(x) dx \\ - \int_{x_A}^{x_B} \Lambda_z(x) dx \end{bmatrix}, \quad (5.7)$$

where,

$$\Lambda_x(x) = (\Phi_t(x) - \Phi_l(x))y_c - \frac{\Phi_t^2(x) - \Phi_l^2(x)}{2}, \quad (5.8a)$$

$$\Lambda_y(x) = (x - x_c)(\Phi_t(x) - \Phi_l(x)), \quad (5.8b)$$

$$\Lambda_z(x) = \Phi_t(x) - \Phi_l(x). \quad (5.8c)$$

Λ_x , Λ_y , and Λ_z are *unit-length moment* and *unit-length area* functions with respect to x in the global coordinate system, that respectively describe the first moment of area on \vec{X} , the first moment of the area on \vec{Y} , and the area per unit length in direction \vec{X} for the strip ‘above’ the yield-line. In addition, let $\Gamma_x = \int_{x_A}^{x_B} \Lambda_x dx$, $\Gamma_y = \int_{x_A}^{x_B} \Lambda_y dx$, and $\Gamma_z = - \int_{x_A}^{x_B} \Lambda_z dx$ represent the *unit live load effect*. Note that the yield-line displacements are defined in a local coordinate system, and it is thus necessary to apply a coordinate transformation to obtain the requisite values:

$$\mathbf{f}_{Li} = q \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Gamma_x \\ \Gamma_y \\ \Gamma_z \end{bmatrix}. \quad (5.9)$$

5.3 Geometry optimization: basic formulation

In geometry optimization, in addition to the original variables (the displacements d in \mathbf{d} and plastic multipliers p in \mathbf{p}), nodal positions x, y are also considered as optimization variables. Also, with respect to the original optimization formulation, the objective function (5.1a), nodal compatibility constraint (5.1b), and unit displacement constraint (5.1d) now become non-linear, thus leading to a non-linear programming (NLP) problem. To solve this problem efficiently, the first and second derivatives of the objective function and constraints can be derived analytically, and efficient non-linear optimization packages such as IPOPT (Vigerske & Wachter 2013) can be utilized. In the following section, mathematical expressions for the geometry optimization problem are given, including the first derivatives with respect to the optimization variables (i.e., x, y, d and p); second derivatives are provided in Appendix C.

5.3.1 First derivative terms

The gradient of the objective function and Jacobian matrices of the constraints are the first derivatives required to solve the NLP problem. Assuming that the optimization variables are in the form $[x_A, y_A, x_B, y_B, \theta_n, \theta_t, \delta, p^+, p^-]$ then the gradient of the objective function (5.4) can be obtained as:

$$\nabla\lambda = \left[-\frac{\lambda x_l}{l^2}, -\frac{\lambda y_l}{l^2}, \frac{\lambda x_l}{l^2}, \frac{\lambda y_l}{l^2}, 0, 0, 0, m_p^+ l, m_p^- l \right]^T. \quad (5.10)$$

Now consider the nodal compatibility constraint. As twisting rotation and out-of-plane displacement will be zero for yield-lines which do not lie on free (or symmetry) boundaries, it is efficient to treat these differently; thus compatibility matrix \mathbf{B}_i can conveniently be divided into two parts, $\mathbf{B}_i = \mathbf{B}_i^I + \mathbf{B}_i^{II}$, where:

$$\mathbf{B}_i^I = \begin{bmatrix} \cos \phi & 0 & 0 \\ \sin \phi & 0 & 0 \\ 0 & 0 & 0 \\ -\cos \phi & 0 & 0 \\ -\sin \phi & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_i^{II} = \begin{bmatrix} 0 & -\sin \phi & 0 \\ 0 & \cos \phi & 0 \\ 0 & \frac{l}{2} & 1 \\ 0 & \sin \phi & 0 \\ 0 & -\cos \phi & 0 \\ 0 & \frac{l}{2} & -1 \end{bmatrix}. \quad (5.11)$$

The Jacobian matrices for these two parts, $\mathbf{B}_i^I \mathbf{d}_i$ and $\mathbf{B}_i^{II} \mathbf{d}_i$, can be calculated separately:

$$\mathbf{J}_{B_i^I d_i} = \begin{bmatrix} -\frac{\theta_n y^2}{l^3} & \frac{\theta_n x y}{l^3} & \frac{\theta_n y^2}{l^3} & -\frac{\theta_n x y}{l^3} & \frac{x}{l} & 0 & 0 & 0 & 0 \\ \frac{\theta_n x y}{l^3} & -\frac{\theta_n x^2}{l^3} & -\frac{\theta_n x y}{l^3} & \frac{\theta_n x^2}{l^3} & \frac{y}{l} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\theta_n y^2}{l^3} & -\frac{\theta_n x y}{l^3} & -\frac{\theta_n y^2}{l^3} & \frac{\theta_n x y}{l^3} & -\frac{x}{l} & 0 & 0 & 0 & 0 \\ -\frac{\theta_n x y}{l^3} & \frac{\theta_n x^2}{l^3} & \frac{\theta_n x y}{l^3} & -\frac{\theta_n x^2}{l^3} & -\frac{y}{l} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5.12a)$$

$$\mathbf{J}_{B_i^{II} d_i} = \begin{bmatrix} -\frac{\theta_t x y}{l^3} & \frac{\theta_t x^2}{l^3} & \frac{\theta_t x y}{l^3} & -\frac{\theta_t x^2}{l^3} & 0 & -\frac{y}{l} & 0 & 0 & 0 \\ -\frac{\theta_t y^2}{l^3} & \frac{\theta_t x y}{l^3} & \frac{\theta_t y^2}{l^3} & -\frac{\theta_t x y}{l^3} & 0 & \frac{x}{l} & 0 & 0 & 0 \\ -\frac{\theta_t x}{2l} & -\frac{\theta_t y}{2l} & \frac{\theta_t x}{2l} & \frac{\theta_t y}{2l} & 0 & \frac{l}{2} & 1 & 0 & 0 \\ \frac{\theta_t x y}{l^3} & -\frac{\theta_t x^2}{l^3} & -\frac{\theta_t x y}{l^3} & \frac{\theta_t x^2}{l^3} & 0 & \frac{y}{l} & 0 & 0 & 0 \\ \frac{\theta_t y^2}{l^3} & -\frac{\theta_t x y}{l^3} & -\frac{\theta_t y^2}{l^3} & \frac{\theta_t x y}{l^3} & 0 & -\frac{x}{l} & 0 & 0 & 0 \\ -\frac{\theta_t x}{2l} & -\frac{\theta_t y}{2l} & \frac{\theta_t x}{2l} & \frac{\theta_t y}{2l} & 0 & \frac{l}{2} & -1 & 0 & 0 \end{bmatrix}. \quad (5.12b)$$

Note that except for yield-lines lying on a free (or symmetry) edge, only \mathbf{B}_i^I is required. The Jacobian matrix of the flow rule constraint (5.3) can be derived as:

$$\mathbf{J}_{N_i d_i - p_i} = [0, 0, 0, 0, -1, 0, 0, 1, -1]. \quad (5.13)$$

For the live load effect constraint (5.1d), the Jacobian matrix can be written as:

$$\mathbf{J}_{f_{L_i}^T d_i - 1} = \left[\frac{\partial f_{L_i}^T}{\partial x_A} \mathbf{d}_i, \frac{\partial f_{L_i}^T}{\partial y_A} \mathbf{d}_i, \frac{\partial f_{L_i}^T}{\partial x_B} \mathbf{d}_i, \frac{\partial f_{L_i}^T}{\partial y_B} \mathbf{d}_i, \mathbf{f}_{L_i}^T \frac{\partial \mathbf{d}_i}{\partial \theta_n}, \mathbf{f}_{L_i}^T \frac{\partial \mathbf{d}_i}{\partial \theta_t}, \mathbf{f}_{L_i}^T \frac{\partial \mathbf{d}_i}{\partial \delta}, 0, 0 \right]. \quad (5.14)$$

Now consider partial derivatives of the unit live load effects (i.e., Γ_x , Γ_y , and Γ_z) in the global coordinate system. Partial derivatives of Γ_α ($\alpha = x, y, z$) can now be written as:

$$\frac{\partial \Gamma_\alpha}{\partial x_A} = \frac{\partial}{\partial x_A} \int_{x_A}^{x_B} \Lambda_\alpha dx = -\Lambda_\alpha + \int_{x_A}^{x_B} \frac{\partial \Lambda_\alpha}{\partial x_A} dx, \quad (5.15a)$$

$$\frac{\partial \Gamma_\alpha}{\partial y_A} = \frac{\partial}{\partial y_A} \int_{x_A}^{x_B} \Lambda_\alpha dx = \int_{x_A}^{x_B} \frac{\partial \Lambda_\alpha}{\partial y_A} dx, \quad (5.15b)$$

$$\frac{\partial \Gamma_\alpha}{\partial x_B} = \frac{\partial}{\partial x_B} \int_{x_A}^{x_B} \Lambda_\alpha dx = \Lambda_\alpha + \int_{x_A}^{x_B} \frac{\partial \Lambda_\alpha}{\partial x_B} dx, \quad (5.15c)$$

$$\frac{\partial \Gamma_\alpha}{\partial y_B} = \frac{\partial}{\partial y_B} \int_{x_A}^{x_B} \Lambda_\alpha dx = \int_{x_A}^{x_B} \frac{\partial \Lambda_\alpha}{\partial y_B} dx. \quad (5.15d)$$

Next consider the local coordinate system. Note that in (5.14), the partial derivatives with respect to the nodal coordinates (i.e., the first four terms) have very similar expressions, and those with respect to yield-line displacements (i.e., the fifth to seventh terms) are similar. In the interests of conciseness, only the first and fifth terms (i.e., $\frac{\partial f_{L_i}^T}{\partial x_A} \mathbf{d}_i$ and $\mathbf{f}_{L_i}^T \frac{\partial \mathbf{d}_i}{\partial \theta_n}$) are shown:

$$\begin{aligned} \frac{\partial f_{L_i}^T}{\partial x_A} \mathbf{d}_i = & q\delta \frac{\partial}{\partial x_A} \Gamma_z - q\theta_t \left(\frac{\Gamma_y}{l} - \frac{x_l}{l} \frac{\partial}{\partial x_A} \Gamma_y + \frac{y_l}{l} \frac{\partial}{\partial x_A} \Gamma_x - \frac{x_l^2}{l^3} \Gamma_y + \frac{y_l x_l}{l^3} \Gamma_x \right) \\ & - q\theta_n \left(\frac{\Gamma_x}{l} - \frac{x_l}{l} \frac{\partial}{\partial x_A} \Gamma_x - \frac{y_l}{l} \frac{\partial}{\partial x_A} \Gamma_y - \frac{x_l^2}{l^3} \Gamma_x - \frac{y_l x_l}{l^3} \Gamma_y \right), \end{aligned} \quad (5.16a)$$

$$\mathbf{f}_{L_i}^T \frac{\partial \mathbf{d}_i}{\partial \theta_n} = q \frac{x_l}{l} \Gamma_x + q \frac{y_l}{l} \Gamma_y. \quad (5.16b)$$

5.3.2 Second derivative terms

Second derivatives (i.e., the Hessian matrices) can sometimes be approximated using Quasi-Newton methods (e.g., the BFGS method described in Nocedal et al. 2006). However, to ensure the NLP process is efficient as possible, they are derived analytically in this paper. Details of the mathematical expressions for the second derivative terms are given in Appendix C.

5.3.3 Assembling the entire problem

For a single yield-line, the analytical expressions for the first and second derivatives have been derived, and thus the entire problem can be readily assembled. In the case of constraint (5.1d), which handles live load effects, the corresponding mathematical expressions are quite complex, but can be obtained using symbolic calculation packages.

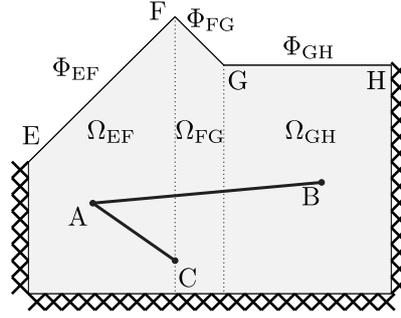


Figure 5.5: Node move limit constraints introduced by a non-smooth top edge: node A moves within zone Ω_{EF} , B within Ω_{GH} , C on line CF

5.4 Geometry optimization: practical issues

5.4.1 Modelling complex slab geometries

In this paper the boundaries of the slabs considered are assumed to be formed from piecewise linear segments, permitting complex slab geometries to be modelled (e.g., a slab with a non-convex polygonal external boundary and internal holes). Complex slab geometries may require special treatment, as will be considered in this section.

Non-smooth top edges

It was indicated that the vector \mathbf{f}_L used in constraint (5.1d) is calculated by considering the effects of load ‘above’ a given yield-line. Quite often, the top edge of a slab will contain several line segments; in this case, $y = \Phi_t(x)$ is a piecewise function that is non-smooth or discontinuous.

Figure 5.5 shows a slab with non-smooth top edge EFGH, and three nodes within the slab domain, A, B, and C. Let Φ_{EF} , Φ_{FG} , and Φ_{GH} denote the line segments of the top edge, dividing the slab into three zones, Ω_{EF} , Ω_{FG} , and Ω_{GH} . The piecewise function $\Phi_t(x)$ for the top edge can be written as:

$$\Phi_t(x) = \begin{cases} \Phi_{EF}(x), & x_E \leq x \leq x_F \\ \Phi_{FG}(x), & x_F \leq x \leq x_G \\ \Phi_{GH}(x), & x_G \leq x \leq x_H \end{cases} . \quad (5.17)$$

The unit-length moment and area functions Λ_α ($\alpha = x, y, z$) are now expressed as:

$$\Lambda_\alpha(x) = \begin{cases} \Lambda_\alpha^{EF}(x), & x_E \leq x \leq x_F \\ \Lambda_\alpha^{FG}(x), & x_F \leq x \leq x_G \\ \Lambda_\alpha^{GH}(x), & x_G \leq x \leq x_H \end{cases} , \quad (5.18)$$

where Λ_α^{EF} , Λ_α^{FG} , and Λ_α^{GH} are unit-length moment and area functions in zones Ω_{EF} , Ω_{FG} , and Ω_{GH} , respectively. The first derivatives of the unit live load effect Γ_x , Γ_y , and Γ_z can be derived

using (5.15). For example, for node A of yield-line AB:

$$\frac{\partial \Gamma_\alpha}{\partial x_A} = -\Lambda_\alpha^{\text{EF}} + \int_{x_A}^{x_F} \frac{\partial \Lambda_\alpha^{\text{EF}}}{\partial x_A} dx + \int_{x_F}^{x_G} \frac{\partial \Lambda_\alpha^{\text{FG}}}{\partial x_A} dx + \int_{x_G}^{x_B} \frac{\partial \Lambda_\alpha^{\text{GH}}}{\partial x_A} dx, \quad (5.19)$$

$$\frac{\partial \Gamma_\alpha}{\partial y_A} = \int_{x_A}^{x_F} \frac{\partial \Lambda_\alpha^{\text{EF}}}{\partial y_A} dx + \int_{x_F}^{x_G} \frac{\partial \Lambda_\alpha^{\text{FG}}}{\partial y_A} dx + \int_{x_G}^{x_B} \frac{\partial \Lambda_\alpha^{\text{GH}}}{\partial y_A} dx. \quad (5.20)$$

These formulae are valid only when node A lies within zone Ω_{EF} , so that node A must be restricted to lie within this zone. Also, node C must be restricted to lie on line CF lying between Ω_{EF} and Ω_{FG} . Thus, when a slab has a non-smooth top edge, each node must be restricted to lie within the zone in which it currently lies and, with only vertical movement permitted in the case of nodes lying directly below a non-smooth point. This can be considered to be a limitation of the method, as currently implemented.

Slab with holes

When a hole is present, calculating the effects of live loads is complicated by the need to exclude areas occupied by the hole in the vertical strip lying above a given yield-line. This has not been considered in the formulae introduced above. A means of incorporating holes using the presented formulae is to use domain decomposition. When using decomposition a slab domain can be divided into several sub-domains in which the holes are excluded; details are provided in Appendix D.

Non-convex polygonal slab

When moving nodes in a non-convex polygonal slab, a yield-line can potentially be moved so as to cross a slab boundary. This can either be addressed via domain decomposition (which involves dividing non-convex domains into several convex sub-domains) or by introducing additional constraints (not considered here). In the examples considered in this paper no yield-lines exhibiting the described behaviour were found to be present, and thus no action was necessary.

5.4.2 Inherited issues

In the truss rationalization formulation presented by He & Gilbert (2015a), steps were taken to address a number of practical issues, for example, restrictions on the movement of nodes, merging of nodes in close proximity, etc; these issues are addressed here using the same basic techniques.

Node move limits

Because of the non-convex nature of the optimization problem, the NLP solver (i.e., IPOPT) may report an unstable status. Furthermore, clearly nodes must be restricted from only lying within the geometry of the slab. To address these issues, in the truss rationalization formulation (He & Gilbert 2015a) node move limits were active for every node. In this paper, the same basic approach

is used; firstly, the nodes can only move within regions defined to be a function of nodal spacing; secondly, line and domain constraints are imposed according to the geometry of the slab.

In the first step, assume that the nodal coordinates of a node are written in \mathbb{R}^3 as $\boldsymbol{\nu} = [x, y, 1]^T$ (as per the truss rationalization formulation (He & Gilbert 2015a), the redundant ‘1’ is used to condense the mathematical expression). Consider two adjacent nodes A and B, and let $r_{AB} = \frac{1}{2} \|\boldsymbol{\nu}_B^0 - \boldsymbol{\nu}_A^0\|_2$ be half the distance between them, ϵ be a gap used to avoid generating a zero length yield-line, and r_s be a program parameter that defines the maximum node move limit for all nodes. The node move limit is then obtained as $r^* = \min\{r_{AB}, r_s\} - \epsilon$.

In the second step, nodes on slab boundaries must be restricted to lie on boundary lines in order to retain the slab geometry; therefore, line constraints are imposed on these nodes. As in the truss rationalization formulation (He & Gilbert 2015a), let \mathbf{T} be the coefficient vector of a line so that the line constraint is written as $\mathbf{T}\boldsymbol{\nu} = 0$; for domain constraints, an inequality constraint is instead used (also note that \mathbf{T} can now be a matrix to describe several lines).

Merging nodes

During the rationalization process, certain nodes may migrate towards each other. A node merge process was introduced in the truss rationalization formulation (He & Gilbert 2015a), and this approach is also adopted here: first, the nodes are grouped based on distances; then, merging every individual group is attempted, provided that the resulting yield-line pattern is validated numerically.

Extracting yield-line patterns from DLO

The rationalization process requires an initial yield-line pattern to be extracted from a DLO analysis. Typically, such a pattern is obtained by removing yield-lines having rotations (θ_n) that are smaller than a prescribed threshold value (except for boundary yield-lines, which are not removed). To ensure a reasonable threshold number is chosen, the extracted yield-line pattern will be used as the basis of a new analysis, and the load factor compared with that obtained originally. If these are not within a prescribed tolerance then the threshold value should be progressively reduced until the load factor obtained is within the prescribed tolerance, and a usable initial yield-line pattern is obtained.

Crossovers

Typically, yield-line patterns obtained using DLO will include crossover points where two or more yield-lines intersect that do not coincide with nodes. As with the truss rationalization formulation (He & Gilbert 2015a), nodes can be added at these locations using a nested-loop strategy: an inner loop performs geometry optimization and, whenever the inner loop finishes, crossover nodes are created in the outer loop, and then a further cycle of the inner loop is performed. The whole process is repeated until no crossover points are found.

5.5 Geometry optimization: full formulation

Consider a slab that comprises $\mathbb{N} = \{1, 2, \dots, n\}$ nodes, with node subsets \mathbb{N}^L and \mathbb{N}^D denoting those nodes that lie on the boundary lines and those close to domain boundaries, respectively. The full optimization problem, now considering nodal move limits, can be written as:

$$\min_{x,y,d,p} \quad \lambda \mathbf{f}_L^T \mathbf{d} = \mathbf{g}^T \mathbf{p} \quad (5.21a)$$

$$\text{s.t.} \quad \mathbf{B} \mathbf{d} = \mathbf{0} \quad (5.21b)$$

$$\mathbf{N} \mathbf{p} - \mathbf{d} = \mathbf{0} \quad (5.21c)$$

$$\mathbf{f}_L^T \mathbf{d} = 1 \quad (5.21d)$$

$$\mathbf{p} \geq \mathbf{0} \quad (5.21e)$$

$$\|\boldsymbol{\nu}_j - \boldsymbol{\nu}_j^0\|_2^2 \leq (r^*)^2 \text{ for all } j \in \mathbb{N} \quad (5.21f)$$

$$\mathbf{T}_j^L \boldsymbol{\nu}_j = 0 \text{ for all } j \in \mathbb{N}^L \quad (5.21g)$$

$$\mathbf{T} \boldsymbol{\nu}_j \geq \mathbf{0} \text{ for all } j \in \mathbb{N}^D \quad (5.21h)$$

$$\mathbf{x}^{\text{lb}} \leq \mathbf{x} \leq \mathbf{x}^{\text{ub}} \quad (5.21i)$$

$$\mathbf{y}^{\text{lb}} \leq \mathbf{y} \leq \mathbf{y}^{\text{ub}}, \quad (5.21j)$$

where \mathbf{x}^{lb} , \mathbf{x}^{ub} , \mathbf{y}^{lb} , and \mathbf{y}^{ub} are the lower and upper bounds of the nodal positions, which are calculated by taking account of the practical issues that affect node movements (e.g. limits imposed to address non-smooth top edges).

5.6 Numerical examples

In this section, the efficacy of the proposed rationalization technique is demonstrated by applying it to various numerical example problems. Unless stated otherwise, the slabs considered have unit moment resistance per unit length, and are subjected to a uniform pressure load of unit intensity. Also, a default node merge radius of $0.25 \times$ the x or y -nodal spacing in the original DLO analysis was assumed. To solve both the LP and NLP problems, the IPOPT 3.11.0 (Vigerske & Wachter 2013) interior point optimization solver was used, with a maximum of 500 iterations allowed. All calculations were performed using MATLAB2013a running under the Microsoft Windows 7 operating system on an Intel i5-2310 powered desktop with 6G RAM. Finally, unless stated otherwise, the line thickness of the plotted yield-lines are proportional to the yield-line rotation.

5.6.1 Gilbert et al. (2014) examples

In Gilbert et al. (2014), the proposed DLO-based automatic yield-line analysis method was applied to several slab problems. These examples will now be revisited, with the DLO derived yield-line patterns now rationalized using the new procedure. Thus in Table 5.1, both standard DLO and rationalized solutions are presented.

It is evident that the rationalization process successfully simplifies the yield-line patterns, and also

improves the solutions (i.e. reduces the load factors). The linear nature of the DLO formulation means that large-scale problems, e.g. involving millions of potential yield-line discontinuities, can be solved without difficulty. In comparison the NLP problem associated with the geometry optimization formulation is considerably more difficult to solve. However, fortunately the size of the problem which needs to be solved in the proposed procedure is much reduced, containing several orders of magnitude fewer yield-line discontinuities. Table 5.2 shows how the CPU time increases with increasing number of nodes and yield-lines, for the fixed square slab problem. Also Figure 5.6 shows solutions for this problem for the 60 and 120 nodal division cases (nodes are shown but, for sake of clarity, a constant yield-line line thicknesses has been used). It can be observed that the rationalized patterns contain far fewer nodes and yield-lines than present in the final DLO solutions.

Alternatively, fewer nodes can be employed in the initial DLO problem to ensure that even simpler solutions are obtained; such solutions are potentially attractive to practitioners, who may require yield-line patterns which are easy to visualise and to hand-check. Thus, Fig. 5.7 shows solutions for the slab with alcoves problem with various nodal divisions. The coarsest solution corresponds to an extremely simple yield-line pattern but is still within 5% of the extrapolated solution (of 35.230) given in Gilbert et al. (2014), which can be considered for all practical purposes to be exact. Also, because a very coarse initial grid has been used, the solution could be obtained in a fraction of a second.

Finally, since the geometry optimization rationalization step will generally improve the numerical solution (i.e. will reduce the load factor), it is of interest to ascertain whether it can be used to reduce the total CPU time required to achieve a solution of a given accuracy. Figure 5.8 presents results for the fixed square slab problem, showing that use of the rationalization step can indeed reduce the CPU time required to give a solution of a given accuracy.

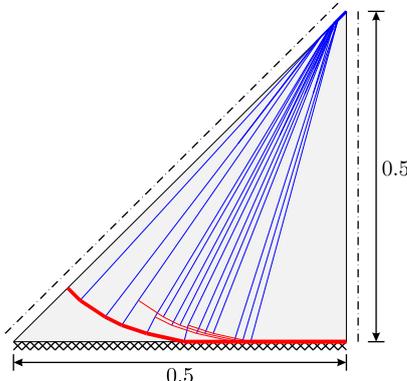
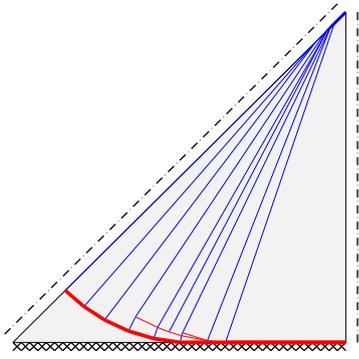
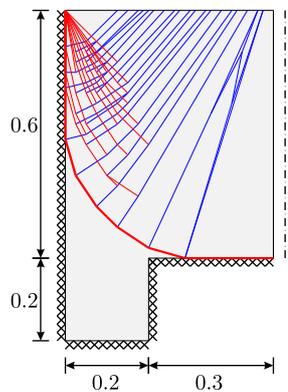
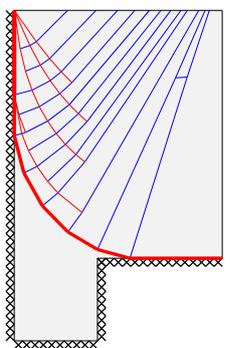
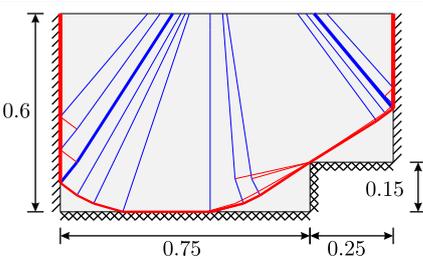
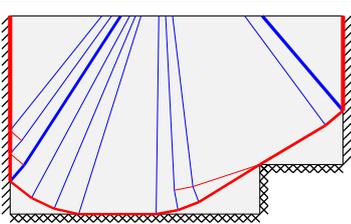
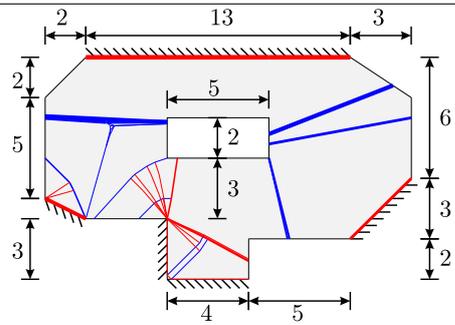
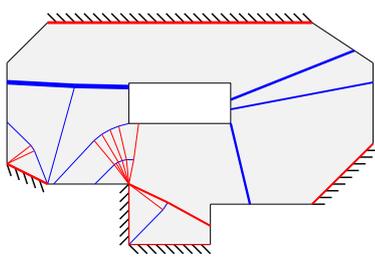
5.6.2 Irregular slabs with corner fans

It is well-understood that fan-type mechanisms develop at clamped corners. However, fan-type mechanisms have proved difficult to identify using traditional automated yield-line analysis methods (e.g. Munro & Da Fonseca 1978, Johnson 1994). It is therefore of interest to consider two representative examples here.

The first example comprises a rectangular slab with fixed supports and a corner cutout, originally considered by Islam & Park (1971), and, more recently, by Jackson (2010). The slab geometry and solutions are shown in Fig. 5.9. To obtain the DLO solution (of 25.135) a total of 20 nodal divisions per unit length were used. The solution was then improved upon using the proposed geometry optimization rationalization technique, giving a solution of 25.103, which is just 0.8% higher than the lower bound solution of 24.9 quoted by Jackson (2010).

The second example is a five-sided slab, originally investigated by Kwan (2004). The slab has fixed supports on two sides, with column supports coinciding with the remaining two vertices (Fig. 5.10). Kwan obtained a load factor of 0.1967 for this problem, with no fan-type mechanism included in his assumed yield-line pattern. In contrast the DLO solution shown in Fig. 5.10(a) clearly shows the presence of a fan-type mechanism, the form of which becomes even clearer

Table 5.1: Gilbert et al. (2014) examples: DLO and rationalized yield-line patterns

Problem	DLO solution	Rationalized solution
Fixed square slab (Fox 1974): 40 nodal divisions along each leg of the right-angled triangle domain	 <p>$\lambda = 42.934$ CPU time: 66 s</p>	 <p>$\lambda = 42.892$ CPU time[†]: 72 s</p>
Alcove slab (Regan & Yu 1973): 40 nodal divisions per unit length	 <p>$\lambda = 35.411$ CPU time: 9 s</p>	 <p>$\lambda = 35.353$ CPU time[†]: 37 s</p>
Indented slab (Regan & Yu 1973): 40 nodal divisions per unit length	 <p>$\lambda = 29.062$ CPU time: 19 s</p>	 <p>$\lambda = 29.026$ CPU time[†]: 6 s</p>
Slab with hole (Olsen 1998): five nodal divisions per unit length	 <p>$\lambda = 0.13557$ CPU time: 48 s</p>	 <p>$\lambda = 0.13554$ CPU time[†]: 70 s</p>

†: Time for geometry optimization rationalization step only

Table 5.2: Fixed square slab: influence of number of DLO nodal divisions

Nodal divisions	DLO				Geometry optimization rationalization			
	No. of nodes	No. of yield-lines	Load factor (error)	CPU time	No. of nodes	No. of yield-lines	Load factor (error)	CPU time [†]
20	291	28037	43.055 (0.48%)	2	9	13	42.969 (0.28%)	2
40	981	285204	42.934 (0.19%)	66	30	52	42.892 (0.10%)	72
60	2071	1041621	42.908 (0.13%)	278	53	88	42.890 (0.10%)	174
80	3561	2430190	42.887 (0.09%)	1105	201	418	42.873 (0.05%)	655
100	5451	4496066	42.879 (0.06%)	1704	487	1118	42.867 (0.04%)	1416
120	7741	7258302	42.874 (0.05%)	4845	774	2069	42.863 (0.03%)	2304

[†]: Time for geometry optimization rationalization step only

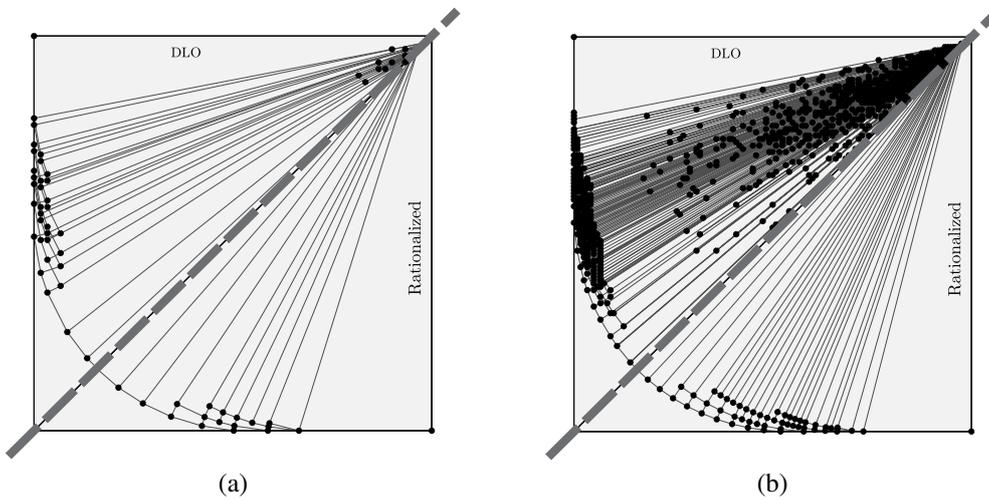


Figure 5.6: Fixed square slab: comparison of DLO and rationalized yield-line patterns for: (a) 60 nodal divisions; (b) 120 nodal divisions

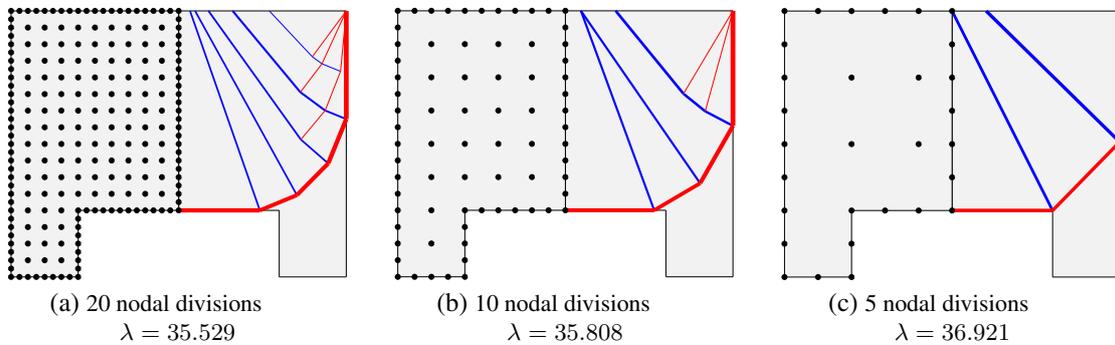


Figure 5.7: Slab with alcoves: coarse resolution DLO solutions suitable for hand checking (left: initial DLO nodal grid; right: rationalized solution)

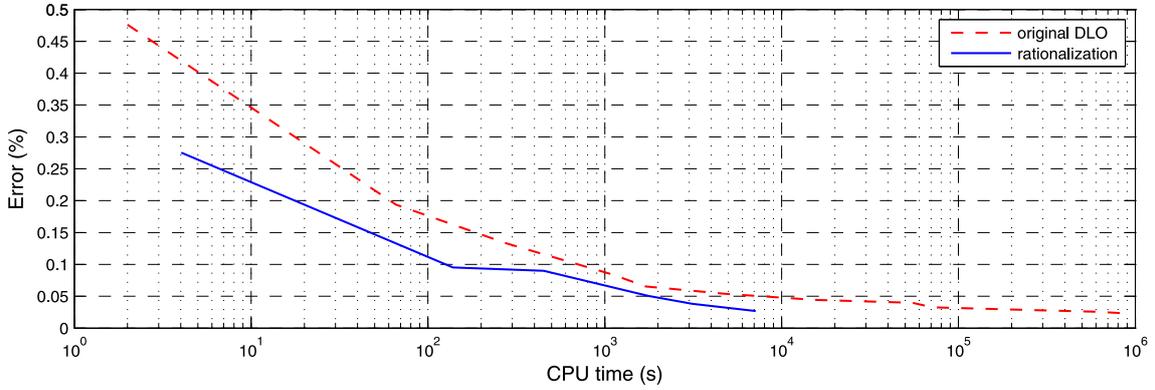


Figure 5.8: Fixed square slab: CPU time vs. percentage error when using DLO alone (dashed line) and DLO with geometry optimization rationalization (solid line). (Note that in the latter case the CPU time includes both DLO and geometry optimization stages.)

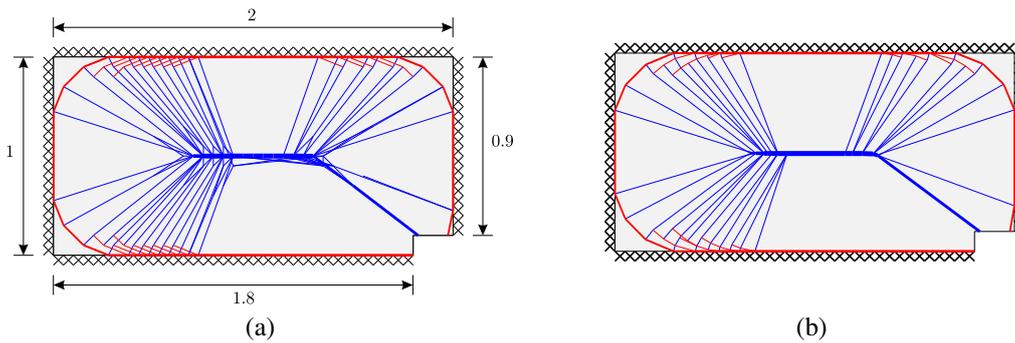


Figure 5.9: Islam and Park's slab: (a) DLO solution (20 nodal divisions per unit length), $\lambda = 25.135$; (b) rationalized solution, $\lambda = 25.103$

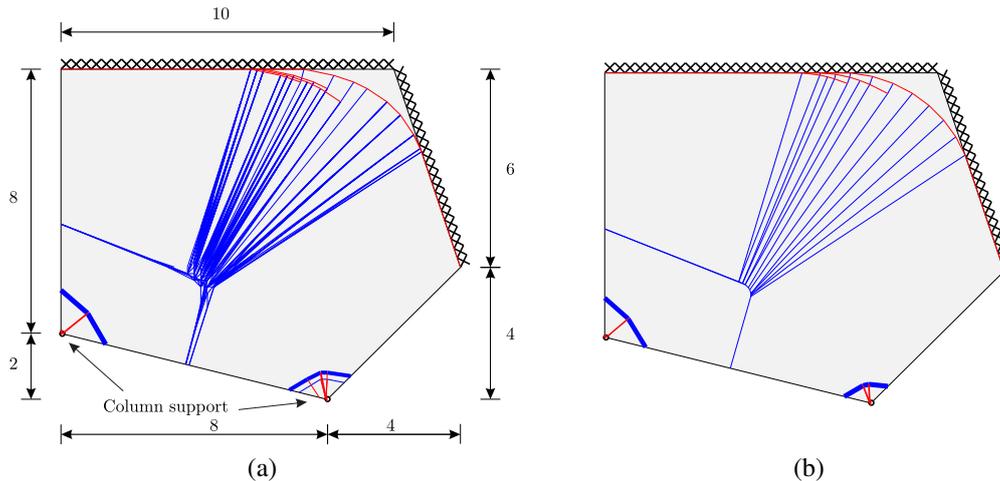


Figure 5.10: Kwan's five sided slab: (a) DLO (five nodal divisions per unit length), $\lambda = 0.18849$; (b) rationalized solution, $\lambda = 0.18775$

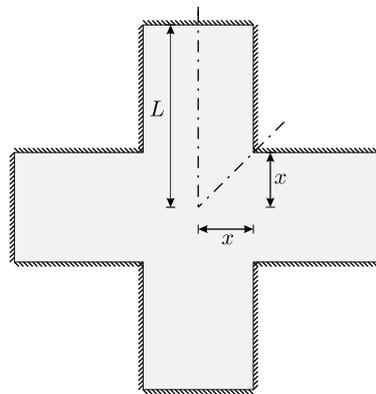


Figure 5.11: Cruciform slab: problem specification ($L = 1$)

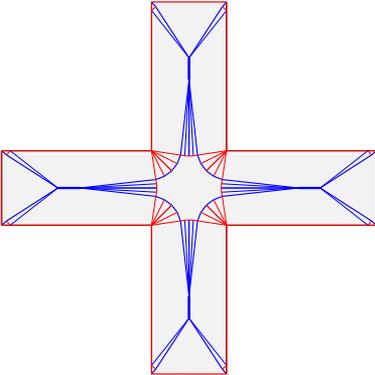
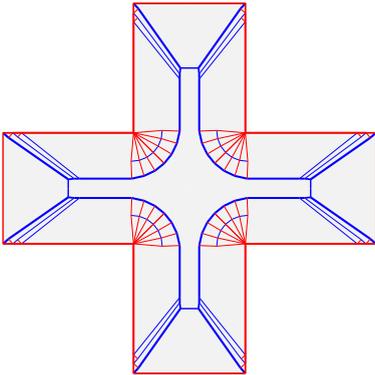
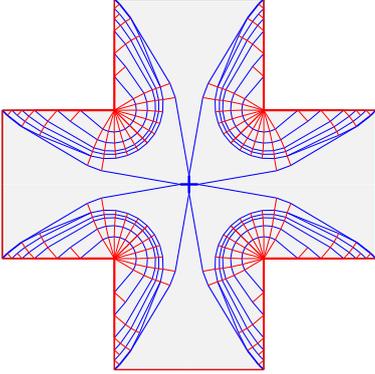
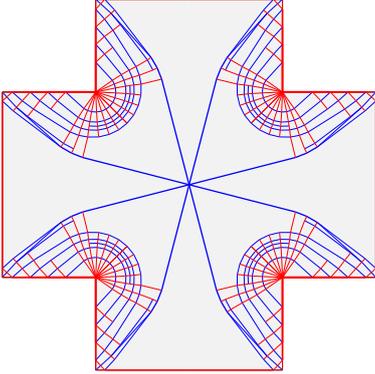
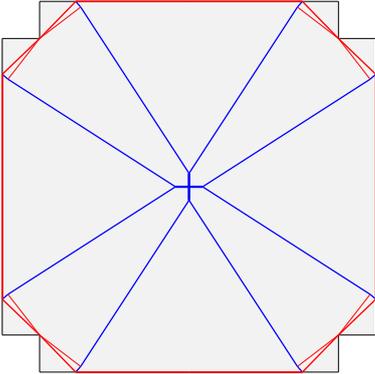
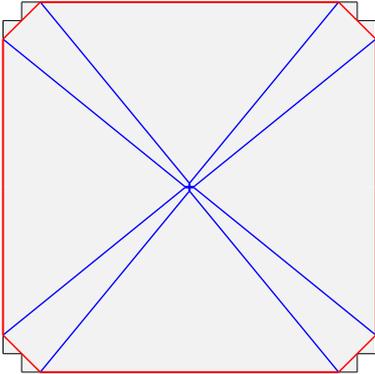
following rationalization. The rationalized solution of 0.18775 is some 4.5% less than the solution obtained by Kwan.

5.6.3 Cruciform slab

Johnson (1994) investigated the critical yield-line patterns for a simply supported cruciform slab of various dimensions; see Fig. 5.11. He identified three yield-line patterns: the 'crossed rectangular slab' mode for low values of x ; the 'modified square slab' mode for intermediate values of x ; and the 'corner lever' mode for high values of x . More recently, Jackson (2010) revisited the problem, though presented only lower bound solutions and 'yield-line indicators' (obtaining yield-line solutions using his proposed method involved human-intervention, and would likely have been labour intensive to perform for multiple geometries).

However, here the rationalization procedure has been used to automatically generate clear patterns for the cruciform slab problem; see Table 5.3. In the first two modes, a fan-type mechanism can clearly be observed near the concave corners.

Table 5.3: Cruciform slab: rationalized yield-line patterns for various x/L ratios

Failure mode	Rationalized solutions	
Crossed rectangular slab mode		
	$x/L = 0.2, \lambda = 71.940$	$x/L = 0.3, \lambda = 38.344$
Modified square slab mode		
	$x/L = 0.4, \lambda = 24.248$	$x/L = 0.5, \lambda = 16.637$
Corner lever mode		
	$x/L = 0.8, \lambda = 7.1549$	$x/L = 0.9, \lambda = 6.2575$

5.7 Conclusions

- For many decades the yield-line method of analysis for reinforced concrete slabs eluded systematic automation. This has finally now been achieved, via the discontinuity layout optimization (DLO) procedure, which can rapidly obtain high accuracy solutions for slabs of arbitrary geometry. However, the use of a fixed nodal grid means that the corresponding yield-line patterns can be somewhat more complex in form than is necessary.
- To address this, in this paper a post-processing rationalization step which involves the use of geometry optimization to adjust the positions of nodes has been proposed. As the yield-line patterns obtained via DLO normally contain only a relatively small number of nodes and yield-lines, solutions to the inherently non-linear geometry optimization problem can be obtained relatively rapidly using an interior point solver. The rationalized solutions are generally both simpler in form and more accurate than the raw DLO solutions, clearly demonstrating the efficacy of the proposed procedure.

5.8 Acknowledgements

The first author acknowledges the doctoral studies scholarship provided by the Faculty of Engineering at the University of Sheffield.

Chapter 6

Automatic yield-line analysis of practical slab problems via discontinuity layout optimization^[i]

Preface

As introduced in Chapter 5 (see also Appendix A), the yield-line method of analysis has now been systematically automated via DLO. It is however of interest to ensure that the slab problems found in engineering practice can be treated using the new method (see also the paper accepted for publication in *The Structural Engineer* in Appendix B). Practical slabs involve more complicated configurations than the slabs analysed in Chapter 5. For example, they can be orthotropically reinforced, involve various forms of loading (e.g., point, line and patch loads may be applied), and more complex support types may be present. This chapter describes the developments required to model these features, and the application of the resulting analysis procedure to a wide range of practical slab problems.

^[i]The content of this chapter was originally prepared for a journal paper: He, L., Gilbert, M., Shepherd, M. (2015), 'Automatic yield-line analysis of practical slab problems via discontinuity layout optimization', *ASCE Journal of Structural Engineering* (submitted for publication).

Abstract The yield-line method provides a powerful means of rapidly estimating the ultimate load which can be carried by a reinforced concrete slab. The method can reveal hidden reserves of strength in existing slabs, and can lead to highly economic slabs when used in design. Originally conceived before the widespread availability of computers, the yield-line method subsequently proved difficult to computerise, limiting its appeal in recent years. However, it was recently demonstrated that the discontinuity layout optimization (DLO) procedure could be used to systematically automate the method, and various isotropically reinforced, uniformly loaded, slab examples were used to demonstrate this. The main purpose of this paper is to demonstrate that the DLO procedure can also be applied to a wide range of more practical slab problems, for example involving orthotropic reinforcement, internal columns, and point, line and patch loads. The efficacy of the procedure is demonstrated via application to a variety of example problems from the literature; for all problems considered solutions are presented which improve upon existing numerical solutions. Furthermore, in a number of cases solutions derived using previously proposed automated yield-line analysis procedures are shown to be highly non-conservative.

Keywords yield-line analysis, plastic analysis, discontinuity layout optimization, slabs, rationalization.

6.1 Introduction

The yield-line method of analysis (Johansen 1943) is a long-established and extremely powerful tool for estimating the maximum load sustainable by a reinforced concrete slab. In order to apply the method successfully it has traditionally been necessary for users to have some knowledge of the rules governing the construction of viable yield-line patterns, though these rules can be memorised, and for simple problems a hand analysis is quick and easy to perform. The guidance document produced by the UK Concrete Centre (Kennedy & Goodchild 2004) discusses the many benefits of yield-line design, in particular highlighting the highly economic reinforcement layouts that can result from its application. Furthermore, various other guidance documents are available to assist new users, some of which also include useful formulae covering standard cases.

However, in many practical cases it can be difficult to identify the critical yield-line pattern by hand. This is true when the slab under consideration has unusual geometry, reinforcement configuration and/or pattern of applied loading. The presence of fixed (or ‘clamped’) edges can also cause difficulties since in reality complex yield-line patterns (e.g. involving ‘corner fans’) will often be critical in such cases, and these can be difficult to deal with in a hand analysis. Most importantly, it must be borne in mind that the yield-line method is an upper-bound method in the context of the fundamental theorems of plasticity, which means that an incorrectly chosen yield-line pattern will result in an unsafe estimate of the strength of the slab under consideration.

To address this, an automated method of identifying critical yield-line patterns was first proposed in the 1970s by Chan (1972), then working at the University of Oxford, and subsequently by Munro & Da Fonseca (1978), working at Imperial College, London. Both groups of researchers discretized the slab under consideration into rigid elements separated by potential yield-lines, and

then used linear programming (LP) techniques to identify the critical yield-line pattern. Unfortunately, when using rigid elements it can be observed that the solutions obtained depend on the layout of the mesh discretization employed. This means that in many cases progressively reducing the size of the mesh does not lead to convergence towards the exact solution (e.g. as demonstrated in the recent study by Bleyer & Buhan (2013)). Various groups of researchers, for example Johnson and co-workers (Johnson 1994, Ramsay & Johnson 1998) and Thavalingam et al. (1999), attempted to address this through the use of a two-stage procedure. This involved supplementing the original rigid element procedure with a geometry optimization phase, allowing the positions of nodes to be adjusted to try to improve the solution. The main drawback is that such procedures rely on the initial solution being of the same form as the true optimal layout. This is not necessarily the case and, in mathematical optimization terms, such procedures will therefore be prone to identifying solutions which are locally rather than globally optimal (e.g. Johnson (1994) conceded that his proposed two stage approach “does not directly generate likely critical collapse modes”).

In the absence of general tools, various automated hand calculation yield-line analysis methods have been developed, such as the COConcrete BRidge ASsessment (COBRAS) package developed by Middleton (1997) specifically for bridge assessment. This proved to be a very useful tool, showing that many existing reinforced concrete slab bridges possessed significantly greater capacity than indicated by elastic analysis methods. However, as the tool relies on the use of an in-built library of predefined yield-line patterns, it is only suitable for analysing a restricted range of slab geometries.

In parallel various methods which seek to identify lower bound solutions have been investigated, such as the methods presented by Anderheggen, E. and Knöpfel, H. (1972), Krabbenhoft & Damkilde (2003) and more recently by Le, Gilbert & Askes (2010). However it should be noted that these methods are comparatively complex since they involve the use of a non-linear yield-function, and also are not capable of identifying discrete yield-lines directly (though these can be manually inferred from the output, as undertaken as part of the two-step slab analysis procedure recently described by Jackson & Middleton (2013)).

Given the inherent limitations of existing techniques, the opportunity was recently taken to apply the Discontinuity Layout Optimization (DLO) procedure (Smith & Gilbert 2007) to the analysis of reinforced concrete slabs. Although full details are provided by Gilbert et al. (2014), in the present paper key features of the procedure are briefly outlined. It is then demonstrated that the procedure may straightforwardly be extended to treat practical slab analysis problems, involving orthotropic reinforcement, a wider variety of support conditions and also slabs which are subject to point, line and patch loads. Additionally, it is shown that a recently developed rationalization procedure (He & Gilbert 2015*b*) can be used to enhance the solutions obtained. In this paper the DLO-based procedure is applied to both benchmark problems from the literature and to more practical slab configurations, with the aim being to clearly demonstrate its accuracy and usefulness.

6.2 Discontinuity Layout Optimization (DLO) formulation

A complete DLO analysis comprises several steps (Fig. 6.1). Firstly, the slab is discretized using nodes spatially distributed across the problem domain (Fig. 6.1*b*), which are then interconnected

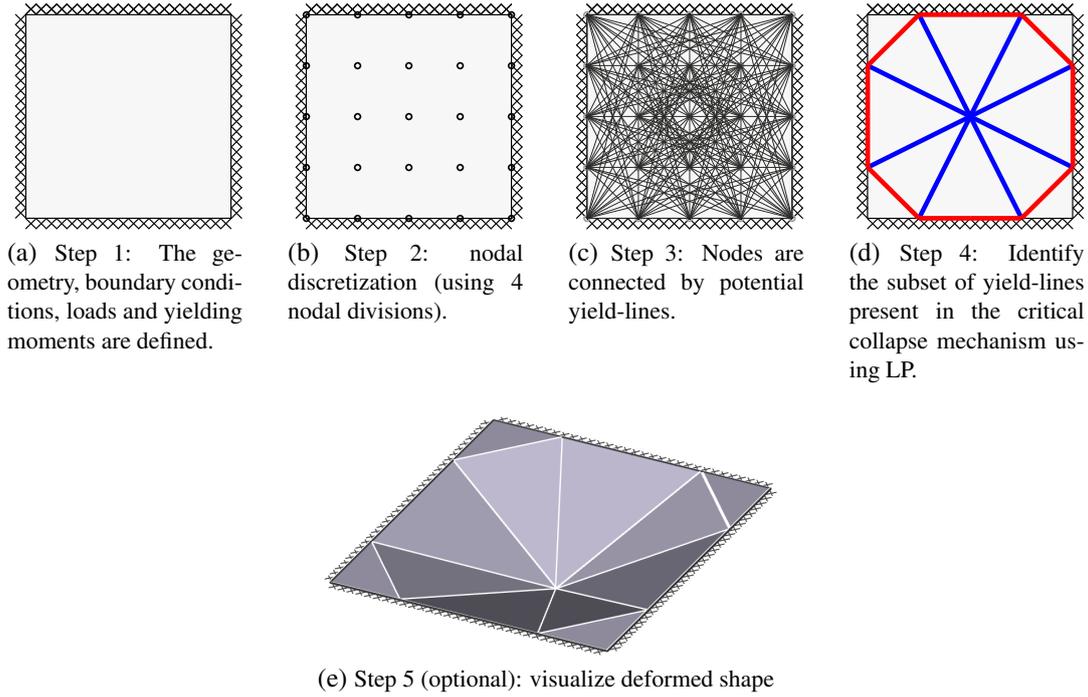


Figure 6.1: Steps in DLO process: a simple example (after He & Gilbert 2015b)

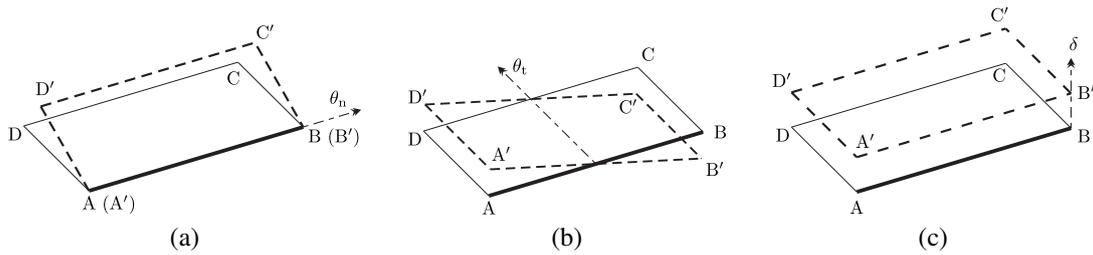


Figure 6.2: Displacement variables for a yield-line AB (assuming block ABCD moves to A'B'C'D'): (a) normal rotation along yield-line; (b) twisting rotation; (c) out-of-plane displacement

with potential yield-lines (Fig. 6.1c). Assuming a kinematic problem formulation, the formulation involves the variables shown in Fig. 6.2: normal rotation θ_n along yield-line, twisting rotation θ_t and out-of-plane displacement δ . With respect to these displacement variables, a linear programming (LP) problem comprising an objective function and constraints can be formulated (after Gilbert et al. 2014):

$$\min_{d,p} \lambda \mathbf{f}_L^T \mathbf{d} = -\mathbf{f}_D^T \mathbf{d} + \mathbf{g}^T \mathbf{p} \quad (6.1a)$$

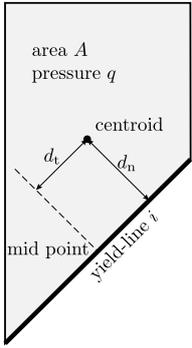
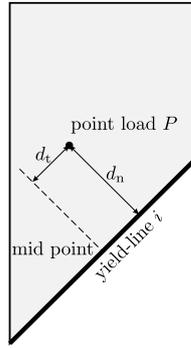
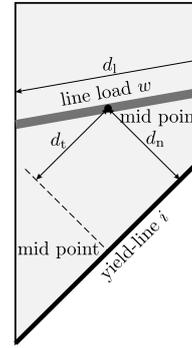
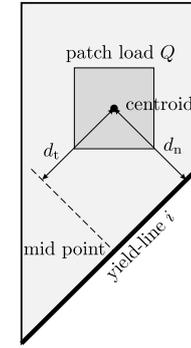
$$\text{s.t. } \mathbf{Bd} = \mathbf{0} \quad (6.1b)$$

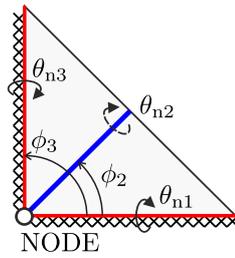
$$\mathbf{Np} - \mathbf{d} = \mathbf{0} \quad (6.1c)$$

$$\mathbf{f}_L^T \mathbf{d} = 1 \quad (6.1d)$$

$$\mathbf{p} \geq \mathbf{0}, \quad (6.1e)$$

Table 6.1: Evaluating the effects of loads lying in a strip ‘above’ yield-line i

Load type	Pressure	Point load	Line load	Patch load
				
Rotational moment f_{ni}	qAd_n	Pd_n	$wd_l d_n$	Qd_n
Torsional moment f_{ti}	qAd_t	Pd_t	$wd_l d_t$	Qd_t
Shear force f_i	qA	P	wd_l	Q



$$\sum_{i=1}^3 \theta_{ni} \cos \phi_i = 0$$

$$\sum_{i=1}^3 \theta_{ni} \sin \phi_i = 0$$

Figure 6.3: Compatibility requirements for yield-lines meeting at a node

where, \mathbf{d} and \mathbf{p} are vectors containing respectively the aforementioned displacement variables and corresponding non-negative plastic multiplier variables. In the objective function (6.1a), λ is a dimensionless load factor, here applied only to live loads; $\lambda \mathbf{f}_L^T \mathbf{d}$ and $\mathbf{f}_D^T \mathbf{d}$ describe the external work done respectively by live and dead loads (calculated in DLO by considering the effects of loads lying in strips ‘above’ each yield-line; the coefficients in \mathbf{f}_L and \mathbf{f}_D for the load types considered in this paper are provided in Table 6.1). Also $\mathbf{g}^T \mathbf{p}$ describes the internal energy dissipation along yield-lines. In constraint (6.1b), \mathbf{B} is a compatibility matrix used to ensure yield-line displacements are kinematically admissible; see also Fig. 6.3. In constraint (6.1c), \mathbf{N} is a plastic flow rule matrix describing the relation between the yield-line displacements in \mathbf{d} and their associated plastic multipliers \mathbf{p} . Also, in constraint (6.1d), the external work done by live load is normalized to ensure that λ directly defines the load factor. By solving the linear optimization problem (6.1), the load factor at collapse and the associated yield-line pattern can be obtained. The deformed shape can also be plotted (Fig. 6.1e), to clearly indicate the form of the predicted failure mechanism.

Whereas the example shown in Fig. 6.1 contains very few nodes, in practice much denser nodal grids can be employed to obtain more accurate solutions. However, a side effect of this is that the resulting yield-line patterns can become quite complex in form. To simplify these a post-processing rationalization step, which involves adjusting the positions of the nodes via geometry optimization, can optionally be performed (He & Gilbert 2015b). Unlike previously proposed

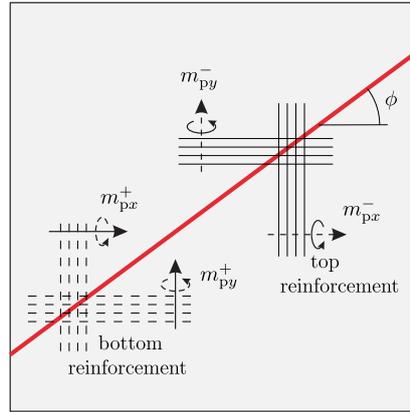


Figure 6.4: Yield moments m_{px}^+ , m_{py}^+ , m_{px}^- and m_{py}^- in an orthotropically reinforced concrete slab

methods which require a manual interpretation step (e.g. Johnson 1995, Jackson & Middleton 2013), here the rationalization is performed automatically following completion of a standard DLO analysis, generating yield-line patterns that are both simplified (i.e. contain fewer nodes and yield-lines) and also more critical (i.e. the solutions are better). The extensions to the mathematical derivations described in He & Gilbert (2015b) required to enable treatment of the practical slabs considered in the present paper are provided in Appendix E.

6.3 Modelling features of practical slabs

Orthotropic slab reinforcement

In engineering practice many slabs contain orthotropic reinforcement; such slabs were not considered in Gilbert et al. (2014). However it will be shown here that orthotropic reinforcement can be handled using the DLO method. Referring to Fig. 6.4, suppose m_{px}^+ , m_{py}^+ and m_{px}^- , m_{py}^- are respectively the x and y direction positive and negative plastic moments of resistance per unit length. Also m_p^+ and m_p^- are the plastic moments of resistance per unit length along a yield-line that is inclined at an angle of ϕ to x -axis. Using the stepped yield-criterion approach proposed by Johansen (1943), the latter can be calculated from:

$$m_p^+ = m_{px}^+ \cos^2 \phi + m_{py}^+ \sin^2 \phi, \quad (6.2)$$

$$m_p^- = m_{px}^- \cos^2 \phi + m_{py}^- \sin^2 \phi. \quad (6.3)$$

In the DLO formulation (Gilbert et al. 2014), the above formulae can be used when calculating the internal energy dissipation terms for a given yield-line (i.e. the coefficients in \mathbf{g} , used in the objective function (6.1a)). Note that, since the orientation of a given yield-line connecting two nodes is known in advance, computing the energy dissipation terms is straightforward; hence orthotropically reinforced slabs can be treated without difficulty in DLO.

Table 6.2: Summary of line support types

Support type	Free	Symmetry	Fixed ^[a]	Simple (anchored)	Simple (non-anchored)	Knife-edge (anchored)	Knife-edge (non-anchored)
Symbol	—————	-----	XXXXXXXX	\\\\\\\\\\\\	\\\\\\\\\\\\	\\\\\\\\\\\\	\\\\\\\\\\\\
Displacement constraints (θ_t and δ)	None	None	$\theta_t = \delta = 0$	$\theta_t = \delta = 0$	$\frac{\theta_t l}{2} \pm \delta \geq 0$	$\theta_t = \delta = 0$	$\frac{\theta_t l}{2} \pm \delta \geq 0$
Type of normal rotation θ_n	Relative to support	Relative to support	Relative to support	Relative to support	Relative to support	Relative to interior of slab	Relative to interior of slab
Internal energy dissipation	0	$m_p \theta_n l$	$i m_p \theta_n l$	0	0	$m_p \theta_n l$	$m_p \theta_n l$

a: Variable support strength = $i m_p$

Boundary conditions

In Gilbert et al. (2014) and He & Gilbert (2015b), only four boundary conditions were considered (free, symmetry, fixed and simple (anchored) boundaries - see the first four columns of Table 6.2). However, in engineering practice further boundary conditions are frequently encountered; the remaining columns of Table 6.2 and Table 6.3 provide details of the additional boundary conditions considered here. Additionally, here an optional support strength factor i is applied in the case of fixed supports (i is then a multiplier in internal energy dissipation terms in the objective function, (6.1a)). Further explanation of the remaining boundary conditions follows.

(i) Non-anchored simple supports An external simple support can be anchored or non-anchored; in the latter case uplift may occur, which means that twisting and out-of-plane displacements can now be non-zero, with $\frac{\theta_t l}{2} \pm \delta \geq 0$ ensuring uplift (only) can occur.

(ii) Knife-edge supports These type of supports may be located internally beneath the slab, with the slab above the support being *continuous*. This means that the slab can rotate along the support. If there exists no *relative* normal rotation at the support (i.e. $\theta_n = 0$), then no yield-line forms at the support. If $\theta_n \neq 0$, then a yield-line develops along the support and internal energy dissipation needs to be accounted for. Regarding twisting rotation θ_t and out-of-plane displacement δ , both (a) anchored and (b) non-anchored knife-edge supports are considered here. In the latter case twisting and out-of-plane displacements can now be non-zero, and uplift can be allowed to occur (as with the non-anchored simple supports).

(iii) Column supports Columns are frequently used in engineering practice; these can effectively be modelled using a combination of the aforementioned line support types. An external column is normally modelled using a simple support, whilst internal columns can now be modelled using either enclosed knife-edge supports or, if an internal column passes through the slab, by fixed supports (since the slab region is now discontinuous). In the latter case an optional support strength factor i can be applied. A summary of all column support types is shown in Table 6.3.

Table 6.3: Summary of column support types

Column type	External	Internal		
		Type I (supporting and anchoring slab)	Type II (supporting but not anchoring slab)	Type III (passing through slab)
Symbol	◦			
Description	Formed by a short-length simple support	Formed by enclosed anchored knife-edge supports; slab is continuous	Formed by enclosed non-anchored knife-edge supports; slab is continuous	Formed by enclosed fixed supports ^[a] ; slab is discontinuous at the column
Displacement constraints along edges	$\theta_t = \delta = 0$	$\theta_t = \delta = 0$	$\frac{\theta_t l}{2} \pm \delta \geq 0$	$\theta_t = \delta = 0$
Internal energy dissipation along edges	0	$m_p \theta_n l$	$m_p \theta_n l$	$im_p \theta_n l$

a: Variable support strength = im_p

6.4 Application of the automated method

Dense nodal grids can be employed using a modern desktop computer, so that highly accurate numerical solutions can if necessary be obtained. To solve the standard DLO problems, including the largest problems, involving approx. 10,000 nodes, the LimitState:SLAB software was used (LimitState Ltd 2015); this software is freely available for academic use. To obtain rationalized solutions, the post-processing step described by He & Gilbert (2015b) was used, programmed in a MATLAB script. All results were obtained using an Intel i5-2310 based desktop PC with 6GB RAM and running Microsoft Windows 7.

Numerical results are summarized in Table 6.5, which contains both DLO solutions obtained using dense nodal grids involving 10,000 nodes, to provide highly accurate solutions, and rationalized DLO solutions, which are more easy to interpret visually. For problems with known analytical solutions, the margin of error was always found to be well within 1%. For other problems the results obtained in the present paper were found to be more accurate (i.e. safer) than those obtained using the numerical methods described in the current literature. Further details of each problem considered are provided in the following sections.

Singly reinforced slabs

Slabs which are singly reinforced (i.e. which have no hogging resistance) will be considered first; these are of particular interest as the critical failure mechanism is likely to involve ‘corner levers’ or ‘corner fans’ (which are often ignored by practicing engineers, who may instead pragmatically apply a margin of safety, e.g. 10%, to account for this and other simplifications).

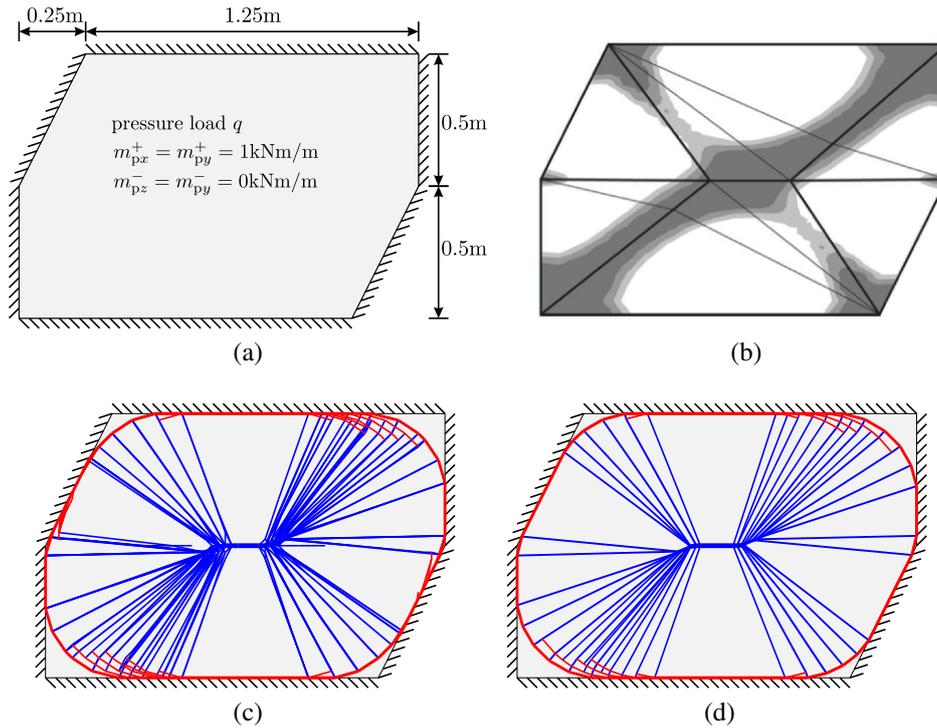


Figure 6.5: Six-sided plate (configuration 1): (a) problem specification; (b) yielding zone and yield-line pattern obtained by Wüst & Wagner (2008), $q = 17.75 \text{ kN/m}^2$; (c) 2000 node DLO yield-line pattern, $q = 15.992 \text{ kN/m}^2$ (when 10,000 nodes were used $q = 15.953 \text{ kN/m}^2$); (d) DLO yield-line pattern (rationalized, based on the 2000 node solution), $q = 15.970 \text{ kN/m}^2$

Isotropic six-sided slabs

The six-sided slabs shown in Fig. 6.5 and Fig. 6.6 were previously analysed by Wüst & Wagner (2008). In both cases $m_{px}^+ = m_{py}^+$ and $m_{pz}^- = m_{py}^- = 0$ (i.e. there is no resistance to hogging moments). It is evident that the yield-line patterns identified by Wüst & Wagner (2008), shown in Fig. 6.5(b) and Fig. 6.6(b), involve yield-lines which intersect each of the corners, and hence do not include corner levers and/or corner fans. In contrast corner fans are evident in both the standard DLO solutions (shown in Fig. 6.5(c) and Fig. 6.6(c)), and the rationalized DLO solutions (shown in Fig. 6.5(d) and Fig. 6.6(d)). Comparing the solutions obtained by Wüst & Wagner (2008) and those obtained using DLO, it is evident that the former are more than 10% higher than the latter (using 10,000 node DLO solutions provided here as benchmarks, differences for the two problems are: $(17.75/15.953 - 1) \times 100\% = 11.3\%$ & $(54.4/47.424 - 1) \times 100\% = 14.7\%$), indicating that using the yield-line patterns identified by Wüst & Wagner (2008) would be non-conservative even if the usual 10% margin of safety was applied.

Orthotropic trapezoidal slab

The trapezoidal slab shown in Fig. 6.7 was previously analysed by Balasubramanyam & Kalyanaraman (1988). The slab is orthotropically reinforced, and simply supported on three sides. The relatively simple yield-line pattern identified by Balasubramanyam & Kalyanaraman (1988), shown in Fig. 6.7(b), does not include corner levers and/or corner fans, and corresponds to a re-

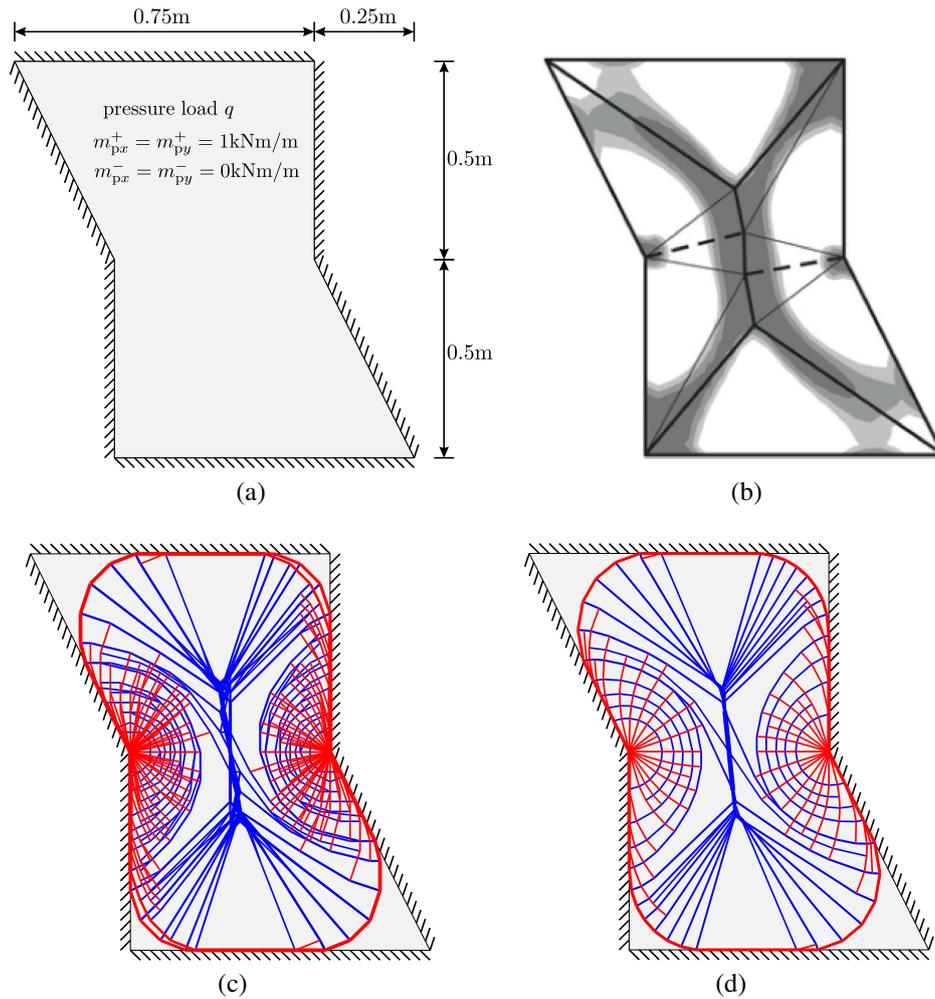


Figure 6.6: Six-sided plate (configuration 2): (a) problem specification; (b) yielding zone and yield-line pattern obtained by Wüst & Wagner (2008), $q = 54.4 \text{ kN/m}^2$; (c) 2000 node DLO yield-line pattern, $q = 47.745 \text{ kN/m}^2$ (when 10,000 nodes were used $q = 47.424 \text{ kN/m}^2$); (d) DLO yield-line pattern (rationalized, based on the 2000 node solution), $q = 47.501 \text{ kN/m}^2$

quired moment capacity of $m_{px}^+ = 11.84q$. In contrast the DLO solution shown in Fig. 6.7(c) is more complex, and corresponds to a required moment capacity that is 18% higher (using the most accurate DLO solution available here, $(13.97/11.84 - 1) \times 100\% = 18\%$). This again indicates that application of the usual 10% margin of safety would be insufficient to ensure a safe design.

Slabs with internal supports

Roof slab

The slab shown in Fig. 6.8 has two external column supports and two knife-edge supports. The problem was originally analysed by Bäcklund (1973), then by Munro & Da Fonseca (1978) and subsequently by Balasubramanyam & Kalyanaraman (1988). The rationalized DLO solution is shown in Fig. 6.8(c). A local failure can be observed near one external column; additionally, a fan-type mechanism is developed near the knife-edge supports. Though the slab is not anchored

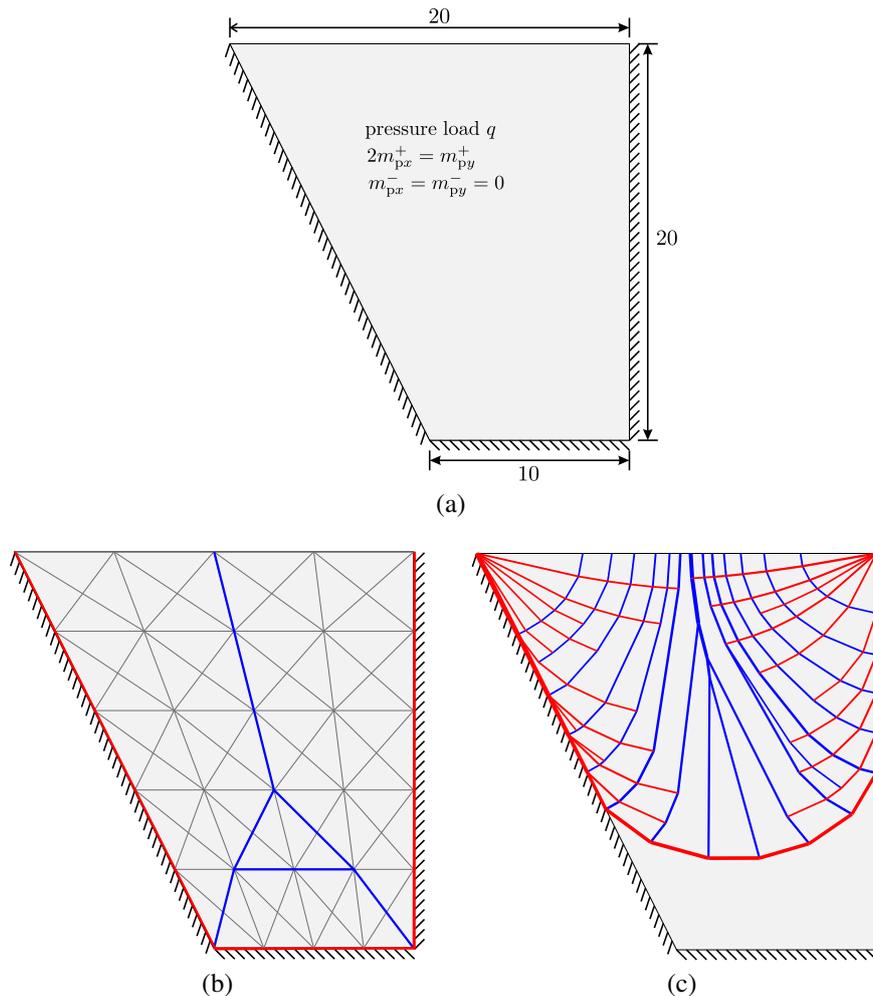


Figure 6.7: Trapezoidal orthotropic slab: (a) problem specification; (b) yield-line pattern obtained by Balasubramanyam & Kalyanaraman (1988), $m_{px}^+ = 11.84q$; (c) DLO yield-line pattern (rationalized), $m_{px}^+ = 13.90q$ (10,000 node DLO solution, $m_{px}^+ = 13.97q$)

to the knife-edge supports, the failure mechanism shows no uplift of the slab.

Johansen's slab with point supports

Johansen (1943) investigated a slab having two point supports (Fig. 6.9(a)). Ideally a point support acts as a fulcrum, providing no rotational restriction and permitting uplift without necessitating plastic deformation of the slab. However, in DLO a point support is most conveniently modelled as a column of finite, though small, size (see Fig. 6.9(b)). This means that care must be exercised since if a column restricts translational displacement of the slab along all its edges (e.g. a Type I or III column in Table 6.3), plastic deformation must occur near the column when uplift behaviour occurs. In contrast, a Type II column behaves similarly to a point support, except that the fulcrum has been offset from the initial position (cf. Fig. 6.9(a)). The influence of the support type and column size μ on the critical yield-line pattern is shown in Table 6.4. It can be observed that:

- Firstly, when a relatively small column size is used ($\mu = 0.001$), the resulting global yield-

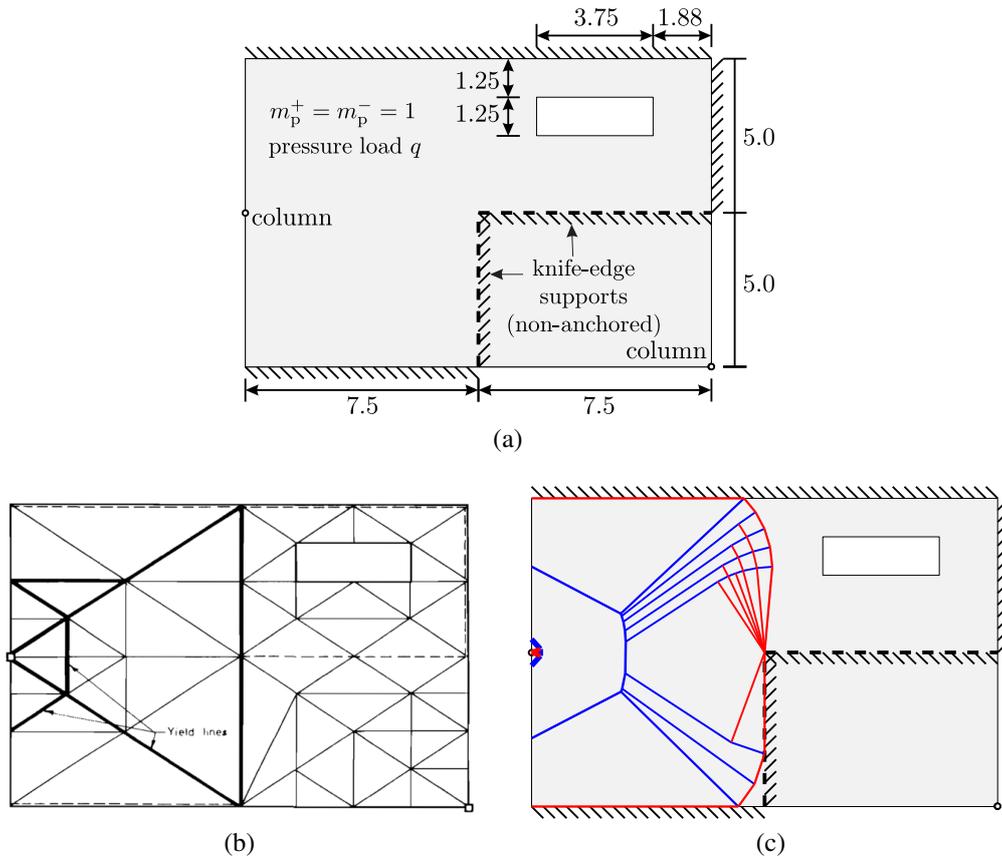


Figure 6.8: Roof slab: (a) problem specification; (b) yield-line pattern obtained by Munro & Da Fonseca (1978), $\lambda = 0.4$; (c) DLO yield-line pattern (rationalized), $\lambda = 0.36028$ (10,000 node, DLO solution, $\lambda = 0.35863$)

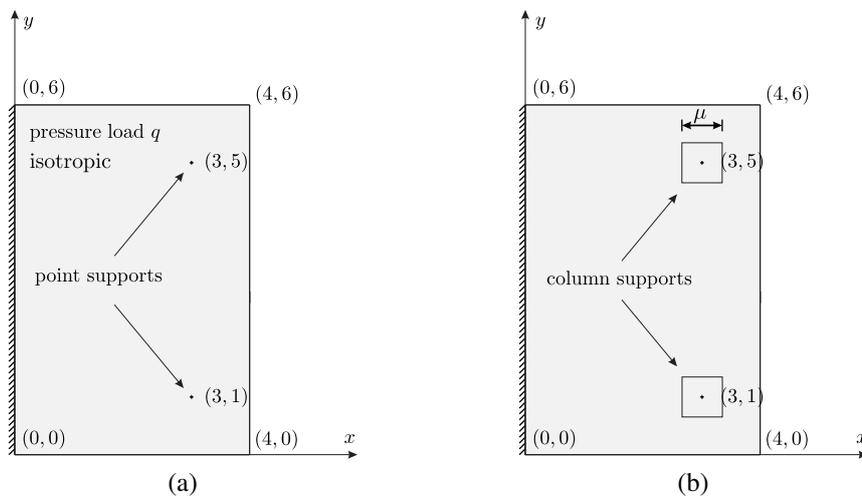
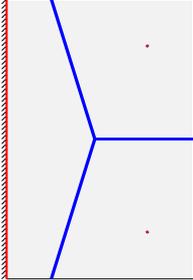
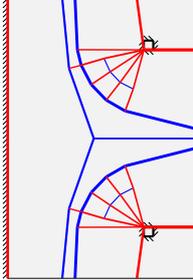
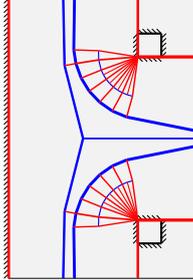
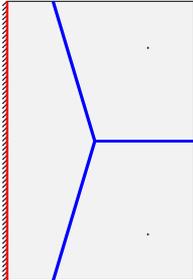
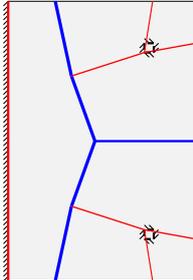
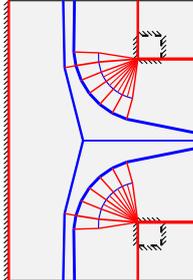
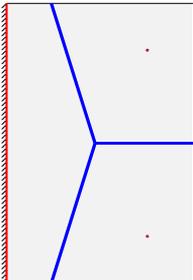
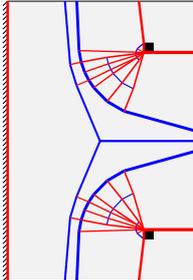
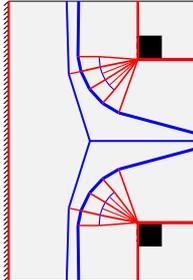


Figure 6.9: Johansen's slab with columns: (a) original point supports in Johansen (1943); (b) square column supports from Table 6.3

Table 6.4: Johansen’s slab with column supports: rationalized DLO yield-line patterns obtained by varying support type and size

Type of internal column	$\mu = 0.001$	$\mu = 0.2$	$\mu = 0.5$
Type I	 $m_p = 0.9583$	 $m_p = 0.8048$	 $m_p = 0.7019$
Type II	 $m_p = 0.9597$	 $m_p = 0.8144$	 $m_p = 0.7019$
Type III (support strength $i=0$)	 $m_p = 0.9587$	 $m_p = 0.8354$	 $m_p = 0.7668$

line pattern is largely insensitive to column type. However, local mechanisms are found close to Type I and III columns, and non-anchored (Type II) columns provide the best approximation of a point support.

- Secondly, when larger columns are used ($\mu = 0.2$), it is clear that the size of the column affect the failure mode significantly: a fan-type mechanism is developed near the Type I and III columns, and the uplift disappears. In contrast, uplift remains when Type II columns are involved, though the failure mechanism is different to that observed when a very small column size is present.
- Thirdly, when relatively large columns are used ($\mu = 0.5$), yield-lines develop along the column edges and no uplift is observed whichever of the three support types are used. In this case the same failure mechanism is observed irrespective of whether Type I or II columns are present.

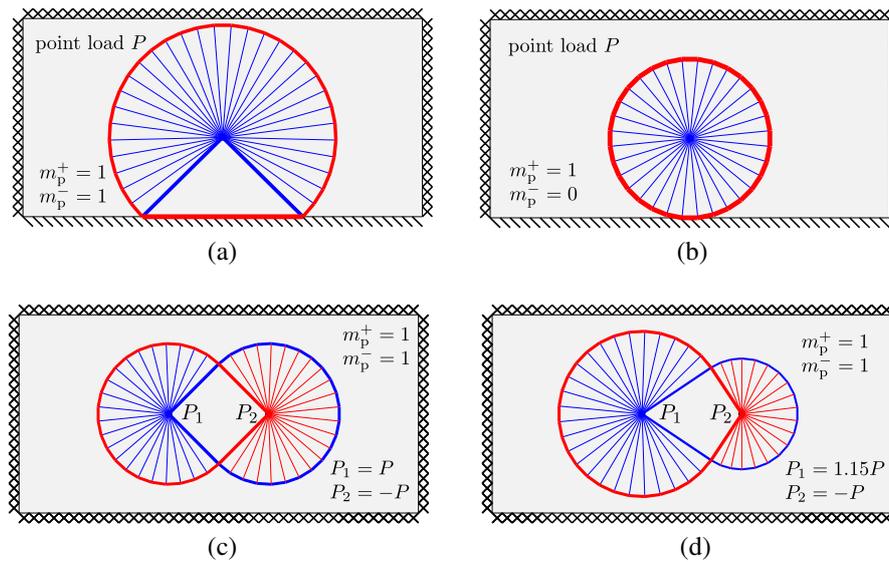


Figure 6.10: Point loaded slabs - DLO yield-line patterns for: (a) point load near boundary (configuration 1), $P = 11.443$; (b) point load near boundary (configuration 2), $P = 6.3034$; (c) slab with two point loads (configuration 1), $P = 11.458$; (d) slab with two point loads (configuration 2), $P = 10.515$ (see Table 6.5 for more accurate, 10,000 node, DLO solutions)

Slabs with point and line loads

Several slab problems involving point loads considered by Johansen (1943) are now revisited using DLO. The resulting yield-line patterns shown in Fig. 6.10 resemble closely the analytical solutions given by Johansen, with the margin of error being less than 1% (see also Table 6.5).

When line loads are present fan-type mechanisms will often develop in the vicinity of the load. Examples originally considered by Johansen (1943, 1968) are here analysed using DLO; the resulting yield-line patterns shown in Fig. 6.11 match Johansen's results closely, though the DLO solutions are more accurate.

Slabs with patch loads

The slab problem shown in Fig. 6.12 is taken from Ramsay & Johnson (1998); this involves self-weight and a patch load Q . Using geometry optimization, Ramsay & Johnson (1998) obtained a relatively accurate solution of $Q = 49.5\text{kN}$ (which is only 3.5% higher than 10,000 node DLO solution $Q = 47.85\text{kN}$). However the process used by Ramsay & Johnson (1998) was cumbersome in that it involved identifying a suitable yield-line pattern for use in the geometry optimization stage. In contrast the yield-line pattern shown in Fig. 6.12(c) was identified using DLO in a matter of seconds, without human-interaction.

Real-world slab

To further demonstrate the efficacy of DLO, the floor slab employed in a building (see Fig. 6.13) is now considered. The floor slab in question was also considered by Kennedy & Goodchild (2004),

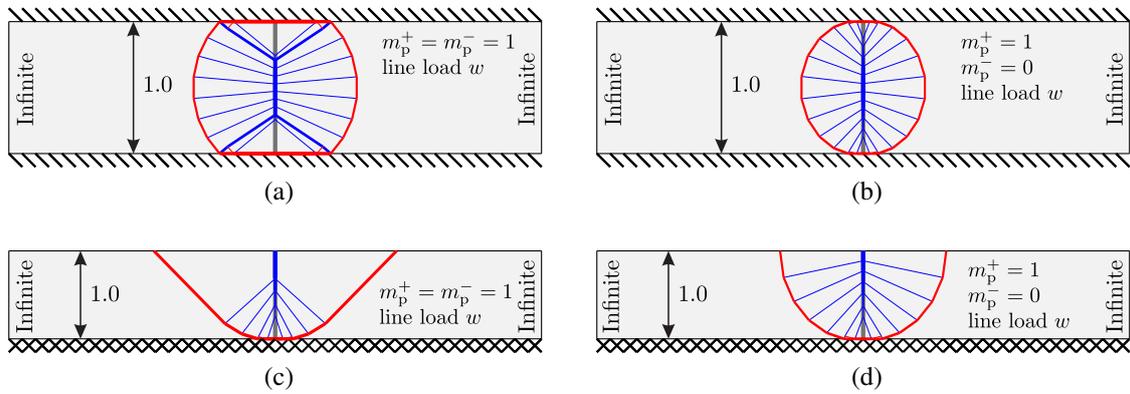


Figure 6.11: Slabs loaded with line loads - DLO yield-line patterns for: (a) slab loaded along a line (configuration 1), $w = 18.867$; (b) slab loaded along a line (configuration 2), $w = 11.693$; (c) cantilever slab loaded along a line (configuration 1), $w = 9.816$; (d) cantilever slab loaded along a line (configuration 2), $w = 5.8393$ (see Table 6.5 for more accurate, 10,000 node, DLO solutions)

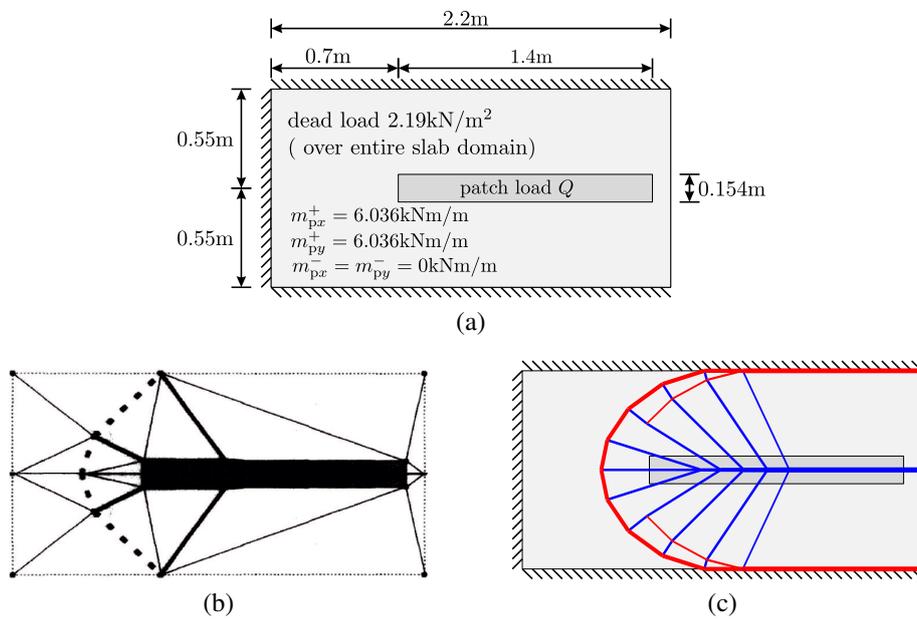


Figure 6.12: Slab with central patch load: (a) problem specification; (b) yield-line pattern obtained by Ramsay & Johnson (1998) using geometry optimization, $Q = 49.5 \text{ kN}$; (c) DLO yield-line pattern (rationalized), $Q = 48.05 \text{ kN}$ (10,000 node DLO $Q = 47.85 \text{ kN}$)



Figure 6.13: Real-world slab: a seven-storey block of flats in London

though for the purposes of this study full dimensions of the slab have been obtained from the original designers, Powell Tolner & Associates Ltd. This has allowed an accurate slab model to be created; full details are provided in Appendix F. A piecewise linear representation of the curved balconies is used and the walls and ‘blade’ columns are modelled using Type III internal columns. Following the lead of Kennedy & Goodchild (2004), here a support strength of $i = 1$ is used to represent the walls around the core, and $i = 0$ for the wall around the stairs. To model the 215mm wide blade columns, two scenarios are considered: (i) $i = 1$, and (ii) $i = 0$ for a more conservative design.

The rationalized yield-line patterns obtained using DLO are shown in Fig. 6.14. The first pattern matches very closely with the ‘folding plate’ mechanism assumed to be critical in Kennedy & Goodchild (2004). However, in the DLO solution fan-type mechanisms also develop around certain columns, leading to 2.8% increase in the required moment capacity (comparing $m_p = 48.6\text{kNm/m}$ from Kennedy & Goodchild (2004) to the 10,000 node DLO solution: $(49.9492/48.6 - 1) \times 100\% = 2.8\%$, assuming a uniform applied pressure load of 21.7kN/m^2). Although the hand calculation result described in Kennedy & Goodchild (2004) appears remarkably accurate, it is worth noting that the process involved manually analysing 11 potential yield-line patterns, each postulated by an experienced engineer. In contrast when using DLO the yield-line patterns are identified automatically. Regarding the second, more conservative scenario, taking $i = 0$ around the blade columns, the resulting yield-line pattern is similar, though there is now a 7.9% increase in the required moment capacity (comparing the Kennedy & Goodchild (2004) solution to the 10,000 node DLO solution: $(52.4439/48.6 - 1) \times 100\% = 7.9\%$).

Discussion

Characteristic features of yield-line patterns

Currently, an engineer analysing a slab using the yield-line method can draw upon well-established rules when postulating the critical yield-line pattern. As presented by Jones & Wood (1967), the

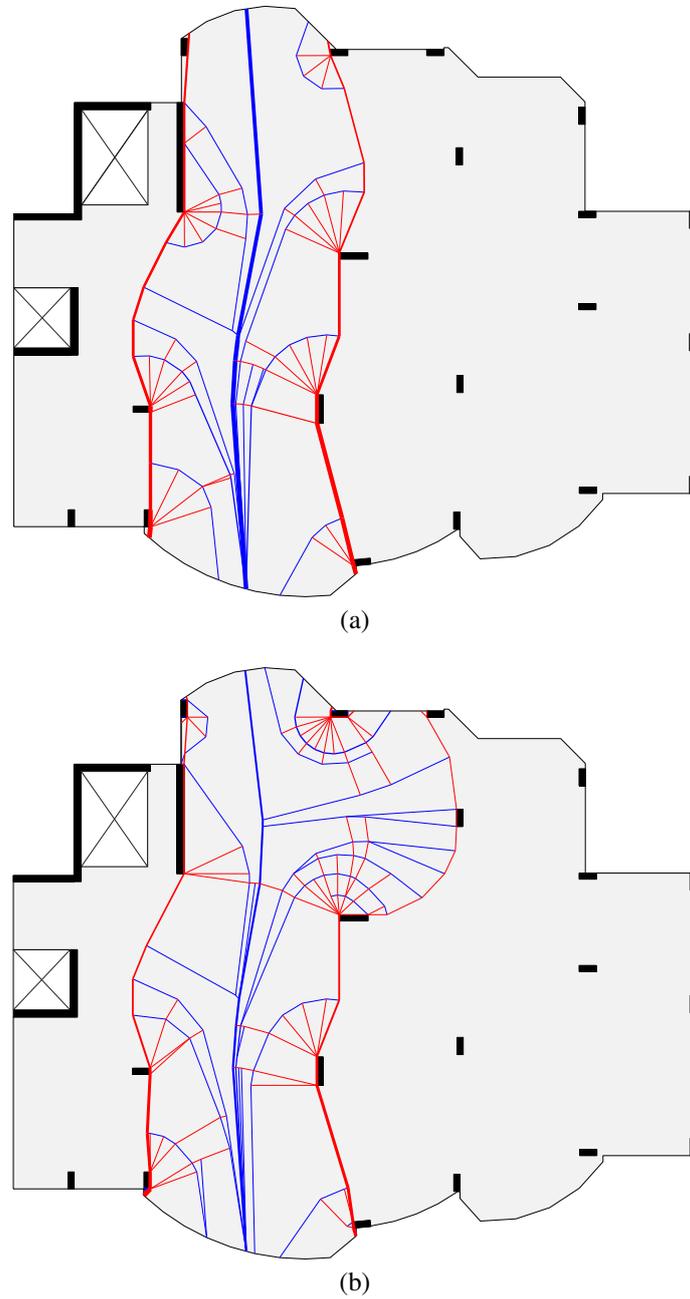


Figure 6.14: Real-world slab: rationalized yield-line patterns using DLO, (a) $i = 1$ for 215mm blade columns: $m_p = 49.3923\text{kNm/m}$ (10,000 node DLO $m_p = 49.9492\text{kNm/m}$); (b) $i = 0$ for 215mm blade columns: $m_p = 51.6699\text{kNm/m}$ (10,000 node DLO $m_p = 52.4439\text{kNm/m}$)

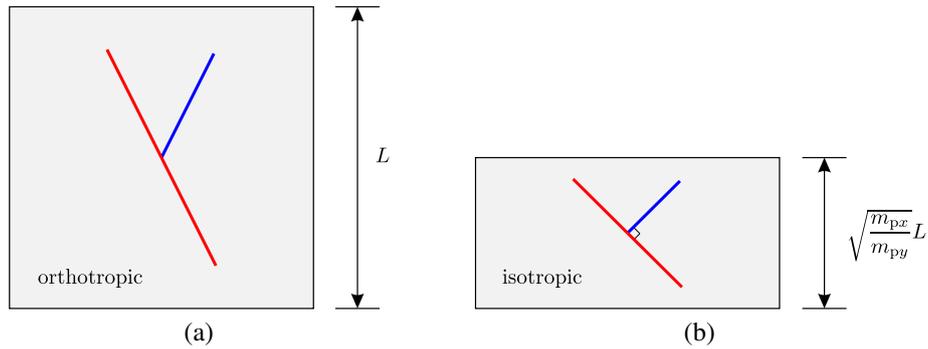


Figure 6.15: Determining characteristic features of a yield-line pattern in an orthotropic slab using an affine transformation: (a) original orthotropic slab; (b) equivalent isotropic slab derived using an affine transformation. The critical yield-line pattern in (b) will involve yield-lines of opposite sign intersecting at 90° ; the geometry of the yield-line pattern in (a) can be derived from (b) via geometrical transformation.

basic rules are:

- Yield-lines are straight except where a region may become completely plastic at any point.
- The yield-line between two or more bordering rigid regions must pass through the intersection of the axes of rotation of these regions.
- Axes of rotation usually lie along supported edges or column lines
- Yield-lines may only change direction when intersecting another yield-line.

Gilbert et al. (2014), used observed features in DLO solutions, and then Mohr's circle analysis, to point out additional characteristic features of yield-line patterns for isotropically reinforced slabs:

- Yield-lines of opposite signs should intersect at 90° , whether in the interior of a slab or at a fixed support.
- Yield-lines of opposite signs should intersect simple supports and free edges at between 45° and 135° .
- Yield-lines of the same sign can intersect at any angle.

However, in the context of the present paper it is of interest to point out that the above relations can be generalized for orthotropically reinforced slabs by using the affine transformation method (Johansen 1943, 1968, Kennedy & Goodchild 2004, Nielsen & Hoang 2011); an example is shown in Fig. 6.15.

Problems identifying critical yield-line patterns by other means

It is now clear that the DLO method provides a very efficient means of identifying critical yield-line patterns in the case of slabs with complex geometries, boundary conditions and/or load conditions. On the other hand it might be presumed that either hand-analysis or one of the previously

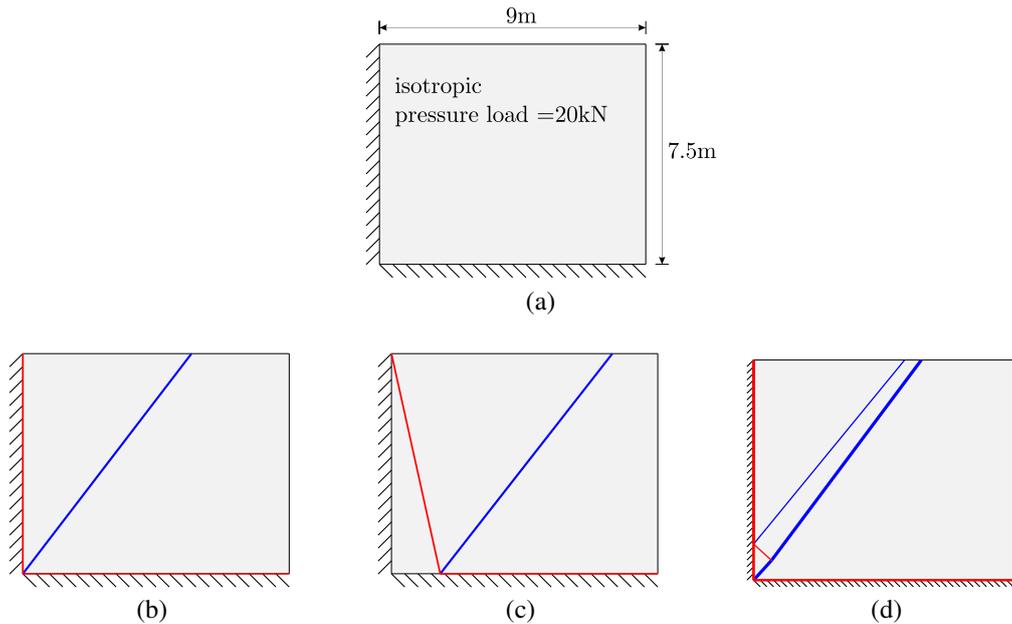


Figure 6.16: Rectangular slab simply supported on two adjacent edges: (a) problem specification; (b) simple yield-line pattern, $m_p = 256.5$ kNm/m; (c) alternative yield-line pattern, obtained by ‘optimization’, $m_p = 168.0$ kNm/m; (d) DLO yield-line pattern (rationalized), $m_p = 256.9$ kNm/m.

proposed numerical procedures can be expected to provide reasonable solutions for simple slab analysis problems. However, two simple example problems encountered during the course of the present study will be used to show that this is not necessarily the case.

Rectangular slab simply supported on two adjacent edges

When carrying out a yield-line analysis by hand, usually a yield-line pattern is postulated by hand (e.g. with the aid of the rules by Jones & Wood (1967)), and then the work method is used to obtain a solution. When using the work method various geometric parameters of a yield-line pattern can be adjusted to seek the minimum collapse load. For this reason, it might appear that the prescribed yield-line pattern need only be only ‘near’ correct. Whilst in many cases use of a simple postulated yield-line pattern can lead to a reasonably accurate solution, an issue is that it can be very difficult to judge whether a prescribed yield-line pattern is truly ‘near’ correct.

For example, consider a seemingly simple example problem provided in Kennedy & Goodchild (2004), a rectangular slab which is simply supported on two adjacent edges (Fig. 6.16(a)). Yield-line patterns provided in Johansen (1968) and Kennedy & Goodchild (2004) are shown in Fig. 6.16(b) and (c) respectively, whilst the yield-line pattern identified using DLO is shown in Fig. 6.16(d). Note that in the case of the pattern shown in Fig. 6.16(c), geometric parameters of the pattern have been adjusted to find the minimum collapse load. The pattern shown in Fig. 6.16(b) leads to a relatively accurate solution, very similar to that derived using DLO, shown in Fig. 6.16(d). However, the solution shown in Fig. 6.16(c) is clearly very inaccurate; the reason for this will therefore now be investigated further.

The geometry of the yield-line pattern given in Kennedy & Goodchild (2004) and shown in

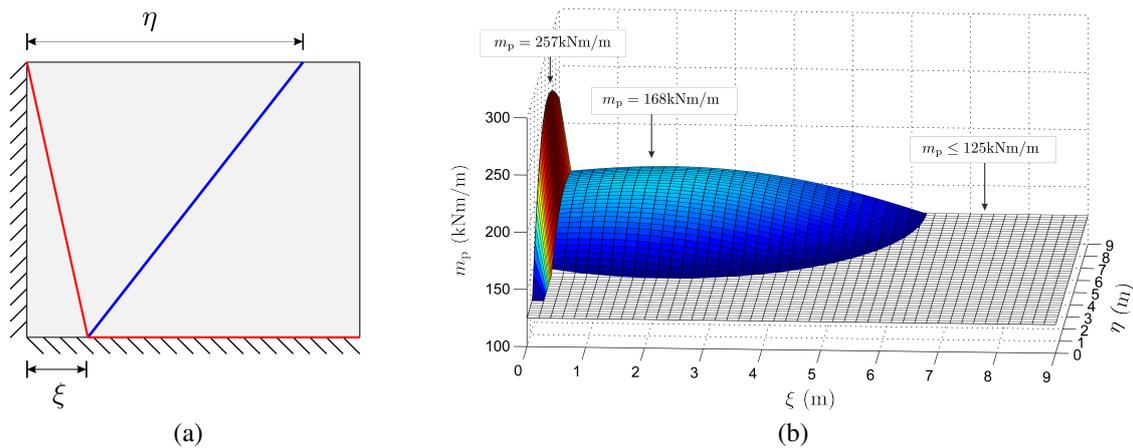


Figure 6.17: Rectangular slab simply supported on two edges: parametric study for the yield-line pattern proposed by Kennedy & Goodchild (2004): (a) relevant geometric parameters ξ , η ; (b) calculated required value of m_p by varying ξ and η .

Fig. 6.16(c) is controlled by two geometric parameters ξ and η , as shown in Fig. 6.17(a). The required moment capacity m_p can be obtained by varying these parameters, as shown in Fig. 6.17(b). It can be noticed that for $0 < \xi \leq 9$, the maximum value of m_p is 168.0kNm/m, which actually needs to be at least 52.9% higher for a safe design (calculated using $(256.9/168.0 - 1) \times 100\%$). However, it is evident in Fig. 6.17 that from $\xi > 0$ to $\xi = 0$, the m_p values rise suddenly, showing the presence of a singularity. This implies that pattern in Fig. 6.16(c) does not transform smoothly to that shown in Fig. 6.16(b) simply by optimizing its geometric parameters (e.g. using the optimization tool in Microsoft Excel). This means that the yield-line pattern shown in Fig. 6.16(c) cannot really be considered to be ‘near correct’. This clearly demonstrates that large errors can easily be encountered when using the yield-line method, even if the problem appears very simple. (Note that although the pattern shown in Fig. 6.16(c) was depicted in Kennedy & Goodchild (2004), the formula provided that document actually corresponds to the pattern shown in Fig. 6.16(b).)

L-shaped slab

The L-shaped slab in Fig. 6.18(a) was investigated in Ramsay & Johnson (1998), using a rigid finite element based numerical analysis procedure. A relatively coarse mesh was used, leading to the yield-line pattern shown in Fig. 6.18(b) being identified. Although this pattern may appear qualitatively reasonable, the problem was recently revisited by Ramsay et al. (2015), who showed that the yield-line pattern is actually far from critical. The corresponding DLO yield-line pattern for this problem is shown in Fig. 6.18(c), which corresponds to a required m_p value of 2.3743kNm/m. This is some 40% higher than the solution given in the 1998 paper (1.70kNm/m), showing that the latter solution was highly non-conservative. This indicates that, even for simple slab problems, the use of previously proposed numerical methods may lead to highly inaccurate solutions being obtained. It would also appear to show that it is difficult to qualitatively judge whether a given yield-line pattern is correct or otherwise (the highly inaccurate solution appears to have gone unnoticed in the literature for some 17 years, despite sustained interest in the field over

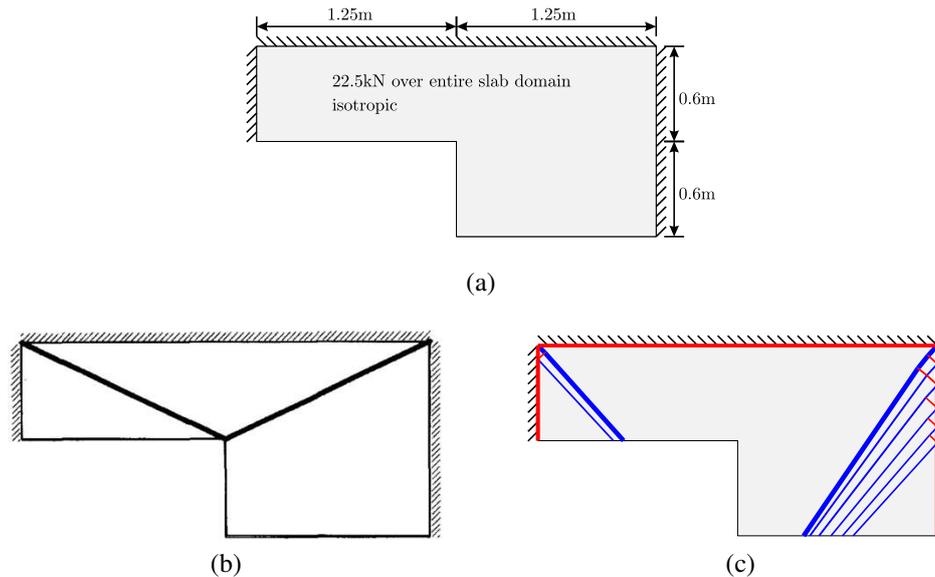


Figure 6.18: Ramsay's L-shaped slab simply-supported on three sides: (a) problem specification; (b) yield-line pattern obtained by Ramsay & Johnson (1998), $m_p = 1.70\text{kNm/m}$; (c) DLO yield-line pattern (rationalized), $m_p = 2.3743\text{kNm/m}$

that period). Fortunately the new DLO based automated yield-line analysis procedure overcomes accuracy issues associated with previously proposed numerical procedures (e.g. those utilizing rigid finite elements).

6.5 Conclusions

Discontinuity layout optimization (DLO) provides a powerful means of automating the yield-line method. Also, a given DLO solution, which can be complex in form, can be rationalized to aid visual interpretation and improve accuracy if required. In the present paper, it has been demonstrated that DLO can be applied to a wide variety of problems incorporating practical features (e.g. orthotropic reinforcement and a wide variety of support conditions and loading types). For all the example problems considered in the paper, DLO solutions have been found which are more accurate than those obtained using previously proposed upper bound numerical analysis techniques; in some cases this has shown that literature solutions are highly non-conservative.

6.6 Acknowledgements

The authors wish to thank Charles H. Goodchild from the UK Concrete Centre and John Sestak from Powell Tolner & Associates for providing the geometric data required for the analysis of the slab shown in Fig. 6.14.

Table 6.5: Summary of DLO solutions against those found in literature

Description	Literature		DLO			Deviation ^[a]
	Source	Load factor	$\leq 2,000$ nodes (Rationalized)	5,000 nodes	10,000 nodes	
Six-sided slab (configuration 1)	Wüst & Wagner (2008)	17.750	15.970	15.956	15.953	11.27%
Six-sided slab (configuration 2)	Wüst & Wagner (2008)	54.400	47.501	47.469	47.424	14.71%
Trapezoidal slab	Balasubramanyam & Kalyanaraman (1988)	0.084459 ^[b]	0.071928	0.071633	0.071566	18.02%
Roof slab	Bäcklund (1973)	0.38000	0.36028	0.35989	0.35863	5.96%
	Munro & Da Fonseca (1978)	0.40000				11.54%
	Balasubramanyam & Kalyanaraman (1988)	0.40200				12.09%
Johansen's slab with column support	Johansen (1943)	1.0411 ^[b]	1.0420 ^[c]	1.0420 ^[c]	1.0420 ^[c]	-0.08%
Point load near boundary (configuration 1)	Johansen (1943)	11.425 ^[d]	11.443	11.442	11.435	-0.09%
Point load near boundary (configuration 2)	Johansen (1943)	6.2832 ^[d]	6.3034	6.2990	6.2943	-0.18%
Slab with two point loads (configuration 1)	Johansen (1943)	11.420 ^[d]	11.458	11.452	11.448	-0.24%
Slab with two point loads (configuration 2)	Johansen (1943)	10.480 ^[d]	10.515	10.512	10.503	-0.22%
Slab loaded along a line (configuration 1)	Johansen (1943)	19.200	18.867	18.861	18.847	1.87%
Slab loaded along a line (configuration 2)	Johansen (1943)	12.700	11.693	11.672	11.646	9.05%
Cantilever slab loaded along a line (configuration 1)	Johansen (1968)	10.460	9.816	9.804	9.795	6.79%
Cantilever slab loaded along a line (configuration 2)	Johansen (1968)	6.3600	5.8393	5.8172	5.8086	9.49%
Ramsay's slab with patch load	Ramsay & Johnson (1998)	49.500	48.054	47.871	47.851	3.45%
Real-world slab ($i = 1$ for 215mm blade columns)	Kennedy & Goodchild (2004)	0.020576 ^[b]	0.020246	0.020083	0.020020	2.78%
Real-world slab ($i = 0$ for 215mm blade columns)	Kennedy & Goodchild (2004)	0.020576 ^[b]	0.019354	0.019138	0.019068	7.91%
Ramsay's L shape slab simply supported on three sides	Ramsay & Johnson (1998)	0.58824 ^[b]	0.42118	0.42125	0.42119	39.66%

a: Difference between literature and DLO 10,000 node solution (using DLO solution as benchmark)

b: Converted to a load factor by multiplying by $1/m_p$

c: Internal column support size $\mu = 0.001$ (Type II)

d: Exact solution

Chapter 7

Discussion

7.1 Introduction

Layout optimization provides a powerful method for identifying minimum weight truss structures. The technique has now also been successfully applied to the problem of automating the yield-line analysis method for slabs, using the analogy between optimum truss layouts and critical yield-line patterns. The frequently complex nature of the truss layouts and yield-line patterns identified by numerical layout optimization motivated the development of a rationalization technique involving the use of geometry optimization. A number of issues relating to the application of both layout and geometry optimization were encountered during the course of the research, and are discussed further in this chapter.

7.2 Issues in layout optimization

7.2.1 Use of a user-defined filter when extracting a layout

In the layout optimization process, a layout is typically ‘extracted’ using a user-defined filter. When geometry optimization rationalization is performed this layout will be used as the starting point of the process, and hence this needs to correspond as closely as possible to the original LP solution. Thus any user-defined filter, which will inevitably affect the extracted solution, must be called into question. In Chapter 4, a numerical validation process was proposed to address this issue; additional theoretical matters are discussed here. Whilst the discussion which follows focuses on trusses, since the user-defined filter affects truss and slab problems equally, the same basic principles can be applied to slab problems.

From a mathematical point of view, considering the truss layout optimization formulation (3.2), the filtering process involves seeking to identify active constraints in (3.2c), i.e., $\mathbf{q} = \mathbf{0}$. When using the interior point method, problems are encountered. Now, using (2.12), (3.2) can be reformulated as:

$$\min_{q,s} \quad \mathbf{c}^T \mathbf{q} - u \sum_{j=1}^m \ln s_j \quad (7.1a)$$

$$\text{s.t.} \quad \mathbf{B}\mathbf{q} = \mathbf{f} \quad (7.1b)$$

$$\mathbf{q} - \mathbf{s} = \mathbf{0}, \quad (7.1c)$$

where the barrier parameter u in (7.1) typically reaches a sufficiently small, *non-zero* positive number, and the slack variable s_j cannot equal zero. This means the inequality constraint (3.2c) is always ‘inactive’. For this reason, a tolerance number, or ‘filter’, is used to numerically determine the ‘active’ constraints. When the ground structure contains a considerable number of bars, setting an appropriate filter value is difficult in practice, so some structurally important bars may be erroneously filtered out due to their near-zero areas.

However, if the simplex method is used to solve (3.2), the active constraints can now be rigorously identified and the above issue no longer exists. It is significant to note that, for relatively large-scale LP problems, the simplex method is typically less efficient than the interior point method, so it is not recommended that computational efficiency is sacrificed to generate a clearer resulting layout. For small-scale problems, using a predefined filter becomes less problematic - qualified layouts can be extracted without significant difficulty - so the simplex method has no significant advantage over the interior point method. For this reason, the filter validation process in this thesis does not use the simplex method, but instead uses a simple but effective strategy that automatically adjusts the filter value until a ‘qualified’ layout is obtained.

7.2.2 Implementation issues in DLO

A number of issues surrounding the practical implementation of DLO, which have not been addressed in published papers (e.g., Gilbert et al. 2014), are of interest and hence are considered here.

Sign convention

Ensuring that the sign convention is consistent is perhaps the most challenging issue facing a developer of a DLO-based software program. Otherwise incorrect results, or perhaps unusual post-solve behaviour (e.g., when displaying deformed shape), are likely to occur, even when the yield-line pattern may appear correct. Also, when non-anchored boundary support types (Table 6.2) are involved, the correct uplift direction needs to be ensured. For these reasons, a systematic study of the sign convention has been undertaken; readers are referred to Appendix G.

Considering point and line loads

When a point load lies directly ‘above’ an end point of a yield-line, as shown in Fig. 7.1, it creates an ambiguous situation. Point load at C lies in the strip above yield-lines AB and BD. To ensure its

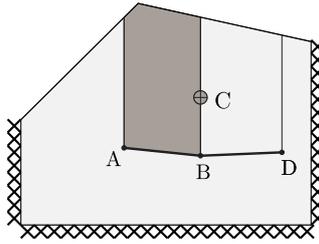


Figure 7.1: Ambiguous load effect by point load at point C

load effect is not erroneously considered twice, it is important that the point load is assumed to lie either in the strip above AB, or in the strip above BD, but not in both strips. This can pragmatically be achieved in a number of ways, for example by displacing the point very slightly left or right, or simply split the load. A similar situation can happen if there is a vertical line load.

7.2.3 The effectiveness of uniform nodal grids in DLO

As with many other numerical methods, the nodal discretization used in DLO affects the accuracy of the solutions which will be obtained. Nevertheless, a vast number of yield-line patterns are considered in a standard DLO analysis, and an evenly distributed nodal grid will normally suffice. In fact increasingly accurate solutions are generally obtained simply by increasing nodal density, without the need for arrange nodes in any particular pattern. For example, solutions within 1% of the exact value can often be obtained within a second or two on a desktop PC, using hundreds or, in some cases, a few thousand nodes. In theory the exact solution can be obtained when an infinite number of nodes is used (see also the extrapolation scheme documented in Appendix A). However, as a numerical method, DLO must use a finite number of nodes. Also, a limitation of using a uniform distribution of nodes is that a yield-line pattern which is small in relation to the grid cannot be accurately described.

A singular pattern near point load

In Johansen (1943), analytical studies showed that if a slab was loaded with a point load, a circular fan type failure pattern would be formed around the loaded point. Furthermore, the radius of the circle was shown not to affect the collapse load. However, if a uniform nodal grid is used in DLO, a mechanism involving a large radius will generally be the preferred solution. This is due to the fact that a larger circle will have more nodes around its perimeter, thus leading to a more accurate solution, e.g., see Fig. 7.2(a) vs. Fig. 7.2(b).

An interesting situation arises when a pressure load is applied in addition to a point load. If this acts in the same direction as the point load, the radius of the circle tends to be as large as possible, so as to maximise the work done by the pressure load. Conversely, if the pressure load acts in the opposite direction, the radius of the circle will tend to be infinitely small, leading to a singularity. In this situation using an evenly distributed nodal grid is problematic, for reasons which will be outlined: firstly, using DLO a relatively small circular mechanism will be identified, but since the number of nodes around the perimeter of the circle will be reduced, accuracy will also be

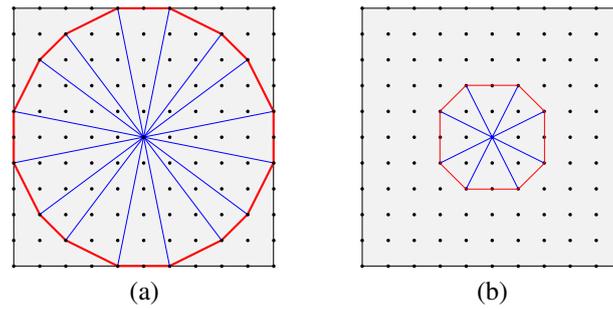


Figure 7.2: Grid influence on the size of a radius of fan around a point load: (a) large circle (more accurate); (b) small circle (less accurate)

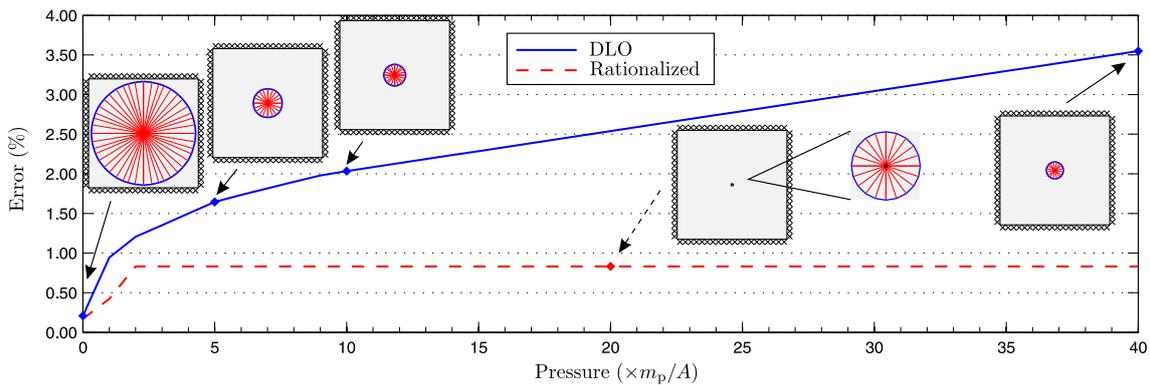


Figure 7.3: Slab with point (live) and pressure (dead) loads acting in different directions: parametric study by varying pressure load (DLO solutions indicated by solid line; rationalized DLO solutions indicated by dashed line; using 64×64 division nodal grid)

reduced (for example, the mechanism shown in Fig. 7.2(b) will be favoured relative to that shown in Fig. 7.2(a)). Secondly, in a nodal grid, there is a finite spacing between nodes which prevents the radius of the circle from becoming infinitely small. Due to these constraints, DLO may fail to generate highly accurate solutions even when a relatively dense nodal grid is used. To investigate this problem, a parametric study was conducted, showing the influence of pressure on the accuracy of the solutions. Note that if the pressure is defined as a live load, and hence is subject to the load factor, the failure mode changes when the pressure becomes sufficiently large, and the singular yield-line pattern is no longer critical. Therefore, to prevent this from occurring the pressure was defined as a dead load in this study.

It can be observed from Fig. 7.3 that the accuracy of DLO (solid-line) decreases as the magnitude of the pressure is increased, going through two distinct phases. In the first phase, the radius of the circle in the yield-line pattern decreases rapidly, reducing the number of nodes lying on its perimeter (i.e., the first issue), so that numerical accuracy decreases rapidly. In the second phase the radius of the circle decreases more slowly, so that the impact of the first issue does not change dramatically; however, the second issue then affects numerical accuracy. Similarly, results obtained using the geometry optimization rationalization technique clearly reflect these two phases. Since the process requires a starting pattern from a standard DLO analysis, the rationalized solution is also affected by the first issue. However, radius of the circle can be reduced via geometry optimization, effectively addressing the second issue. For this reason, the error does not increase significantly in the second phase.

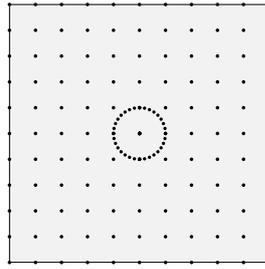


Figure 7.4: Circular grid to address infinitely small failure circle

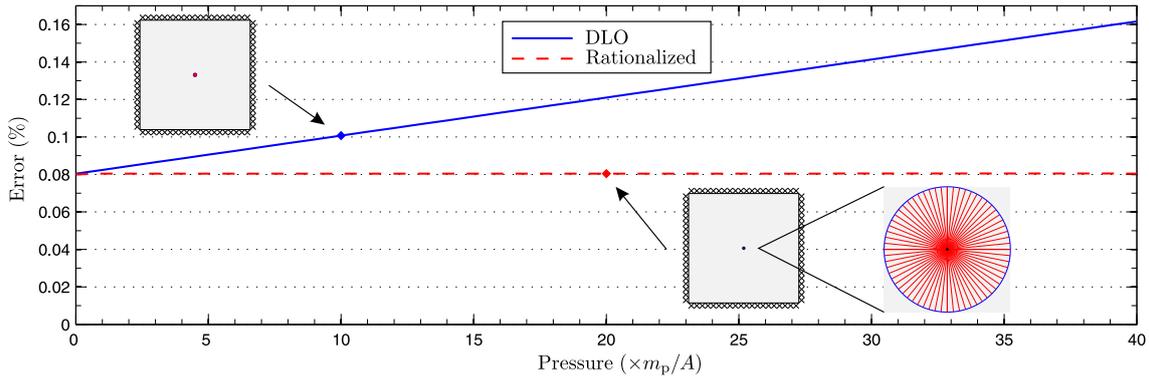


Figure 7.5: Slab with point (live) and pressure (dead) loads acting in different directions: using an extra circular grid with 65 nodes along the perimeter

The parametric study shows a limitation of using an evenly distributed nodal grid in DLO. A simple approach to address this limitation is to create a small circular grid around a point load, in addition to the standard grid, to permit a singular failure circle to be identified with relatively high accuracy. For example, a circular grid with a radius equal to the nodal spacing in the standard nodal grid is used in Fig. 7.4. A ‘singular failure circle’ can be readily formed using this circular grid, which has a number of nodes along its perimeter; this effectively addresses the first issue. Furthermore, if the radius of the circular grid is sufficiently small, the impact of the second issue is negligible. Note that the circular grid is created to identify a potential singular failure circle, so the extra nodes in its perimeter do not need connections to other existing nodes in a standard grid. Thus, it has negligible impact on the computational cost even if a relatively dense circular grid is employed. Using this approach, the same parametric study was conducted in Fig. 7.5 for comparison. Highly accurate solutions were produced. Due to the second issue, the accuracy of DLO (solid-line) still decreased slowly with increasing pressure; however, the accuracy of the rationalized solution (dashed-line) was completely unaffected.

7.3 Issues in geometry optimization

One major difference between layout and geometry optimization is that the supplied data (e.g., matrices in the truss layout optimization formulation of (3.2)) remain ‘static’ in the former, so the process to obtain those data does not need explicitly defined formulae. In the latter, functions are ‘dynamically’ updating with respect to nodal positions, requiring mathematical expressions to be defined explicitly. This means that some design considerations which appear simple in layout

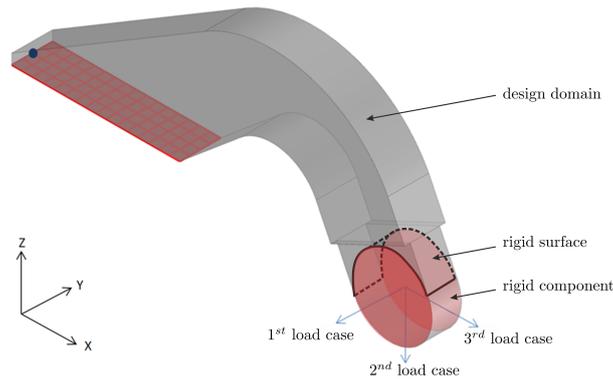


Figure 7.6: A truss design problem studied by Smith et al. (2015)

optimization may turn out to be relatively challenging in the geometry optimization formulation. For this reason, limitations affecting the latter are now considered.

7.3.1 Trusses in engineering practice

Chapter 4 is mainly concerned with the theoretical developments of the rationalization technique; for this reason some practical issues are not addressed. In engineering practice, truss problems can be more complicated; for example, consider the design problem shown in Fig. 7.6.

This example raises two questions: how can nodes be restricted to lie within an irregular 3D design domain, and how can a rigid component be modelled. For the first question, noting that 2D domain constraints have been introduced in Chapter 4, a similar approach can be employed here: given mesh layouts of the design domain, plane equations of every mesh can be derived, with the plane equations replacing the line equations in (4.16). Given a convex 3D domain, this approach is straightforward; however, for non-convex cases, other issues exist that will be discussed in Section 7.3.2.

Regarding the second question, the precise distribution of internal forces from the truss structure onto the rigid component, or, more specifically, onto a surface of the rigid component, will be unknown in advance. To model these forces, one approach is to idealise them as a set of point loads which are distributed across the rigid surface using the following steps: first, create nodes on the rigid surface and apply point loads to them, permitting the direction and magnitude of each point load to be varied (i.e., using optimization variables); second, impose restrictions on these nodes so that they can only move along the rigid surface; finally, ensure these loads are in equilibrium with any external loads applied to the rigid component.

7.3.2 Restricting nodes and line connections in non-convex domains

Geometry optimization involving non-convex design domains can be challenging. Chapters 5 and 6 address this by converting a concave region into a combination of a convex regions using domain decomposition. However, this approach has a disadvantage; solution accuracy is potentially decreased. In this section, issues associated with the use of non-convex domains are revisited and

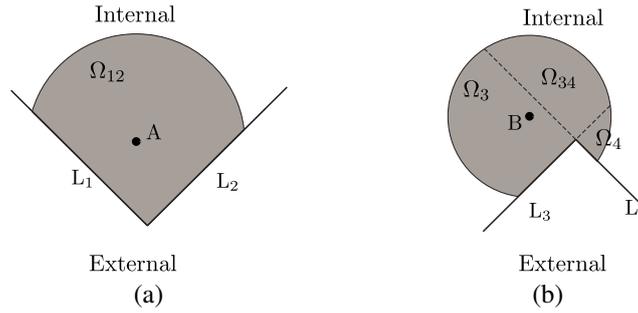


Figure 7.7: Domain constraints for (a) convex domain and (b) concave domain (shaded area indicates permitted regions for node A and B)

alternative approaches are proposed.

Design domain constraints for nodes

It has been shown in Chapter 4 that, to restrict a node within a convex design domain, the line constraints of all boundary lines are imposed using (4.16). For the example shown in Fig. 7.7(a), assume lines L_1 and L_2 and node A, the restriction is:

$$\mathbf{T}_1^L \boldsymbol{\nu}_A \geq 0, \quad (7.2a)$$

$$\mathbf{T}_2^L \boldsymbol{\nu}_A \geq 0. \quad (7.2b)$$

However, for the example shown in Fig. 7.7(b), imposing similar constraints causes problems: simultaneously imposing line constraints L_3 and L_4 means that node B can only lie in region Ω_{34} ; since B is currently in Ω_3 , these constraints lead to an infeasible state. In this case, the line constraint L_4 must be removed. The domain restriction becomes: a node cannot simultaneously violate all line constraints - or equivalently, at least one line constraint must be satisfied. This situation is very similar to active and inactive constraints in optimality conditions (Section 2.4.1), so a similar approach is potentially useful. Assuming non-negative complementary variables φ_3 and φ_4 , the domain constraints imposed on node B are written as:

$$\varphi_3 \mathbf{T}_3^L \boldsymbol{\nu}_B \geq 0, \quad (7.3a)$$

$$\varphi_4 \mathbf{T}_4^L \boldsymbol{\nu}_B \geq 0, \quad (7.3b)$$

$$\varphi_3 + \varphi_4 > 0, \quad (7.3c)$$

$$\varphi_3 \geq 0, \quad (7.3d)$$

$$\varphi_4 \geq 0. \quad (7.3e)$$

A line constraint is ‘activated’ when its associated complementary variable is non-zero. In addition, constraint (7.3c) states that at least one line constraint must be activated. Note that complementary variables φ_3 and φ_4 are extra optimization variables, so that constraints (7.3a) and (7.3b) are non-linear.

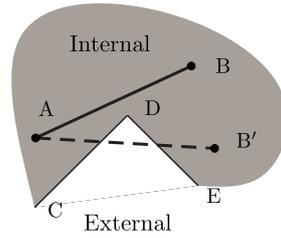


Figure 7.8: Line collides with concave domain

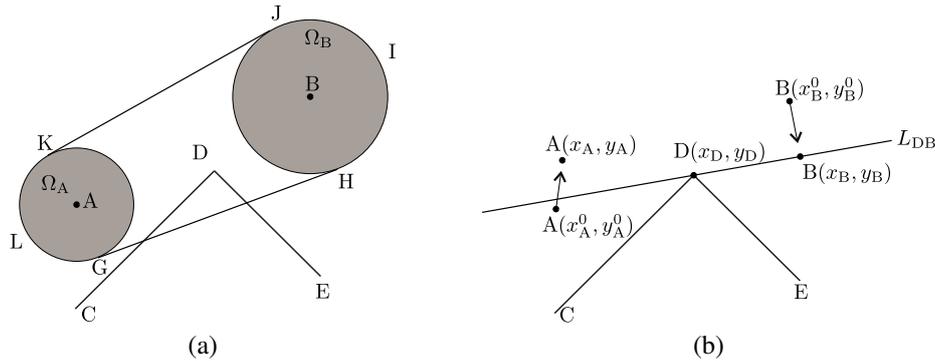


Figure 7.9: Preventing line collision with concave boundary: (a) potential collision if D lies in geometry enclosed by lines GH and KJ, and curves HIJ and KLG; (b) imposing additional line constraint to node A

Alternatively, a simple approach is to only impose the valid constraint (e.g., apply only constraint L_3 in Fig. 7.7(b)) without using (7.3). In this case, a smaller feasible region is assumed (e.g., in the case shown in Fig. 7.7(b), Ω_4 is now excluded). Nevertheless, there exists a node on a vertex in a slab problem that imposes move limits (i.e., constraint (4.13)). For this reason, node B cannot be moved to Ω_4 . Considering this, this simple approach was employed in Chapters 5 and 6 instead of using constraint (7.3). On the other hand, given a truss design problem, especially in a 3D domain, a node may not exist on the vertex, so constraint (7.3) can be imposed.

Moving lines near a concave region

Lines can potentially collide with a concave region without their nodes moving out of the design domain; e.g., see Fig. 7.8. Domain decomposition was used in Chapters 5 and 6 to tackle this issue. Though it is perhaps the easiest approach, it can also affect the solution; here, the potential to address this issue without domain decomposition is discussed.

Since nodes have move limits, it is convenient to check whether a collision can occur. As shown in Fig. 7.9(a), assume move limits Ω_A and Ω_B for nodes A and B, respectively. Vertex D of the concave boundary lies within the geometry formed by lines GH and KJ and curves HIJ and KLG, so AB could potentially collide with boundaries CD and DE. Movement of one node will impose an additional move limit on the other to prevent collision. Fig. 7.9(b) illustrates this restriction. Assume $B(x_B^0, y_B^0)$ moves to $B(x_B, y_B)$; $A(x_A, y_A)$ must then satisfy $\mathbf{T}_{DB}^L \boldsymbol{\nu}_A \geq 0$, where \mathbf{T}_{DB}^L is the coefficient vector of line L_{DB} , constructed by letting its normal lie outside cone CDE. \mathbf{T}_{DB}^L contains optimization variables x_B and y_B , and constraint $\mathbf{T}_{DB}^L \boldsymbol{\nu}_A \geq 0$ is now non-linear.

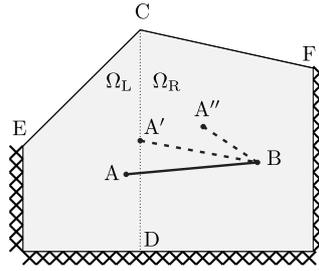


Figure 7.10: Regions divided by non-smooth point of the top edge

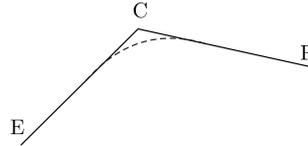


Figure 7.11: Approximating a non-smooth edge using a curved line

7.3.3 Non-smooth issues in geometry optimization for slabs

As stated in Chapter 5, the live load effect f_L in constraint (5.1d) can be non-smooth or even discontinuous with respect to nodal positions. This issue is further investigated here.

Non-smooth ‘top’ edge

It has been shown that the smoothness of the ‘top’ edge of a slab problem is important. For this reason the live load effect may not be differentiable everywhere with respect to nodal coordinates. For the example shown in Fig. 7.10, the top edge ECF has a non-smooth point C; let f_L^{AB} represent the live load effect on yield-line AB. To permit node A to move across the entire domain, it requires that f_L^{AB} be smooth everywhere with respect to x_A . It can be observed that at any point of line CD, for example at A' , this condition is not satisfied. Using the approach shown in Chapter 5, the domain is divided into two parts (Ω_L and Ω_R) via a vertical line CD. To ensure f_L^{AB} is smooth A must be restricted in Ω_L , which potentially affects the solution. To permit A to move in both regions, two alternative approaches are now discussed.

The first approach involves geometric approximation, where the non-smooth regions are approximated using curved lines; an example is illustrated in Fig. 7.11. Since curved lines are used, the original live load effect terms and their derivatives are modified. The curved lines cannot be uniquely determined using the top edges; therefore, the optimization program also needs to control certain characteristics of those lines, for example their curvature, which means that the level of approximation can be adjusted.

The second approach is to impose complementary conditions to automatically take directional derivatives at a non-smooth point. Now assume $A(x_A, y_A)$, $C(x_C, y_C)$, and a non-negative complementary variable φ ; in addition, let f_L^{ABL} denote the live load effects calculated in region Ω_L and f_L^{ABR} in region Ω_R . Note that f_L^{AB} can be written as a linear combination of the two:

$$f_L^{AB} = \varphi f_L^{ABL} + (1 - \varphi) f_L^{ABR}. \quad (7.4)$$

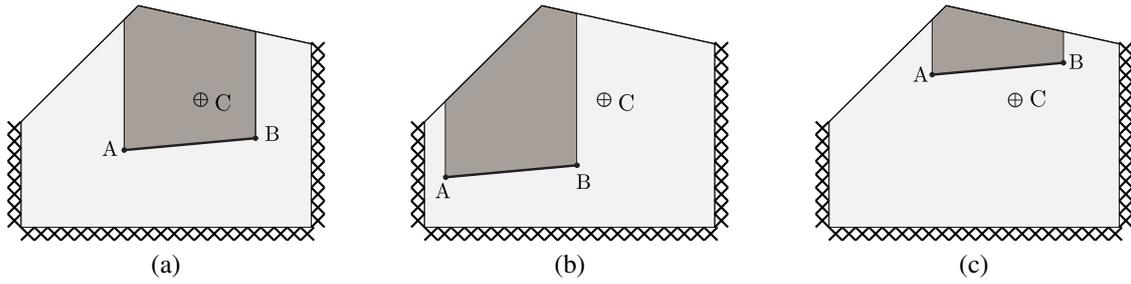


Figure 7.12: Situation when point load at C is taken on yield line AB: (a) taken; (b) and (c) not taken.

To determine the complementary variable, the following constraints are added:

$$\varphi(x_A - x_C) \geq 0, \quad (7.5a)$$

$$(1 - \varphi)(x_C - x_A) \geq 0, \quad (7.5b)$$

$$\varphi(1 - \varphi) = 0, \quad (7.5c)$$

φ must equal ‘1’ when node A lies in region Ω_L , and ‘0’ when in Ω_R . It provides a way to account for directional derivatives in the optimization. However, the equality constraint (7.5c) may cause issues due to the fact that the feasible set of φ consists of only two discrete values: ‘0’ and ‘1’. To address this, a tolerance parameter ϵ ($\epsilon \geq 0$) is now introduced, so (7.5) is modified as:

$$\varphi(x_A - x_C) + \epsilon \geq 0 \quad (7.6a)$$

$$(1 - \varphi)(x_C - x_A) + \epsilon \geq 0 \quad (7.6b)$$

$$0 \leq \varphi \leq 1 \quad (7.6c)$$

which is an approximation of (7.5). Given a sufficiently large ϵ , then φ is not enforced to be ‘0’ or ‘1’. Inspired by the solution strategy used in the interior point method, the following approach is proposed here: at the start of the optimization, φ can have a relatively high value to permit nodes to be moved throughout the entire domain; its value can then be reduced when nodes are near their optimum locations, so nodes may not move large distances, and the non-smooth issue has a reduced impact on the solutions obtained.

Incorporating point and line loads

It has been stated in Chapter 6 that point and line loads cause non-smooth issues in geometry optimization. An example is shown in Fig. 7.12, where a point load is applied on point C. For a yield-line AB, the point load contributes to the live load effect only if it lies in the strip above the yield-line (e.g., Fig. 7.12(a)).

In geometry optimization, no mathematical information has currently been provided to ensure that AB lies below C. By continuously moving nodes A and/or B, the load effect can suddenly disappear, implying that f_L is potentially discontinuous. Fortunately, in examples shown in Chapter 6, this behaviour did not occur so f_L remained smooth for the entire optimization process. However, additional steps must be taken to tackle this issue in the case of general slab problems. One ap-

proach is to prevent the load effect from disappearing and appearing suddenly; e.g., if a yield-line is below a point load, then movements of the end nodes can be restricted so that the point load always lies above this yield-line.

However, restricting movements can impact the solution. Unfortunately, for the time being there are no clear alternative approaches for addressing this issue. A similar issue arises when line loads are involved. Fortunately the examples shown in Chapter 6 were relatively simple, such that this issue did not arise.

7.4 Computational aspects in layout optimization

Layout optimization involves a fixed grid of nodes, so its solution relies highly upon the chosen numerical discretization. Typically, employing a denser nodal grid will produce a more efficient truss structure (or more critical yield-line pattern in the case of a slab problem). However, when increasing the number of nodes employed, the size of its LP problem grows much more rapidly. Though efficient, an LP solver can experience difficulties due to enormous problem size. This highlights the need to develop efficient solution schemes; one has been given by Gilbert & Tyas (2003): the member adding scheme.

7.4.1 Mathematical aspects of the member adding scheme

Instead of directly attacking the primal truss layout optimization directly (3.2), and hence also problem (2.5), the member adding scheme involves also referring to the dual problem. Using problem (2.6), the dual of (3.2) can be written as:

$$\max_{\mathbf{u}} \quad W = \mathbf{f}^T \mathbf{u} \quad (7.7a)$$

$$\text{s.t.} \quad \mathbf{B}^T \mathbf{u} \leq \mathbf{c}, \quad (7.7b)$$

where W is virtual work, and \mathbf{u} is a virtual displacement vector of all nodes. It is significant to investigate problem (7.7) because: (i) LP problems have strong duality (Section 2.2.4) such that solution to (7.7) is also the solution to (3.2); (ii) using ground structure, the size of \mathbf{u} is significantly smaller than that of \mathbf{q} ; and (iii) \mathbf{u} is obtained automatically, without further computation, after solving the primal problem (3.2) using a modern LP solver.

The inequality constraint (7.7b) describes the maximum axial deformation permitted in every bar. Also note that for a bar (i.e., member) to be present in the resulting layout, its corresponding constraint in (7.7b) must be active (see Section 2.4.1). Similarly, for a bar not present in the resulting layout, its corresponding constraint is likely to be inactive. Given the optimality conditions (Section 2.4.1) it is clear that eliminating the inactive constraints in (7.7b) does not affect the objective value (i.e., the maximum virtual work in this case). This also means that the corresponding variables (and hence bars) in the primal problem (3.2) can be removed, leading to a smaller LP problem without losing accuracy. An iterative solution strategy could be employed: initially a very small number of bars are included in the ground structure, resulting in a small, but potentially inaccurate, primal problem (3.2). The error associated with the reduced problem can then be

detected by examining constraint (7.7b) for all bars: if any violations exist, these constraints can be imposed by adding the corresponding bars to (3.2); if there are no violations, the constraints associated with the ‘non-existing’ bars must truly be inactive, and the optimum solution must have been found.

7.4.2 On potential improvements to the member adding scheme

The member adding scheme (Gilbert & Tyas 2003) permits relatively large LP problems to be tackled. However, the computational cost can still be high when problems are very large. In the current research, it has been observed that a primal problem typically contains far more connections than are present in the ‘extracted’ layout. For example, the Hemp cantilever shown in Fig. 4.6(a) was investigated. Table 7.1 shows a comparison of the numbers of bars present in the full, the primal (in the last iteration), and the extracted problem for increasing numbers of nodal divisions. Though only relatively few bars are present in the extracted structures, it is clear that the difference in the volume of the extracted solution compared with the volume obtained from the primal problem (at the last iteration) is small. This phenomenon shows that a considerable proportion of the bars in the primal problem are not structurally important, suggesting that better strategies could be developed to improve computational efficiency.

Table 7.1: Hemp cantilever in Fig. 4.6(a): comparing no. of bars in full, primal, and extracted problems

Full problem		Primal problem (the last iteration)		Extracted problem		
Nodal divs	No. of bars	No. of bars	Percentage [†] (%)	No. of bars	Percentage [†] (%)	Volume difference [‡] (%)
30×15	74655	2878	3.86	163	0.218	3×10^{-8}
60×30	892702	19033	2.13	605	0.068	2×10^{-8}
90×45	3149297	63995	2.03	1480	0.047	2×10^{-7}
120×60	7004968	156839	2.24	2519	0.036	4×10^{-7}
150×75	12456601	295175	2.37	4244	0.034	5×10^{-5}

[†] Relative to no. of bars in the full problem.

[‡] Relative to volume computed from full and primal problem (which are the same).

7.4.3 Estimating duality gap to bracket solution

In the member adding scheme, an LP problem is solved iteratively. In the last few iterations the reduction of the objective function is negligible. An example is shown in Table 7.2. At the ninth iteration, the margin of error has been reduced to $1 \times 10^{-7}\%$; however, the elapsed time at this point is only 52.8% of the total time. Even though violated constraints exist in the dual problem (7.7), the duality gap (see Section 2.1) is extremely small, so an early termination is possible. Therefore, evaluating the duality gap in each iteration can be of particular interest when developing efficient solution schemes that yields results within given margin of error. However, in the standard member adding scheme the dual problem (7.7) is not solved, and only its constraints (7.7b) are examined. This means its objective value is unknown and the duality gap is not readily obtainable. However, methods can be developed to provide an upper bound estimate of the bound gap, with a view to making the member adding scheme more efficient.

Table 7.2: Hemp cantilever in Fig. 4.8(a): iterative solutions via member adding scheme

Iteration	Volume difference [†] (%)	No. of bars	CPU time (s)	Progress [‡] (%)
1	15.6	45225	11	0.2
2	4.2	132206	22	4.3
3	1.4	180850	29	8.0
4	0.7	211540	42	13.4
5	0.3	242153	57	20.7
6	0.07	272690	66	29.1
7	0.003	293444	65	37.4
8	6×10^{-5}	295022	60	45.1
9	1×10^{-7}	295146	60	52.8
10	3×10^{-8}	295163	63	60.9
11	2×10^{-9}	295166	64	69.1
12	2×10^{-9}	295170	61	77.0
13	2×10^{-9}	295172	61	84.9
14	2×10^{-9}	295174	60	92.6
15	0	295175	58	100

[†]: Compare with volume reported in the last iteration

[‡]: Measured using CPU time

7.4.4 Investigating the heuristics

Though rigorous solutions are ensured in the member adding scheme, heuristics are involved in identifying the members (and hence variables) to be added to the primal problem. The original contribution by Gilbert & Tyas (2003) used a set of fixed parameters to control this heuristic process, while others have proposed modified schemes (e.g., Pritchard 2004 and Sokół 2011). Perhaps because of its heuristic character, a systematic study of member adding schemes appears to be lacking. Also, existing strategies appear not to have taken into account many mathematical characteristics that may be useful. For example, the aforementioned duality gap can be used as a performance index, and perhaps a better strategy could be developed to identify the most cost-efficient way of adding variables to minimize the duality gap.

7.4.5 Parallel processing and decomposition

In the past few decades there has been rapid development of single core CPUs, such that merely upgrading a CPU can automatically eliminate many computational obstacles (Sutter & Larus 2005). However, this trend has changed in recent years (Kirk & Hwu 2010); now single core CPU speed is no longer improving rapidly, and more emphasis has been placed on parallel processing. This implies that, solving a problem in parallel may receive significant benefits. The use of the member adding scheme is amenable to parallel processing as checking that constraints are satisfied (e.g., (7.7b)) can be performed in parallel, potentially using GPU acceleration^[i]. However, the primal problem (e.g., (3.2)) is solved using existing LP software packages. Though improvements have been made by the developers of LP software, numerical experiments conducted by the author suggest that better parallel strategies are needed.

One popular topic involving parallel processing is domain decomposition. In the context of the finite element method, sub-domains can normally be created by grouping meshes into regions;

^[i]GPU acceleration is a technique often used in parallel processing. Unlike CPUs, a modern GPU typically has a large amount of cores, however, each core can only handle relatively simple operations.

however, difficulties have been found in layout optimization. Dividing the domain will inevitably split some connections (e.g., bars in a truss), so steps need be taken. One option is to remove these connections, but this is likely to affect solution accuracy. Another option is to split any connections that intersect a sub-domain boundary. This involves creating extra nodes on the sub-domain boundary, which may result in a large number of nodes being created and can cause other issues. Here, domain decomposition is considered in conjunction with the member adding scheme: it involves solving sub-problems in parallel to determine the nodes and connections which need to be added to the master problem. An example is shown in Fig. 7.13; the steps involved are listed below:

Framework of the domain decomposition technique	
Single	(a) Initially, a master problem containing a relatively small number of nodes is considered.
	(b) Create sub-problems by transferring the layout of the master problem and then employing dense nodal grids in certain regions, depending on locations of the sub-domains.
Parallel	(c) The sub-problems are then solved in parallel. (d) Extract nodes and connections from sub-problems
	(e) Merge some or all nodes and connections in the sub-problems to the master problem
Single	(f) Perform a layout optimization for the master problem. (g) Extract nodes and connections from the master problem. (h) Estimate duality gap: if negligible then the whole process terminates, otherwise go to (b).

Some advantages can be envisioned: first, updating sub-problems normally involves only minor modifications to the previously analysed problems, so setting up a new sub-problem is inexpensive except for the first iteration; second, results obtained in a sub-problem can be used as a basis to provide a ‘warm-start’ for the new sub-problem in the following iteration; finally, the master problem can be processed without waiting for all sub-problems to be solved as long as it successfully reads in new nodes and connections from the solved sub-problems.

It is important to ensure that the main problem is kept as simple as possible. For this reason, nodes and connections in the master problem can be removed. For example, in Fig. 7.13 certain nodes in (a2) have not been included in any of the solved sub-problems; they are then removed in (c2).

Since this parallel technique has not been implemented, its efficacy is unknown. However, it may be expected that some issues may exist: firstly, a means of estimating the duality gap is not currently available; secondly, sub-problems are completely separated - the only ‘communication’ between sub-problems is when the master problem has progressed to the next iteration, in which case a sub-domain can have some nodes initially belonging to others; thirdly, to progress the master problem, it is important to ensure that at least one sub-problem identifies nodes (and connections) that can be added.

7.5 Computational aspects in geometry optimization

As stated in Chapter 4, geometry optimization is comparatively computationally expensive. Some potential strategies for improving its computational efficiency are now discussed.

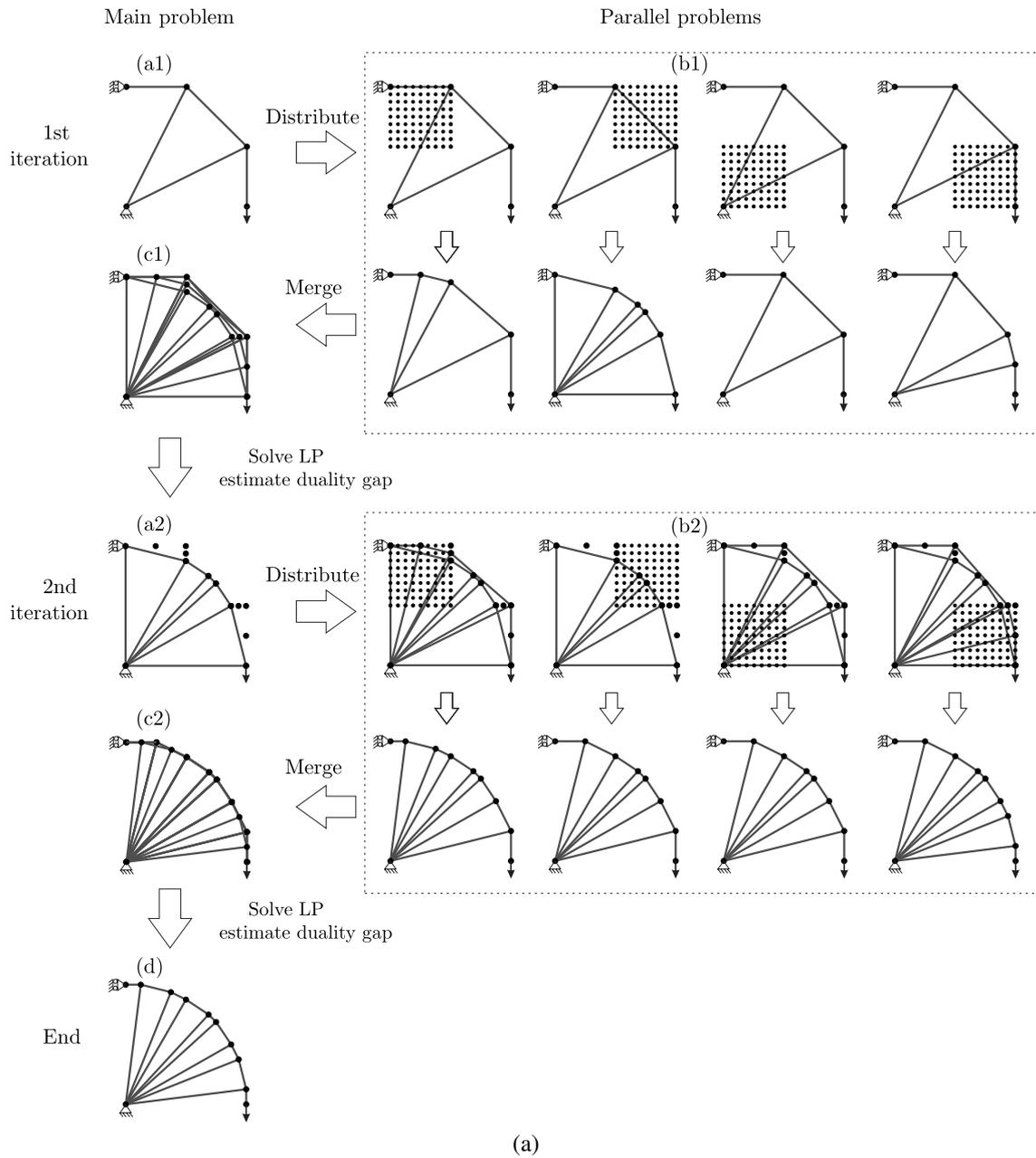


Figure 7.13: Potential parallel truss layout optimization solution strategy

7.5.1 Steps ready for parallel processing

The main developments in modern computing technology involve parallel processing techniques. In the geometry optimization process, there are steps that can readily be processed in parallel. The matrices of the whole geometry optimization problem introduced in Chapter 4 are assembled using locally derived matrices for every connection (e.g., a bar in truss problem). Since calculation of these matrices are completely separate, a parallel process can readily be employed. Furthermore, given that the operations required to evaluate these matrices are normally simple, GPU acceleration can be useful. Another process that can be parallelized is the numerical validation procedure (e.g., that used in merging nodes introduced in Chapter 4).

7.5.2 Taking information from previous iterations

At present, every iteration in the geometry optimization rationalization process is a new problem, with no information passed from previous iterations. The whole process can be made more efficient by taking some information from previous solutions. For example, Lagrange multipliers can be taken since they indicate the effectiveness of every constraint. In the node move limit constraints, if a Lagrange multiplier is non-zero, the corresponding constraint is then active - a node has moved to its limit, and a larger move limit may be allowed in the next iteration. As for inactive constraints, these may be temporarily removed to increase computational efficiency.

7.5.3 Alternative non-linear optimization methods

In this thesis, the interior point method (Section 2.4.2) has been utilized via the IPOPT software package to solve NLP problems. In general the method is numerically stable and efficient, but it may be useful to explore the potential of other methods, such as SQP and MMA (Section 2.4.3).

Before the advent of the interior point method, SQP was the most efficient method for solving general NLP problems. It is still viewed as one of the most efficient methods. Some studies (e.g., Nocedal et al. 2006) show that the interior point method is superior to SQP for solving large-scale problems, while SQP is better for highly non-linear problems. However, there are no mathematical proofs of these observations, so various numerical examples can be examined to explore the potential of using SQP in geometry optimization.

Another method discussed here is MMA. Though it was not initially designed for solving general NLP problems, it has been used widely in the field of topology optimization (Section 3.3.2). Compared with standard mathematical methods, MMA is particularly effective in solving problems that have a large number of variables. Another advantage is that the method is relatively simple to implement. Using self-developed software code is less problematic - unlike the use of the interior point method or SQP, where a self-developed software code is unlikely to be as efficient as existing packages. This is significant because MMA allows potential 'customization', so its algorithm can be tailored for particular geometry optimization problems to increase efficiency.

Chapter 8

Conclusions

The structural analysis and design tools developed in the thesis have been designed to provide a rapid computational capability for use in industry. Plasticity theory has been used to ensure that the methods involved are simple, fast and lead to solutions of clear status. The major research findings can be summarized as follows:

- To design truss structures, numerical layout optimization provides an efficient means of identifying (near-)optimal layouts. The same basic method is used in the discontinuity layout optimization (DLO) procedure, where the structural failure mechanism is identified directly, via a similar optimization process. To estimate the ultimate upper bound load-carrying capacity of reinforced concrete slabs, the yield-line method has been used in industry because of its power and simplicity. However, a problem has been noted: the solution depends on the assumed failure mechanism. Identification of a suitable failure mechanism is thus important. The DLO method can identify a suitable failure mechanism, overcoming this issue. However, when using layout optimization, the resulting layouts will often be found to be quite complex in form, suggesting that a means for generating rationalized truss layouts / yield-line patterns would be potentially useful.
- Two rationalization techniques have been investigated; the first technique involves a minor modification to the standard layout optimization formulation by including penalty factors associated with nodes (e.g., joints in a truss). The objective (cost) function is now affected by the number of nodes in the resulting layout. For this reason, short-length connections (e.g., truss bars) become potentially expensive, pushing the optimizer towards simplified layouts. The use of this technique does not alter the linear nature of the layout optimization problem and therefore can readily be implemented and solutions found. However, the solutions obtained are normally less efficient (e.g., a larger design volume will be found in the case of a truss structure). In addition, scenarios exist where eliminating short-length connections will not simplify (and may even complicate) the final layout.
- The second rationalization technique investigated involves performing a geometry optimization post-processing step, attempting to simplify a given layout by adjusting the positions of nodes. The linear nature of layout optimization is no longer retained; instead, a general

non-linear optimization is required. For truss and slab problems, the analytical expressions involved have been deduced, leading to an effective optimization process. In addition, many practical issues have been addressed, such as identifying strategies for merging nodes, creating crossover nodes, etc., to develop a versatile and reliable rationalization technique. Due to its non-linear nature, the computational cost (i.e., CPU time) associated with this technique is relatively high compared to standard layout optimization. Furthermore, the global optimality of the solution is not ensured. Nevertheless, assuming a good starting layout can be derived using standard layout optimization, the layouts obtained using this technique are effectively rationalized, and the corresponding solutions are normally improved (compared with the starting layout optimization solutions).

- Slabs found in industry are often quite complex (e.g. these may be of irregular shape and/or have unusual boundary conditions). Traditionally yield-line analysis is performed by hand, which is a tedious process and may lead to non-conservative designs. The DLO method thus provides an extremely efficient tool for use in industry, bringing significant benefits. To ensure various practical slab problems can be modelled, a number of new features have been added into DLO (and have also been included in the rationalization process) in this thesis: orthotropic reinforcement can now be modelled; to permit uplift behaviour, new support types have been introduced, including several column support types so that flat slabs can be modelled; additional loading types, including point, line and patch loads, have also been incorporated. A number of example problems taken from literature and from industry have been used to demonstrate the effectiveness of DLO when tackling slabs incorporating these new features.

Chapter 9

Recommendations for future work

Areas for further work are indicated below:

- It has been shown that the geometry optimization rationalization technique may be affected by issues related to non-smoothness of the mathematical functions involved. Though in this thesis steps have been taken to ensure that only smooth functions are present in the optimization process, these steps can affect the solutions. There is therefore the potential to address the non-smoothness issue more rigorously, leading to less, or even no, impact on the solutions.
- In this thesis, efforts have been made to enhance the computational efficiency of the geometry optimization process. However, as discussed earlier, there is room to improve computational efficiency further. For example, investigating the dual problem and the Lagrange multipliers would potentially be useful. Also it is worth exploring the potential for parallel processing techniques to be applied in both layout and geometry optimization. For industrial applications, highly accurate solutions (e.g. having less than 0.1% margin of error) may not be necessary, so that schemes may be designed to obtain approximate solutions, leading to significant enhancement of computational efficiency.
- The efficacy of the rationalization technique applied to 2D and (simple) 3D trusses has been demonstrated in this thesis. However, to develop software suitable for use in industry, real-world 3D structures need to be considered. As discussed earlier, 3D cases bring many practical issues that have not been addressed here; these can be investigated in the near future. Also, a real-world structure often contains additional structural components, not just truss bars; therefore, it is useful to incorporate these components into the layout and geometry optimization process (e.g. shells, cables, etc) .

References

- Achtziger, W. (2007), ‘On simultaneous optimization of truss geometry and topology’, *Structural and Multidisciplinary Optimization* **33**, 285–304.
- Allaire, G., Jouve, F. & Toader, A. (2002), ‘A level-set method for shape optimization’, *Comptes Rendus de l’Académie des Sciences - Series I* **334**, 1125–1130.
- Allaire, G., Jouve, F. & Toader, A. (2004), ‘Structural optimization using sensitivity analysis and a level-set method’, *Journal of Computational Physics* **194**, 363–393.
- Anderheggen, E. and Knöpfel, H. (1972), ‘Finite element limit analysis using linear programming’, *International Journal of Solids and Structures* **8**(12), 1413–1431.
- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S. & Sigmund, O. (2011), ‘Efficient topology optimization in MATLAB using 88 lines of code’, *Structural and Multidisciplinary Optimization* **43**, 1–16.
- Azid, I. A., Kwan, A. S. K. & Seetharamu, K. N. (2002), ‘An evolutionary approach for layout optimization of a three-dimensional truss’, *Structural and Multidisciplinary Optimization* **24**, 333–337.
- Bäcklund, J. (1973), Finite element analysis of nonlinear structures, PhD thesis, Chalmers Tekniska Högskola, Göteborg.
- Balasubramanyam, K. V. & Kalyanaraman, V. (1988), ‘Yield-line analysis by linear programming’, *Journal of Structural Engineering* **114**(6), 1431–1437.
- Bauer, S. & Lackner, R. (2015), ‘Gradient-based adaptive discontinuity layout optimization for the prediction of strength properties in matrix-inclusion materials’, *International Journal of Solids and Structures* **63**, 82–98.
- Beeby, A. W. (1997), ‘Ductility in reinforced concrete: why is it needed and how is it achieved?’, *The structural Engineer* **75**, 311–318.
- Bendsøe, M. P. (1989), ‘Optimal shape design as a material distribution problem’, *Structural optimization* **1**, 193–202.
- Bendsøe, M. P., Ben-Tal, A. & Zowe, J. (1994), ‘Optimization methods for truss geometry and topology design’, *Structural Optimization* **7**, 141–159.

- Bendsøe, M. P. & Kikuchi, N. (1988), 'Generating optimal topologies in structural design using a homogenization method', *Computer Methods in Applied Mechanics and Engineering* **71**, 197–224.
- Bendsøe, M. P. & Sigmund, O. (1999), 'Material interpolation schemes in topology optimization', *Archive of applied mechanics* **69**, 635–654.
- Bendsøe, M. P. & Sigmund, O. (2003), *Topology optimization: theory, methods and applications*, 2nd edn, Springer.
- Bleyer, J. & Buhner, P. (2013), 'On the performance of non-conforming finite elements for the upper bound limit analysis of plates', *International Journal for Numerical Methods in Engineering* **94**(3), 308–330.
- Bourdin, B. & Chambolle, A. (2003), 'Design-dependent loads in topology optimization', *ESAIM: Control, Optimisation and Calculus of Variations* **9**, 19–48.
- Boyd, S. & Vandenberghe, L. (2004), *Convex optimization*, Cambridge University Press.
- Calladine, C. R. (1969), *Engineering plasticity*, 1st edn, Pergamon Press.
- Chakrabarty, J. (2006), *Theory of plasticity*, Elsevier.
- Chakrabarty, J. (2010), *Applied plasticity*, 2nd edn, Springer.
- Chan, A. S. L. (1962), The design of Michell optimum structures, Technical Report 3303, Aeronautical Research Council Reports and Memoranda, London.
- Chan, H. S. Y. (1972), 'The collapse load of reinforced concrete plates', *International Journal for Numerical Methods in Engineering* **5**, 57–64.
- Cheng, G. (1995), 'Some aspects of truss topology optimization', *Structural and Multidisciplinary Optimization* **10**, 173–179.
- Corliss, G., Faure, C., Griewank, A., Hascoet, L. & Naumann, U., eds (2002), *Automatic Differentiation of Algorithms: From Simulation to Optimization*, Springer.
- Cox, H. L. (1965), *The design of structures of least weight*, 1st edn, Pergamon press, Oxford.
- Czarnecki, S. (2003), 'Compliance optimization of the truss structures', *Computer Assisted Mechanics and Engineering Sciences* **10**, 117–137.
- Dantzig, G. B. (1949), 'Programming in a linear structure', *Econometrica* **17**, 73–74.
- Darwich, W., Gilbert, M. & Tyas, A. (2010), 'Optimum structure to carry a uniform load between pinned supports', *Structural and Multidisciplinary Optimization* **42**, 33–42.
- Descamps, B. & Filomeno Coelho, R. (2013), 'A lower-bound formulation for the geometry and topology optimization of truss structures under multiple loading', *Structural and Multidisciplinary Optimization* **48**(1), 49–58.

- Dorn, W. S., Gomory, R. E. & Greenberg, H. J. (1964), 'Automatic design of optimal structures', *Journal de Mecanique* **3**, 25–52.
- Excell, J. & Nathan, S. (2010), 'The rise of additive manufacturing'.
URL: <http://www.theengineer.co.uk/in-depth/the-big-story/the-rise-of-additive-manufacturing/1002560.article>
- Fiacco, A. V. & McCormick, G. P. (1968), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York.
- Forth, S. A., Tadjouddine, M., Pryce, J. D. & Reid, J. K. (2004), 'Jacobian code generated by source transformation and vertex elimination can be as efficient as hand-coding', *ACM Transactions on Mathematical Software* **30**(3), 266–299.
- Fox, E. N. (1974), 'Limit analysis for plates: the exact solution for a clamped square plate of isotropic homogeneous material obeying the square yield criterion and loaded by uniform pressure', *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **227**, 121–155.
- Frisch, K. R. (1955), The logarithmic potential method of convex programming, Technical report, University Institute of Economics, Oslo, Norway.
- Gil, L. & Andreu, A. (2001), 'Shape and cross-section optimisation of a truss structure', *Computers & Structures* **79**, 681–689.
- Gilbert, M., He, L., Smith, C. C. & Le, C. V. (2014), 'Automatic yield-line analysis of slabs using discontinuity layout optimization', *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **470**(2168).
- Gilbert, M., Smith, C. C. & Pritchard, T. J. (2010), Masonry arch analysis using discontinuity layout optimisation, in 'Proceedings of the ICE - Engineering and Computational Mechanics', Vol. 163, pp. 155–166.
- Gilbert, M. & Tyas, A. (2003), 'Layout optimization of large-scale pin-jointed frames', *Engineering Computations: International Journal for Computer-Aided Engineering* **20**(8), 1044–1064.
- Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A. & Wright, M. H. (1986), 'On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method', *Mathematical Programming* **36**, 183–209.
- Graczykowski, C. & Lewiński, T. (2005), 'The lightest plane structures of a bounded stress level transmitting a point load to a circular support', *Control and Cybernetics* **34**(1), 227–253.
- Graczykowski, C. & Lewiński, T. (2006a), 'Michell cantilevers constructed within trapezoidal domains - Part I: geometry of Hencky nets', *Structural and Multidisciplinary Optimization* **32**, 347–368.
- Graczykowski, C. & Lewiński, T. (2006b), 'Michell cantilevers constructed within trapezoidal domains - Part II: virtual displacement fields', *Structural and Multidisciplinary Optimization* **32**, 463–471.

- Graczykowski, C. & Lewiński, T. (2007a), 'Michell cantilevers constructed within trapezoidal domains-Part III: force fields', *Structural and Multidisciplinary Optimization* **33**, 1–19.
- Graczykowski, C. & Lewiński, T. (2007b), 'Michell cantilevers constructed within trapezoidal domains: Part IV: complete exact solutions of selected optimal designs and their approximations by trusses of finite number of joints', *Structural and Multidisciplinary Optimization* **33**, 113–129.
- Graczykowski, C. & Lewiński, T. (2010), 'Michell cantilevers constructed within a half strip. Tabulation of selected benchmark results', *Structural and Multidisciplinary Optimization* **42**(6), 869–877.
- Griewank, A. & Walther, A. (2008), *Evaluating derivatives: principles and techniques of algorithmic differentiation*, 2nd edn, Society for Industrial and Applied Mathematics.
- GUROBI (2014), 'GUROBI optimizer quick start guide'.
URL: <http://www.gurobi.com/products/gurobi-optimizer>
- Hawksbee, S., Smith, C. & Gilbert, M. (2013), 'Application of discontinuity layout optimization to three-dimensional plasticity problems', *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **469**(2155).
- He, L. & Gilbert, M. (2015a), 'Rationalization of trusses generated via layout optimization', *Structural and Multidisciplinary Optimization* **52**, 677–694.
- He, L. & Gilbert, M. (2015b), Rationalization technique for yield-line analysis of slabs using geometry optimization. Submitted for publication.
- Hemp, W. S. (1958), Theory of structural design, Technical report, The college of aeronautics, Cranfield.
- Hemp, W. S. (1973), *Optimum structures*, Clarendon Press, Oxford.
- Hemp, W. S. (1974), 'Michell frameworks for uniform load between fixed supports', *Engineering Optimization* **1**, 61–69.
- Hillerborg, A. (1956), 'Theory of equilibrium for reinforced concrete slabs', *Betong* **41**(1), 171–181.
- Hillerborg, A. (1959), *The strip method*, Stockholm, Svenska Riks-byggen.
- Hillerborg, A. (1982), 'The advanced strip method - a simple design tool', *Magazine of Concrete Research* **34**(121), 175–181.
- Hodge, P. G. & Belytschko, T. (1968), 'Numerical methods for the limit analysis of plates', *Journal of Applied Mechanics* **35**, 796–802.
- Huang, X. & Xie, Y. M. (2010a), 'A further review of ESO type methods for topology optimization', *Structural and Multidisciplinary Optimization* **41**(5), 671–683.

- Huang, X. & Xie, Y. M. (2010*b*), 'Evolutionary topology optimization of continuum structures with an additional displacement constraint', *Structural and Multidisciplinary Optimization* **40**, 409–416.
- Ingerslev, A. (1923), 'The strength of rectangular slabs', *Journal of the Institution of Civil Engineers* **1**(1), 3–14.
- Islam, S. & Park, R. (1971), 'Yield-line analysis of two way reinforced concrete slabs with openings', *The Structural Engineer* **49**(6), 269–276.
- Jackson, A. (2010), Modelling the collapse behaviour of reinforced concrete slabs, PhD thesis, Department of Engineering, University of Cambridge.
- Jackson, A. M. & Middleton, C. R. (2013), 'Closely correlating lower and upper bound plastic analysis of real slabs', *The Structural Engineer* **91**, 34–40.
- Johansen, K. W. (1943), *Brudlinieteorier*, Gjellerup Forlag, Copenhagen. (English translation: Yield-Line Theory, Cement and Concrete Association, London, 1962).
- Johansen, K. W. (1968), *Pladeformler*, Polyteknisk Forlag, Copenhagen. (English translation: Yield-line formulae for slabs, Cement and Concrete Association, London, 1972).
- Johnson, D. (1994), 'Mechanism determination by automated yield-line analysis', *Structural Engineering* **72**, 323–327.
- Johnson, D. (1995), 'Yield-line analysis by sequential linear programming', *International Journal of Solids and Structures* **32**, 1395–1404.
- Johnson, D. (2006), Collapse analysis of reinforced concrete slabs: are the up and down roads one and the same?, in 'Advances in Engineering Structures, Mechanics & Construction', Springer, Ontario, Canada, pp. 823–831.
- Johnson, W. (1961), 'An analogy between upper-bound solutions for plane-strain metal working and minimum-weight two-dimensional frames', *International Journal of Mechanical Sciences* **3**, 239–246.
- Jones, L. L. & Wood, R. H. (1967), *Yield-line analysis of slabs*, American Elsevier Publishing Company, Inc., New York.
- Kane, C. & Schoenauer, M. (1996), 'Topological optimum design using genetic algorithms', *Control and Cybernetics* **25**, 1059–1087.
- Karmarkar, N. K. (1984), 'A new polynomial-time algorithm for linear programming', *Combinatorica* **4**, 373–395.
- Kennedy, G. & Goodchild, C. H. (2004), *Practical yield line design*, The Concrete Centre, Surrey.
- Kirk, D. & Hwu, W. (2010), *Programming massively parallel processors*, Elsevier.

- Krabbenhoft, K. & Damkilde, L. (2003), 'A general non-linear optimization algorithm for lower bound limit analysis', *International Journal for Numerical Methods in Engineering* **56**(2), 165–184.
- Krabbenhøft, K., Lyamin, A. & Sloan, S. (2007), 'Formulation and solution of some plasticity problems as conic programs', *International Journal of Solids and Structures* **44**(5), 1533–1549.
- Krenk, S., Damkilde, L. & Høyer, O. (1994), 'Limit analysis and optimal design of plates with equilibrium elements', *Journal of Engineering Mechanics* **120**(6), 1237–1254.
- Kwan, A. K. H. (2004), 'Dip and strike angles method for yield line analysis of reinforced concrete slabs', *Magazine of Concrete Research* **56**(8), 487–498.
- Kwan, A. K. H. (2013), 'Yield line analysis of reinforced concrete slabs with openings by dip and strike angles method', *Advances in Structural Engineering* **16**(7), 1223–1223.
- Le, C., Nguyen-Xuan, H. & Nguyen-Dang, H. (2010), 'Upper and lower bound limit analysis of plates using FEM and second-order cone programming', *Computers and Structures* **88**(1–2), 65–73.
- Le, C. V., Gilbert, M. & Askes, H. (2009), 'Limit analysis of plates using the EFG method and second-order cone programming', *International Journal for Numerical Methods in Engineering* **78**, 1532–1552.
- Le, C. V., Gilbert, M. & Askes, H. (2010), 'Limit analysis of plates and slabs using a meshless equilibrium formulation', *International Journal for Numerical Methods in Engineering* **83**(13), 1739–1758.
- Lewiński, T. (2004), 'Michell structures formed on surfaces of revolution', *Structural and Multidisciplinary Optimization* **28**(1), 20–30.
- Lewiński, T. (2005), 'Personal communication'.
- Lewiński, T. & Rozvany, G. I. N. (2007), 'Exact analytical solutions for some popular benchmark problems in topology optimization II: three-sided polygonal supports', *Structural and Multidisciplinary Optimization* **33**, 337–349.
- Lewiński, T. & Rozvany, G. I. N. (2008a), 'Analytical benchmarks for topological optimization IV: square-shaped line support', *Structural and Multidisciplinary Optimization* **36**, 143–158.
- Lewiński, T. & Rozvany, G. I. N. (2008b), 'Exact analytical solutions for some popular benchmark problems in topology optimization III: L-shaped domains', *Structural and Multidisciplinary Optimization* **35**, 165–174.
- Lewiński, T., Rozvany, G. I. N., Sokó, T. & Bobotowski, K. (2013), 'Exact analytical solutions for some popular benchmark problems in topology optimization III: L-shaped domains revisited', *Structural and Multidisciplinary Optimization* **47**(6), 937–942.
- Lewiński, T., Zhou, M. & Rozvany, G. I. N. (1994a), 'Extended exact solutions for least-weight truss layouts - Part I: cantilever with a horizontal axis of symmetry', *International Journal of Mechanical Sciences* **36**(5), 375–398.

- Lewiński, T., Zhou, M. & Rozvany, G. I. N. (1994b), 'Extended exact solutions for least-weight truss layouts - Part II: unsymmetric cantilevers', *International Journal of Mechanical Sciences* **36**(5), 399–419.
- LimitState (2008), 'Analysis & design software for engineers'.
URL: <http://www.limitstate.com/>
- LimitState Ltd (2015), 'LimitState:SLAB, version 1.0', <http://www.limitstate.com/slab>.
- Martínez, P., Martí, P. & Querin, O. M. (2007), 'Growth method for size, topology, and geometry optimization of truss structures', *Structural and Multidisciplinary Optimization* **33**, 13–26.
- Maunder, E. A. W. & Ramsay, A. C. A. (2012), 'Equilibrium models for lower bound limit analyses of reinforced concrete slabs', *Computers & Structures* **108-109**, 100–109.
- Mazurek, A. (2012), 'Geometrical aspects of optimum truss like structures for three-force problem', *Structural and Multidisciplinary Optimization* **45**, 21–32.
- Mazurek, A., Baker, W. F. & Tort, C. (2011), 'Geometrical aspects of optimum truss like structures', *Structural and Multidisciplinary Optimization* **43**, 231–242.
- McConnell, R. E. (1974), 'Least-weight frameworks for loads across span', *Journal of the Engineering Mechanics Division* **100**, 885–901.
- Michell, A. G. M. (1904), 'The limits of economy of material in frame-structures', *Philosophical Magazine* **8**, 589–597.
- Middleton, C. R. (1997), 'Concrete bridge assessment: an alternative approach', *The Structural Engineer* **75**(23/24), 403–409.
- Mlejnek, H. P. (1992), 'Some aspects of the genesis of structures', *Structural Optimization* **5**, 64–69.
- MOSEK (2011), 'The MOSEK optimization tools manual'.
URL: <http://docs.mosek.com/6.0/tools/index.html>
- Munro, J. & Da Fonseca, A. M. A. (1978), 'Yield-line method by finite elements and linear programming', *The Structural Engineer* **56**(2), 37–44.
- Nagtegaal, J. C. & Prager, W. (1973), 'Optimal layout of a truss for alternative loads', *International Journal of Mechanical Sciences* **15**, 585–592.
- Neidinger, R. D. (2010), 'Introduction to automatic differentiation and MATLAB object-oriented programming', *Society for Industrial and Applied Mathematics* **52**(3), 545–563.
- Nielsen, M. P. & Hoang, L. C. (2011), *Limit analysis and concrete plasticity*, 3rd edn, Taylor and Francis Group.
- Nocedal, J., Wright, S. J. & Robinson, S. M. (2006), *Numerical Optimization*, 2nd edn, Springer.

- Olsen, P. C. (1998), 'The influence of the linearisation of the yield surface on the load-bearing capacity of reinforced concrete slabs', *Computer Methods in Applied Mechanics and Engineering* **162**(1), 351–358.
- Park, R. & Gamble, W. L. (2000), *Reinforced concrete slabs*, 2nd edn, John Wiley & Sons, Inc.
- Parkes, E. W. (1975), 'Joints in optimum frameworks', *International Journal of Solids and Structures* **11**(9), 1017 – 1022.
- Pichugin, A. V., Tyas, A. & Gilbert, M. (2012), 'On the optimality of Hemp's arch with vertical hangers', *Structural and Multidisciplinary Optimization* **46**, 17–25.
- Pichugin, A. V., Tyas, A., Gilbert, M. & He, L. (2015), Optimum structure for a uniform load over multiple spans. (in press).
- Prager, W. (1958), On a problem of optimal design, in 'Non-homogeneity in elasticity and plasticity', Warsaw, pp. 125–132.
- Prager, W. (1978a), 'Nearly optimal design of trusses', *Computers and Structures* **8**, 451–454.
- Prager, W. (1978b), 'Optimal layout of trusses with finite numbers of joints', *Journal of the Mechanics and Physics of Solids* **26**, 241–250.
- Pritchard, T., Gilbert, M. & Tyas, A. (2005), 'Plastic layout optimization of large-scale frameworks subject to multiple load cases, member self-weight and with joint length penalties', *6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil*.
- Pritchard, T. J. (2004), Novel techniques in structural layout optimisation, PhD thesis, University of Sheffield.
- Querin, O. M. (1997), Evolutionary structural optimisation: stress based formulation and implementation, PhD thesis, University of Sydney.
- Rahami, H., Kaveh, A. & Gholipour, Y. (2008), 'Sizing, geometry and topology optimization of trusses via force method and genetic algorithm', *Engineering Structures* **30**, 2360–2369.
- Ramsay, A. C. A. & Johnson, D. (1997), 'Geometric optimization of yield-line patterns using a direct search method', *Structural optimization* **115**, 108–115.
- Ramsay, A. C. A. & Johnson, D. (1998), 'Analysis of practical slab configurations using automated yield-line analysis and geometric optimization of fracture patterns', *Engineering Structures* **20**(8), 647–654.
- Ramsay, A. C. A., Maunder, E. & Gilbert, M. (2015), 'Yield-line analysis - is the 10% rule safe?', *NAFEMS Benchmark magazine* pp. 15–20. January.
- Rees, D. W. A. (2006), *Basic engineering plasticity*, Elsevier.
- Regan, P. E. & Yu, C. W. (1973), *Limit state design of structural concrete*, Chatto and Windus, London, United Kingdom.

- Rozvany, G. I. N. (1996), ‘Some shortcomings in Michell’s truss theory’, *Structural Optimization* **12**, 244–250.
- Rozvany, G. I. N. (1998), ‘Exact analytical solutions for some popular benchmark problems in topology optimization’, *Structural Optimization* **15**, 42–48.
- Rozvany, G. I. N. (2009), ‘A critical review of established methods of structural topology optimization’, *Structural and Multidisciplinary Optimization* **37**, 217–237.
- Rozvany, G. I. N. & Lewiński, T., eds (2014), *Topology Optimization in Structural and Continuum Mechanics*, Springer.
- Sauer, T. (2011), *Numerical Analysis*, 2nd edn, Pearson.
- Sigmund, O. (2001), ‘A 99 line topology optimization code written in Matlab’, *Structural and Multidisciplinary Optimization* **21**, 120–127.
- Sigmund, O. (2007), ‘Morphology-based black and white filters for topology optimization’, *Structural and Multidisciplinary Optimization* **33**, 401–424.
- Sigmund, O. (2011), ‘On the usefulness of non-gradient approaches in topology optimization’, *Structural and Multidisciplinary Optimization* **43**, 589–596.
- Sigmund, O. & Maute, K. (2013), ‘Topology optimization approaches’, *Structural and Multidisciplinary Optimization* **48**, 1031–1055.
- Smith, C. C. & Gilbert, M. (2007), ‘Application of discontinuity layout optimization to plane plasticity problems’, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **463**, 2461–2484.
- Smith, C., Gilbert, M., Todd, I. & Derguti, F. (2015), Application of layout optimization to the design of additively manufactured metallic components. Accepted for publication.
- Sokół, T. (2011), Topology optimization of large-scale trusses using ground structure approach with selective subsets of active bars, in ‘19th International Conference on ‘Computer Methods in Mechanics’, CMM2011’, ACM Press, pp. 457–458.
- Sokół, T. & Rozvany, G. I. N. (2012), ‘New analytical benchmarks for topology optimization and their implications. Part I: bi-symmetric trusses with two point loads between supports’, *Structural and Multidisciplinary Optimization* **46**(4), 477–486.
- Sokolowski, J. & Zochowski, A. (1999), ‘The topological derivative method in shape optimization’, *SIAM Journal on Control and Optimization* **37**(4), 1251–1272.
- Spillers, W. R. & Lev, O. (1971), ‘Design for two loading conditions’, *International Journal of Solids and Structures* **7**(9), 1261–1267.
- Strang, G. & Kohn, R. V. (1983), ‘Hencky-Prandtl nets and constrained Michell trusses’, *Computer Methods in Applied Mechanics and Engineering* **36**, 207–222.
- Sutter, H. & Larus, J. (2005), ‘Software and the concurrency revolution’, *ACM Queue* **3**, 54–62.

- Svanberg, K. (1987), 'The method of moving asymptotes - a new method for structural optimization', *International Journal for Numerical Methods in Engineering* **24**, 359–373.
- Svanberg, K. (1995), A globally convergent version of MMA without linesearch, in 'Proceedings of the first world congress of structural and multidisciplinary optimization', Pergamon Press, pp. 9–16.
- Svanberg, K. (2002), 'A class of globally convergent optimization methods based on conservative convex separable approximations', *SIAM journal on optimization* **12**(2), 555–573.
- Tai, K. & Akhtar, S. (2005), 'Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme', *Structural and Multidisciplinary Optimization* **30**, 113–127.
- Thavalingam, A., Jennings, A., Sloan, D. & McKeown, J. J. (1999), 'Computer-assisted generation of yield-line patterns for uniformly loaded isotropic slabs using an optimisation strategy', *Engineering Structures* **21**, 488–496.
- Tyas, A., Gilbert, M. & Pritchard, T. J. (2006), 'Practical plastic layout optimization of trusses incorporating stability considerations', *Computers & Structures* **84**(3-4), 115–126.
- Tyas, A., Pichugin, A. V. & Gilbert, M. (2010), 'Optimum structure to carry a uniform load between pinned supports: exact analytical solution', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **467**(2128), 1101–1120.
- Vanderbei, R. J. (2001), *Linear programming: foundations and extensions*, 2nd edn, Springer Verlag.
- Vigerske, S. & Wachter, A. (2013), 'Introduction to IPOPT: a tutorial for downloading, installing, and using IPOPT'.
URL: <http://www.coin-or.org/Ipopt/documentation/>
- Wächter, A. & Biegler, L. T. (2006), 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming', *Mathematical Programming* **106**, 25–57.
- Wang, S. Y. & Tai, K. (2005), 'Structural topology design optimization using Genetic Algorithms with a bit-array representation', *Computer Methods in Applied Mechanics and Engineering* **194**, 3749–3770.
- Wikipedia (2012), 'Anji bridge'.
URL: en.wikipedia.org/wiki/Anji_Bridge
- Wood, R. H. & Jones, L. L. (1967), *Yield-line analysis of slabs*, Thames and Hudson, Chatto and Windus.
- Wright, M. H. (2004), 'The interior-point revolution in optimization: history, recent developments, and lasting consequences', *Bulletin of the American Mathematical Society* **42**(1), 39–57.
- Wüst, J. & Wagner, W. (2008), 'Systematic prediction of yield-line configurations for arbitrary polygonal plates', *Engineering Structures* **30**(7), 2081–2093.

-
- Xie, Y. M. & Steven, G. P. (1993), 'A simple evolutionary procedure for structural optimization', *Computers & Structures* **49**(5), 885–896.
- Zegard, T. & Paulino, G. H. (2014), 'GRAND - Ground structure based topology optimization for arbitrary 2D domains using MATLAB', *Structural and Multidisciplinary Optimization* **50**(5), 861–882.
- Zhou, M. & Rozvany, G. I. N. (1991), 'The COC algorithm, Part II: topological, geometrical and generalized shape optimization', *Computer Methods in Applied Mechanics and Engineering* **89**(1-3), 309–336.

Appendix A

Automatic yield-line analysis of slabs using discontinuity layout optimization



Cite this article: Gilbert M, He L, Smith CC, Le CV. 2014 Automatic yield-line analysis of slabs using discontinuity layout optimization. *Proc. R. Soc. A* **470**: 20140071.
<http://dx.doi.org/10.1098/rspa.2014.0071>

Received: 28 January 2014

Accepted: 14 April 2014

Subject Areas:

structural engineering, civil engineering,
mechanical engineering

Keywords:

slabs, plates, plasticity, limit analysis,
layout optimization, yield-line analysis

Author for correspondence:

Matthew Gilbert

e-mail: m.gilbert@sheffield.ac.uk

Automatic yield-line analysis of slabs using discontinuity layout optimization

Matthew Gilbert¹, Linwei He¹, Colin C. Smith¹
and Canh V. Le²

¹Department of Civil and Structural Engineering, University of Sheffield, Sir Frederick Mappin Building, Mappin St., Sheffield S1 3JD, UK

²Department of Civil Engineering, International University - VNU HCMC, Ho Chi Minh City, Vietnam

The yield-line method of analysis is a long established and extremely effective means of estimating the maximum load sustainable by a slab or plate. However, although numerous attempts to automate the process of directly identifying the critical pattern of yield-lines have been made over the past few decades, to date none has proved capable of reliably analysing slabs of arbitrary geometry. Here, it is demonstrated that the discontinuity layout optimization (DLO) procedure can successfully be applied to such problems. The procedure involves discretization of the problem using nodes interconnected by potential yield-line discontinuities, with the critical layout of these then identified using linear programming. The procedure is applied to various benchmark problems, demonstrating that highly accurate solutions can be obtained, and showing that DLO provides a truly systematic means of directly and reliably automatically identifying yield-line patterns. Finally, since the critical yield-line patterns for many problems are found to be quite complex in form, a means of automatically simplifying these is presented.

1. Introduction

The yield-line method is a long established and highly effective means of estimating the ultimate load-carrying capacity of slabs and plates. The term ‘yield-line’ was coined by Ingerslev [1], with a comprehensive

© 2014 The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License <http://creativecommons.org/licenses/by/3.0/>, which permits unrestricted use, provided the original author and source are credited.

theory developed by Johansen [2], and, in parallel, by Gvozdev [3]. The upper bound status of the method within the context of the then emerging plastic theories of structural analysis was later confirmed by others (e.g. [4,5]). The method traditionally involves postulating a collapse mechanism which is compatible with the boundary conditions and then using the principle of virtual work to compute the ultimate load, or 'load factor'.

For certain special cases, it has been possible to calculate provably exact failure load factors (e.g. Fox [6] established the exact solution for the case of a uniformly loaded fixed square slab). However, in the case of most real-world geometrical configurations exact load factors are not available. In such cases, unless the critical yield-line pattern has been identified, the computed load factor will over-estimate the true load factor. While lower bound methods can be used to bound the load factor from below, the gap between a yield-line solution and a solution obtained using common hand-based lower bound analysis methods (e.g. the strip method proposed by Hillerborg [7], which simplifies the problem by allowing analyst/designer to select load paths while ignoring twisting moments) will typically be found to be quite wide. This situation is clearly unsatisfactory and has undoubtedly limited the extent to which hand-based yield-line analysis is used in practice.

Consequently, various computational methods have been applied to the problem over the past few decades. For example, Anderheggen & Knopfel [8] were among the first to apply finite-element limit-analysis techniques to slabs, showing that rigorous lower bound solutions could be obtained providing a suitable element formulation was employed. More recently, it has been demonstrated that nonlinear optimization [9] and the second-order cone programming techniques [10–12] can be applied, obviating the need to linearize the yield surface. Meshless (element-free Galerkin) methods have also been applied to slab problems, and reasonably good approximations of the collapse load factor can be obtained rapidly [13]. However, despite the promise of such methods, they have not found their way into routine engineering practice and at present practising engineers typically have to instead rely on potentially cumbersome iterative elasto-plastic analysis methods. Furthermore, since finite-element (and meshless) methods are concerned with treatment of an underlying continuum mechanics problem, these methods do not directly identify patterns of yield-lines, though in many cases these can subsequently be inferred from the output.

To address this, computational methods capable of explicitly identifying yield-lines have also been developed in parallel. For example, Chan [14], and later workers such as Munro & Da Fonseca [15] and Balasubramanyam & Kalyanaraman [16], proposed (very similar) methods in which potential yield-lines are placed at the boundaries of rigid elements arranged in a finite-element mesh. This permits linear programming (LP) to then be used to identify the most critical layout of yield-lines. While available computing resources of the time meant that only relatively coarse meshes could be treated, the most significant problem is sensitivity of the results obtained to the chosen initial mesh layout, with the consequence that refining the mesh alone does not necessarily lead to an improved estimate of the collapse load factor. This, for example, means that when using a structured triangular mesh, however fine, it is impossible to accurately simulate a fan-type mechanism. Numerous attempts to overcome this fundamental problem have been made, for example, by subsequently changing the topology of the initial rigid finite-element mesh through the use of geometry optimization or other techniques (e.g. [17–19]), but no fully satisfactory solution to the problem has been found. (This was also the conclusion of Johnson [20], who, after many years work in the field, asserted that the upper bound problem was simply 'too difficult' to solve computationally.) A possible way round this was recently put forward by Jackson [21] and Jackson & Middleton [22], who proposed that the lower bound solution could be used to suggest the form of the yield-line solution. Promising results were presented, but the procedure involves both a manual interpretation step and a potentially problematic and time-consuming nonlinear optimization step, suggesting that a truly systematic means of identifying yield line patterns had yet to be found.

However, the popularity of application-specific yield-line analysis tools, for example the COBRAS reinforced concrete bridge assessment tool developed at the University of Cambridge,

and which involves automatically searching through a library of possible yield-line failure mechanisms [23], indicates that a systematic yield-line method would undoubtedly find widespread application. Furthermore, a 2004 industry report reiterated the potential economic benefits of using yield-line design, despite the fact that at present the analysis must by necessity be performed by hand [24]. In the report, it is recommended that, because a hand analysis may not lead to identification of the most critical mechanism, a 10% margin of error (safety factor) should pragmatically be assumed. However, the basis for this particular value is not entirely clear, and the fact that a factor of this sort is needed at all is clearly not entirely satisfactory.

In this paper, the upper bound problem will be revisited using a ‘discontinuous’ rather than continuum analysis approach, on the surface similar to the methods proposed by Chan [14], Munro & Da Fonseca [15] and others. However, the significant difference here is that by formulating the problem in terms of *discontinuities* rather than *elements*, a very much wider range of failure modes will be able to be identified, thereby overcoming the sensitivity to the initial mesh layout encountered when using previously proposed methods. Furthermore, rather than initially considering the yield-line analysis problem directly, as most others have done (with only limited success), the procedure described in this paper was developed following a conjecture that there existed a direct analogy between the layout of bars in optimum trusses and the layout of yield-lines in slabs, since such an analogy had been identified in the case of in-plane plasticity problems [25]. As the problem formulation is somewhat different in this case, the original sequence of development is also preserved in this paper, with the nature of the analogy examined initially.

2. Analogy between optimal layouts of truss bars and yield-lines

(a) Background

The analogy between the compatibility requirements of yield-line patterns and the equilibrium requirements of trusses appears to have been identified comparatively recently [26]. This finding is of interest since numerical layout optimization techniques have been applied to the problem of identifying optimal trusses for several decades (e.g. [27,28]). Furthermore, the efficiency of such methods have dramatically increased recently, with the advent of modern interior point LP solvers and also the application of adaptive refinement procedures [29]. Thus, layout optimization problems containing several billion potential connections between nodes (i.e. bars or yield-lines in this case) can now be solved on current generation personal computers.

However, while Denton [26] showed that a truss corresponding to a compatible yield-line pattern must have at least one state of self-stress (or ‘degree of redundancy’), it can be shown that there must always exist a statically determinate optimum solution for the single load case truss layout optimization problem. This makes the analogy perhaps less immediately obvious than that identified between discretized optimal truss layouts and the critical arrangement of slip-lines in plane plasticity problems [25]; in the latter case, many important plane plasticity problems have patterns of slip-lines defining the failure mechanism which correspond to the layouts of bars in statically determinate trusses. Furthermore, it is not immediately obvious how issues such as the presence of distributed out-of-plane live loading can be treated using the type of procedure used to identify optimal truss layouts (such loading is obviously often present in slab problems, but is absent from the basic truss layout optimization problem). To investigate this further, various approximate-discretized LP truss layout optimization formulations will now be considered.

(b) Layout optimization of trusses: linear programming formulations

First, consider a potential planar design domain which is discretized using n nodes and m potential nodal connections (truss bars). The classical ‘equilibrium’ plastic truss layout

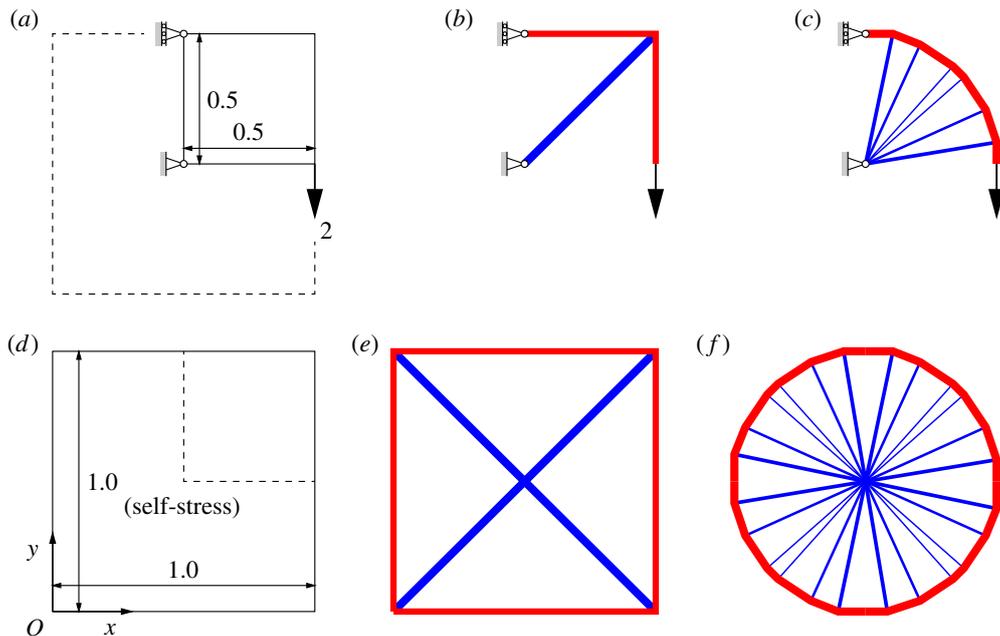


Figure 1. Simple truss layout optimization problems: (a) design domain with fixed pin and pin/roller supports and an applied load; (b) a solution to problem (a) with 2×2 nodes (volume = 4 when $\sigma^- = \sigma^+ = 1$); (c) as (b) but with 13×13 nodes (volume = 3.164, less than 1% greater than exact value of π); (d) alternative ‘self-stress’ problem; (e) solution to problem (d) with 2×2 nodes (volume = 16) and (f) as (e) but with 25×25 nodes (volume = 12.656, 4 \times the volume for problem (c)). (Online version in colour.)

optimization formulation for a single load case is defined in equation (2.1) as follows (after [27]):

$$\begin{aligned} \min \quad & V = \mathbf{c}^T \mathbf{q} \\ \text{subject to:} \quad & \mathbf{B} \mathbf{q} = \mathbf{f} \\ & \mathbf{q} \geq \mathbf{0}, \end{aligned} \tag{2.1}$$

where V is the total volume of the structure, $\mathbf{q}^T = \{q_1^+, q_1^-, q_2^+, q_2^- \dots q_m^-, q_m^-\}$, and q_i^+, q_i^- are the tensile and compressive forces in bar i ; $\mathbf{c}^T = \{l_1/\sigma_1^+, l_1/\sigma_1^-, l_2/\sigma_2^+, l_2/\sigma_2^- \dots l_m/\sigma_m^-, l_m/\sigma_m^-\}$, where l_i, σ_i^+ and σ_i^- are, respectively, the length and tensile and compressive yield stress of bar i . \mathbf{B} is a suitable $(2n \times 2m)$ equilibrium matrix containing direction cosines and $\mathbf{f}^T = \{f_1^x, f_1^y, f_2^x, f_2^y \dots f_n^x, f_n^y\}$ where f_j^x and f_j^y are the x and y components of the external load applied to node j ($j = 1 \dots n$). The presence of supports at nodes can be accounted for by omitting the relevant terms from \mathbf{f} , together with the corresponding rows from \mathbf{B} . This problem is in a form which can be solved using linear optimization, with the bar forces in \mathbf{q} being the LP variables.

Figure 1a shows the definition of a typical truss layout optimization problem, with the solutions when 2×2 nodes and 13×13 nodes are used to discretize the problem given in figure 1b,c, respectively. (In both cases, each node was inter-connected to every other node to create a ‘fully connected ground structure’, with LP then used to identify the optimum subset of truss bars). Note that, in the latter case, the solution is within 1% of the analytical optimum solution.

However, noting the observation of Denton [26] that the truss corresponding to a compatible yield-line pattern must have at least one state of self-stress (i.e. is ‘pre-stressed’), it is of interest to instead consider the closely related problem of finding the optimal layout of a truss which has no external loading (i.e. where $\mathbf{f} = \mathbf{0}$), but which is in a state of self-stress. Though this particular problem appears not to be explicitly considered in existing structural optimization literature, an

appropriate mathematical formulation can tentatively be postulated. Thus, since this remains a ‘layout optimization’ problem, it seems appropriate to prescribe the state of self-stress rather loosely, for example, leaving open the possibility of many different bars being subjected to the self-stress (i.e. so as not to over-constrain the problem). This means that a single constraint can be added to give the following modified problem formulation:

$$\begin{aligned} \min \quad & V = \mathbf{c}^T \mathbf{q} \\ \text{subject to:} \quad & \mathbf{B}\mathbf{q} = \mathbf{0} \\ & \mathbf{h}^T \mathbf{q} = 1 \\ & \mathbf{q} \geq \mathbf{0}, \end{aligned} \tag{2.2}$$

where $\mathbf{h}^T = \{h_1, -h_1, h_2, -h_2 \dots -h_m\}$ and where h_i is a factor used to prescribe how the self-stress is to be distributed between each bar i ($i = 1 \dots m$) in the frame. Alternatively, specific bars could be allocated specific prescribed self-stress forces, if required.

A sample self-stress problem is defined in figure 1d, with the solutions when 2×2 nodes and 25×25 nodes given in figure 1ef, respectively. To obtain the particular results shown, the self-stress coefficients in the constraint $\mathbf{h}^T \mathbf{q} = 1$ for each truss bar were defined by using the centre-point of the domain as a focus, achieved by using the following simple, though perhaps not intuitively obvious, rules: if the centre-point (i.e. [0.5, 0.5] in this case) lies in a vertical strip drawn directly above a given potential truss-bar i then coefficient h_i is taken as the perpendicular distance from the truss bar to the centre-point of the domain; otherwise, this is taken as zero. This gives solutions which are by inspection directly comparable to those for the problem defined in figure 1a, with the optimum structures shown in figure 1b,c clearly representing one-quarter of the structures shown in figure 1ef, respectively (which are in fact simple two-dimensional *tensegrity* structures, with the former being the main part of the ‘X-shaped module’ referred to by Snelson [30], hinting at the potential for this type of problem formulation to be adapted to synthesize such structures).

It is also evident that the topology of the solution given in figure 1f is reminiscent of the ‘fan’-type mechanism which is critical when a slab is subjected to a point load (e.g. [5]; the numerically computed volume is also within 1% of the analytical load factor for the slab problem when a unit load is applied). In fact, it will now be demonstrated that it is this latter formulation which is directly analogous to the yield-line layout optimization problem, with the *equilibrium* truss optimization problem corresponding to the *kinematic* yield-line layout optimization problem.

(c) Yield-line layout optimization: linear programming formulation

Maintaining precisely the same form of linear optimization problem as given in (2.2), the kinematic yield-line layout optimization formulation for an out-of-plane, quasi-statically loaded, perfectly plastic slab with supported edges and discretized using m nodal connections (yield-line discontinuities), n nodes and a single load case can be defined in equation (2.3) as follows:

$$\begin{aligned} \min \quad & E = \mathbf{g}^T \mathbf{d} \\ \text{subject to:} \quad & \mathbf{B}\mathbf{d} = \mathbf{0} \\ & \mathbf{f}_l^T \mathbf{d} = 1 \\ & \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{2.3}$$

where E is the energy dissipated due to rotation along the yield-lines, $\mathbf{d}^T = \{\theta_1^+, \theta_1^-, \theta_2^+, \theta_2^- \dots \theta_m^-\}$, where θ_i^+, θ_i^- are the positive and negative relative rotations along the yield-line i ; $\mathbf{g}^T = \{m_{p1}^+ l_1, m_{p1}^- l_1, m_{p2}^+ l_2, m_{p2}^- l_2 \dots m_{pm}^+ l_m\}$, where l_i , m_{pi}^+ and m_{pi}^- are, respectively, the length and positive and negative plastic moment of resistance per unit length for potential yield-line i . Note that when Johansen’s square yield criterion [2] is applied to isotropic slab problems, the plastic moment of resistance per unit length will be the same for all potential yield-lines, irrespective of

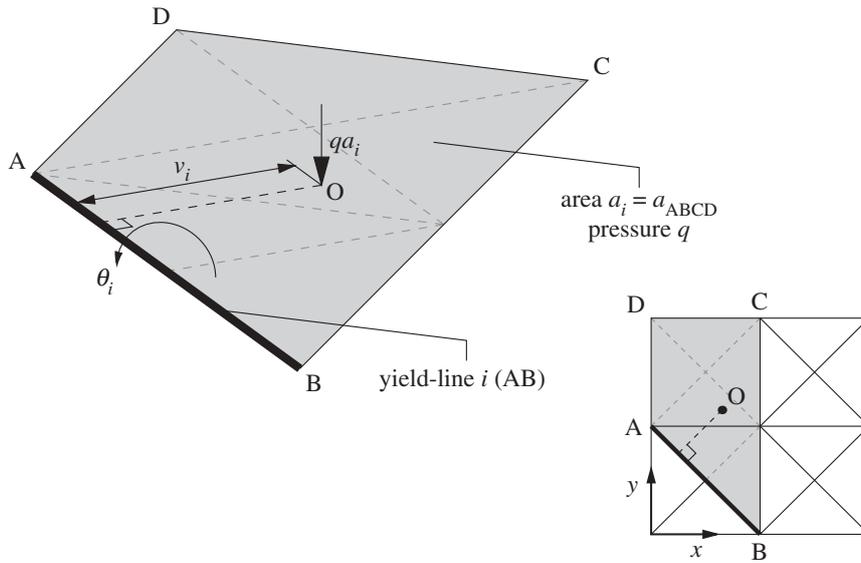


Figure 2. Strip ‘above’ potential yield-line i (AB), considered when calculating the effects of uniform live loading q (where O is the centroid of the strip).

their orientation. \mathbf{B} is a suitable $(2n \times 2m)$ compatibility matrix. The (relative) rotations along the yield-lines in \mathbf{d} are the LP variables. (Note that for convenience the terms ‘energy dissipation’ and ‘rotation’ are here used as shorthand for ‘rate of energy dissipation’ and ‘rotation rate’, respectively.)

In this problem, $\mathbf{f}_L^T \mathbf{d} = 1$ can be interpreted as the unit displacement constraint required in a standard virtual work formulation, where the coefficients in \mathbf{f}_L are a function of the external live load. This ensures that the work done by the external live load is normalized, such that only the internal work done needs to be explicitly minimized in the formulation. However, it must be borne in mind that the coefficients in \mathbf{f}_L must relate to the current problem variables, i.e. the yield-line rotations in \mathbf{d} , which are *relative* rather than absolute values. Thus, the contribution to the left-hand side of the global unit displacement constraint, $\mathbf{f}_L^T \mathbf{d} = 1$, of a given yield-line i will be

$$\mathbf{f}_{Li}^T \mathbf{d}_i = [m_{Lni} \quad -m_{Lni}] \begin{bmatrix} \theta_i^+ \\ \theta_i^- \end{bmatrix}, \quad (2.4)$$

where m_{Lni} is the moment caused by the external (unfactored) live loading on the slab. This can conveniently be calculated by considering only the effects of loads which lie in a strip of slab lying ‘above’ potential yield-line i (it is only necessary to work parallel to one co-ordinate axis, in this case the Cartesian y -axis). Thus, if it is assumed that the slab is subjected to a point load, the moment is calculated as the magnitude of the point load multiplied by the perpendicular distance to the potential yield-line. If a uniform pressure of intensity q is applied, then it can be seen that $m_{Lni} = qa_i v_i$, where a_i is the area of the strip and where v_i is the perpendicular distance to the centroid O of the strip, as indicated in figure 2. In summary, the use of *relative* rotations in the calculations means that the effect of a relative rotation at an individual discontinuity on the work done by the external live loads can readily be accounted for. Then, through summation over all discontinuities, the total work done by all external live loads can be determined, and then conveniently set to unity using the constraint $\mathbf{f}_L^T \mathbf{d} = 1$.

Table 1. Features of analogy between truss and yield-line layout optimization problems.

	truss problem	slab problem
LP problem variables	internal bar forces in \mathbf{q}	yield-line rotations in \mathbf{d}
governing coefficient matrix	equilibrium: \mathbf{B}	compatibility: \mathbf{B}
additional constraint prescribes	self-stress	unit displacement
objective function	minimize volume V	minimize work E

(e) Commentary

Layouts of bars in optimal ‘Michell’ trusses [31] form Hencky–Prandtl nets, which are orthogonal curvilinear co-ordinate systems (e.g. [32]). It has also been known for many years that, when Johansen’s square yield criterion is employed, the layouts of yield-lines in slabs also form Hencky–Prandtl nets [33]. However, prior to the studies of the present authors, the approximate-discretized solution method developed for truss layout optimization [27] had apparently not been adapted to treat slab problems. This is despite the fact that the similarity in the form of the LP problems involved was noted many years ago by Chan [14], a talented researcher at the time active in both fields at the University of Oxford. Rectifying this situation has been the main goal of this paper.

The key features of the analogy are summarized in table 1; however, with the formulation considered thus far it is for example not yet clear how more general boundary conditions (e.g. the presence of free edges) or more complex slab geometries can be handled. The applicability of the general *discontinuity layout optimization* (DLO) formulation described by Smith & Gilbert [25, 34] will therefore now be investigated.

3. Discontinuity layout optimization

(a) General formulation

The general discretized kinematic DLO problem formulation may be stated as follows (after [25]):

$$\min \quad \lambda \mathbf{f}_L^T \mathbf{d} = -\mathbf{f}_D^T \mathbf{d} + \mathbf{g}^T \mathbf{p} \tag{3.1a}$$

subject to: $\mathbf{Bd} = \mathbf{0} \tag{3.1b}$

$$\mathbf{Np} - \mathbf{d} = \mathbf{0} \tag{3.1c}$$

$$\mathbf{f}_L^T \mathbf{d} = 1 \tag{3.1d}$$

$$\mathbf{p} \geq \mathbf{0}. \tag{3.1e}$$

Or alternatively as an equivalent ‘equilibrium’ formulation (derived using duality principles—e.g. [35]) as

$$\max \quad \lambda \tag{3.2a}$$

subject to: $\mathbf{B}^T \mathbf{t} + \lambda \mathbf{f}_L - \mathbf{q} = -\mathbf{f}_D \tag{3.2b}$

$$\mathbf{N}^T \mathbf{q} \leq \mathbf{g} \tag{3.2c}$$

where λ is a dimensionless load factor, \mathbf{f}_D and \mathbf{f}_L are vectors, respectively, prescribing specified dead and live load effects, \mathbf{d} contains displacements along the discontinuities, \mathbf{B} is a suitable compatibility matrix and \mathbf{N} is a suitable flow matrix. Finally, \mathbf{p} and \mathbf{g} are vectors of plastic multipliers and their corresponding work equation coefficients and \mathbf{t} and \mathbf{q} are vectors of equivalent nodal forces and forces along discontinuities, respectively.

In the kinematic formulation, the discontinuity displacements in \mathbf{d} and the plastic multipliers in \mathbf{p} are the LP variables, whereas in the corresponding equilibrium formulation the equivalent nodal forces in \mathbf{t} , the forces along discontinuities in \mathbf{q} and the load factor λ are the LP variables.

Comparing (2.3) with (3.1), the most obvious difference is that in the latter case plastic multiplier variables have been introduced, thereby effectively decoupling the compatibility and flow constraints. A consequence of this is that when duality principles are applied to obtain the dual ‘equilibrium’ formulation, the equilibrium constraint (3.2*b*) and yield constraint (3.2*c*) are properly separated.

Given that (3.1) and (3.2) only express general relations, it is now necessary to identify appropriate variables for the slab problem now being studied, starting by considering the kinematic formulation.

(b) Kinematic formulation for slabs

Considering the kinematic problem formulation for slabs, the contributions of a given yield-line i to the global compatibility constraint equation (3.1*b*) can be written as

$$\mathbf{B}_i \mathbf{d}_i = \begin{bmatrix} \alpha_i & -\beta_i & 0 \\ \beta_i & \alpha_i & 0 \\ 0 & \frac{l_i}{2} & 1 \\ -\alpha_i & \beta_i & 0 \\ -\beta_i & -\alpha_i & 0 \\ 0 & \frac{l_i}{2} & -1 \end{bmatrix} \begin{bmatrix} \theta_{ni} \\ \theta_{ti} \\ \delta_i \end{bmatrix}, \tag{3.3}$$

where θ_{ni} , θ_{ti} and δ_i are, respectively, the normal rotation along a potential yield-line, the twisting rotation and the out-of-plane displacement (measured at the yield-line mid-point), and where α_i and β_i are x -axis and y -axis direction cosines. Note that, unlike in (2.3), the displacement variables in \mathbf{d}_i are no longer restricted to be non-negative since additional non-negative plastic multiplier variables will ensure positive energy dissipation.

Suppose that there exists no coupling between normal and twisting rotations, and between the shear displacement along a yield-line. In this case, the contributions of a given yield-line i to the global flow rule constraint (3.1*c*) can be written as

$$\mathbf{N}_i \mathbf{p}_i - \mathbf{d}_i = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} p_i^1 \\ p_i^2 \\ p_i^3 \\ p_i^4 \\ p_i^5 \\ p_i^6 \end{bmatrix} - \begin{bmatrix} \theta_{ni} \\ \theta_{ti} \\ \delta_i \end{bmatrix}. \tag{3.4}$$

However, at a typical yield-line, it can generally be assumed that the torsional (twisting) and out-of-plane displacements, θ_{ti} and δ_i , respectively, will be zero, and hence these variables can be omitted from the formulation, along with their corresponding plastic multiplier variables, p_i^3, p_i^4, p_i^5 and p_i^6 . This situation does not apply at free boundaries however, where θ_{ti} and δ_i should be free to take on arbitrary values, i.e. such variables should be added to signal the presence of such a boundary. This is because at a free boundary there is no limitation that the out-of-plane and torsional displacements must be zero, as would implicitly be the case if these terms were omitted from the formulation. (This makes the above formulation intrinsically more flexible than that considered in §2). Similarly, at a line of symmetry, δ_i should be free to take on an arbitrary value.

The objective function, (3.1*a*), and unit displacement constraint, (3.1*d*), can be formulated in a similar way to before (in §2), although now taking account of the fact that rotation normal to

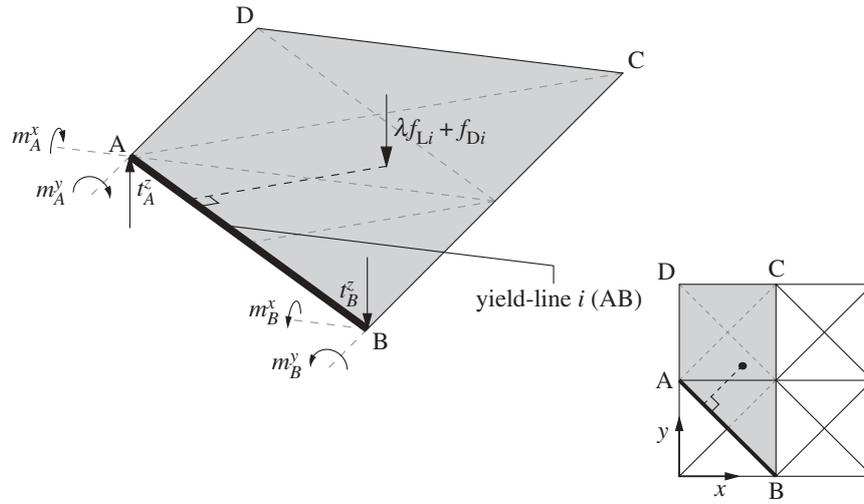


Figure 3. Nodal moments and forces at ends of yield-line i (AB), for problem shown in figure 2.

a yield-line is represented by a single unrestricted LP variable (the plastic multiplier variables in \mathbf{p} are instead now restricted to be non-negative, ensuring the plastic dissipation $\mathbf{g}^T \mathbf{p}$ is always positive; the coefficients in \mathbf{g} are as before for an internal yield-line). It should also be noted that along a free-edge (if present) $\mathbf{f}_{Li}^T = \{m_{Lni}, m_{Lti}, f_{Li}\}$, and hence values for m_{Lti} and f_{Li} will additionally need to be calculated (where f_{Li} will equal the sum of all loads lying in the slab strip 'above' yield-line i and where m_{Lti} will equal f_{Li} multiplied by the distance between the mid-point of the yield-line and the centre of the line of action of the load in the slab strip, measured parallel to the yield-line).

(c) Equilibrium formulation for slabs

Considering the equilibrium problem formulation for slabs, the required equilibrium constraint can be written for a potential yield-line discontinuity i as follows:

$$\mathbf{B}_i^T \mathbf{t}_i + \lambda \mathbf{f}_{Li} - \mathbf{q}_i = -\mathbf{f}_{Di} \quad (3.5)$$

or, in expanded form as

$$\begin{bmatrix} \alpha_i & \beta_i & 0 & -\alpha_i & -\beta_i & 0 \\ -\beta_i & \alpha_i & \frac{l_i}{2} & \beta_i & -\alpha_i & \frac{l_i}{2} \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} m_A^x \\ m_A^y \\ t_A^z \\ m_B^x \\ m_B^y \\ t_B^z \end{bmatrix} + \lambda \begin{bmatrix} m_{Lni} \\ m_{Lti} \\ f_{Li} \end{bmatrix} - \begin{bmatrix} M_{ni} \\ M_{ti} \\ S_i \end{bmatrix} = - \begin{bmatrix} m_{Dni} \\ m_{Dti} \\ f_{Di} \end{bmatrix}, \quad (3.6)$$

where m_j^x , m_j^y and t_j^z can be interpreted, respectively, as x and y direction equivalent nodal moments and out-of-plane nodal force, all acting at a given node j , and where M_{ni}, M_{ti} and S_i represent, respectively, the yield-line normal moment, torque and shear force acting on discontinuity i (figure 3). Finally, m_{Dni}, m_{Dti}, f_{Di} and m_{Lni}, m_{Lti}, f_{Li} represent the dead and live load effects acting at discontinuity i .

Now considering the contribution of a given yield-line i to the global yield constraint (3.2c), initially assuming that N_i is as defined in equation (3.4)

$$\mathbf{N}_i^T \mathbf{q}_i = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} M_{ni} \\ M_{ti} \\ S_i \end{bmatrix} \leq \begin{bmatrix} m_{pi}^+ \\ m_{pi}^- \\ m_{ti}^+ \\ m_{ti}^- \\ s_i^+ \\ s_i^- \end{bmatrix}. \quad (3.7)$$

Although at a typical yield-line, inequality equation (3.7) reduces simply to $m_{pi}^- \leq M_{ni} \leq m_{pi}^+$, by inspection it is clear that more complex yield functions could be introduced if required, for example involving interaction between the normal and torsional moments (though in doing so the traditional ‘yield-line’ character of the solution is likely to be lost, e.g. a twisting failure would lead to loss of contact between the two ends of the parts of a slab adjoining a given yield-line).

4. Extensions to the basic discontinuity layout optimization procedure

(a) Treating non-convex problem domains

Although the benchmark plane strain metal-forming and geotechnical problems considered in Smith & Gilbert [25] all had simple rectangular problem domains, real-world slab-geometries will often be considerably more complex, e.g. comprising complex non-convex problem domains. Although such geometries present no particular difficulties for conventional finite-element-based formulations, various issues arise when the DLO procedure is applied. These will now be explored.

(i) Inter-nodal connections in non-convex problem domains

Consider the non-convex slab (ABCDEFGHIJKL) shown in figure 4. If it is assumed that each node is connected to every other node by potential yield-lines, then it is evident that some potential yield-lines (e.g. the highlighted yield-line CJ in figure 4a) cross ‘free space’. To address this, it has been found to be convenient to disallow such potential yield-lines. However, since this means that a good representation of a previously well represented possible mode of response may then not be achievable (e.g. figure 4b), a finer nodal discretization can be used along all boundaries to partially compensate for this, figure 4c; consequently in all example problems considered herein the nodal spacing along boundaries has been set to be half that used within the interior of a slab.

(ii) Computing load effects in non-convex problem domains

It is also necessary to consider how the load terms in \mathbf{f}_L and \mathbf{f}_D should be computed when a non-convex slab is involved. Thus, referring to figure 4, suppose that the slab has material properties, support and loading conditions which mean that, at failure, part of the slab (CDEFGHIJKL) rotates as a rigid element about a single yield-line CL, i.e. as indicated in figure 4d. Assuming both dead and live loads are involved, it is instructive to consider how the components in \mathbf{f}_{Li} and \mathbf{f}_{Di} can be calculated for $i = CL$. In this case, as only the area shaded (CDEFIJKL) will be directly influenced by rotation along CL, only loading within this shaded area need be accounted for in the calculations. The remaining unshaded area lying ‘above’ potential yield-line CL (i.e. area FGHI) will clearly also move in the mechanism postulated, but the work associated with this movement will be accounted for through displacement along edge FG (combined translation and rotation), with the relative displacements at the edge of the slab in effect being absolute displacements.

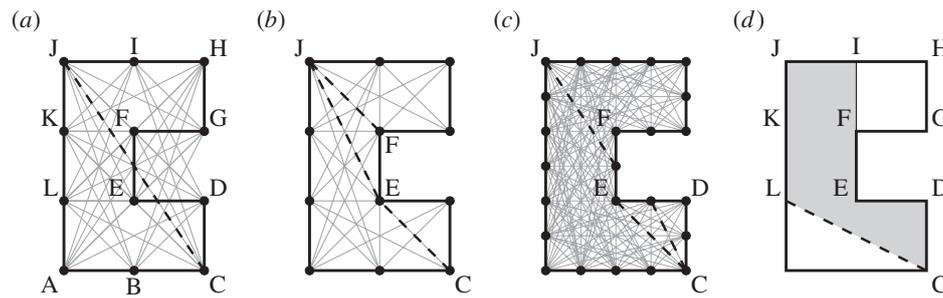


Figure 4. Slab with non-convex 'C'-shaped geometry: (a) potential yield-lines, with critical yield-line CJ which crosses domain boundaries highlighted; (b) reduced set of potential yield-lines with alternatives to CJ highlighted; (c) as (b) but with finer nodal spacing along edges and (d) shaded area to be considered when formulating f_1 and f_0 terms for potential yield-line CL.

(b) Simplifying complex yield-line patterns

It was pointed out earlier in the paper that the layouts of yield-lines in slabs will, like bars in optimal trusses, take the form of Hencky–Prandtl nets, which are orthogonal curvilinear coordinate systems. A side-effect of this is that it will frequently be found that the true critical failure mechanism will include one or more areas comprising an infinite number of infinitely short yield-lines. Although strictly speaking correct, such mechanisms do not appear to be in the spirit of the original yield-line analysis method, and the presence of large numbers of yield-lines can also make visualization of the collapse mechanism and hand checking of solutions difficult; the latter is potentially very important in engineering practice. (Furthermore, considering application to reinforced concrete slabs, cracks tend in reality to be discrete and spaced of the order of centimetres apart in yielding regions, owing to the finite tensile strength of the concrete.)

A practical means of simplifying the yield-line patterns identified is to use a coarse nodal refinement (e.g. compare the simple layout of figure 1b with that of figure 1c). However, this means that there is a danger that important detail will be lost. Thus, the efficacy of a method which involves penalizing short yield-lines in order to simplify failure mechanisms will be investigated. Such a method appears to have been first proposed by Parkes [36], though in the context of truss layout optimization.

In essence, this method only requires that $\mathbf{g}^T = \{m_{p_1}^+ l_1, m_{p_1}^- l_1, \dots, m_{p_m}^- l_m\}$ is replaced with $\hat{\mathbf{g}}^T = \{m_{p_1}^+(l_1 + k), m_{p_1}^-(l_1 + k), \dots, m_{p_m}^-(l_m + k)\}$ when formulating the optimization problem, where k is a value designed to give the desired level of simplification. Then, once the optimization process is complete, a corrected computed load factor can be obtained by back-substituting the original values from \mathbf{g} into the objective function equation (assuming the kinematic formulation is being used). The efficacy of this approach will be explored for the example problems considered in §5.

5. Examples

The procedure will now be applied to a range of isotropic slab problems previously studied in the literature, including some which have known analytical solutions.

(a) Computational issues

To obtain the solutions, a workstation equipped with an Intel Xeon E5-2670 CPU and running 64-bit CENTOS Linux was employed. The Mosek commercially available interior point LP optimizer, which uses the homogeneous and self-dual algorithm, was used [37]. The problem was initially passed to the optimizer in memory and subsequently only changes to the current problem needed to be passed to the optimizer, rather than the entire revised problem. The pre-solve feature of the

optimizer was enabled and default tolerances were used. In all cases, nodes were distributed on a uniform Cartesian grid with the specified number of nodal divisions being the number used across a specified length of the interior of a given slab. The number of nodal divisions used along exterior edges was twice that used within the slab interior, as described in §4a.

(i) Adaptive nodal connection scheme

To significantly increase the size of problem which could be solved, the adaptive nodal connection procedure, described by Gilbert & Tyas [29] for layout optimization of trusses, and in the context of DLO by Smith & Gilbert [25], was used when solving all problems. Using this procedure, it is only necessary to connect adjacent nodes with potential discontinuities initially, with additional potential discontinuities then added as required (a simple check for yield violation is carried out following an LP iteration to decide whether further potential discontinuity connections need to be added, and hence whether a further LP iteration is required). In the examples considered here, it was specified that not more than 5% of the number of connections present in the initial, adjacent connectivity, problem could be added at each iteration. Even though changes to the LP problem at each iteration might be relatively modest, with the interior point optimizer used it was not possible to use the solution from a previous iteration as a starting point for the next optimization (i.e. a 'warm start' was not used). Additionally, although the adaptive procedure is amenable to parallelization, and a parallel version of the Mosek optimizer is available, a single processor was used for all computations. The CPU times quoted include only the time to solve the LP problem(s); in practice, some additional time is required to identify candidate connections for admission at the next iteration in the adaptive solution procedure used.

(ii) Treating overlapping discontinuities

The greatest common divisor algorithm referred to in Smith & Gilbert [25] was used to remove overlapping potential discontinuities, except when the simplification algorithm outlined in §4b was used (since this requires overlapping potential discontinuities to be present in order to work effectively).

(b) Square slabs with known exact solutions

Initially consider a square slab of side length L which is subjected to uniform pressure loading q and which has a plastic moment of resistance per unit length of m_p . If the support type around the perimeter is unchanging, then symmetry conditions mean that only one-eighth of the slab needs to be modelled. DLO solutions and corresponding CPU times for slabs with fixed and simple supports are shown in table 2, for various nodal discretizations. Figure 5 shows the solution for the fixed support case when using the finest nodal discretization, involving 320 nodal divisions.

When simple supports are present the exact solution ($\lambda = 24.0(m_p/qL^2)$) can be obtained when only three nodes are present (i.e. at the corners of the portion of slab being modelled). Increasing the total number of nodes therefore does not change the solution in this case.

For the fixed support problem, it is evident from table 2 that the DLO procedure can obtain a solution which is within 0.5% of the exact analytical solution in only 2 s. This is in contrast to previously proposed automated yield-line analysis methods, which have struggled to obtain accurate solutions for this particular problem without recourse to specially tailored meshes. The best solution obtained for the fixed support problem ($42.857(m_p/qL^2)$) is just 0.015% higher than the exact solution ($42.851(m_p/qL^2)$), though in this case the CPU time required was long (912 559 s). The solutions obtained using nodal divisions of between 20 and 320 were used to calculate an extrapolated solution (refer to appendix A for details of the extrapolation method used). The extrapolated solution was found to be $42.851(m_p/qL^2)$, which matches the exact solution quoted by Fox [6] to all five significant figures, indicating that the DLO procedure can, if required, be used to obtain extremely accurate numerical solutions.

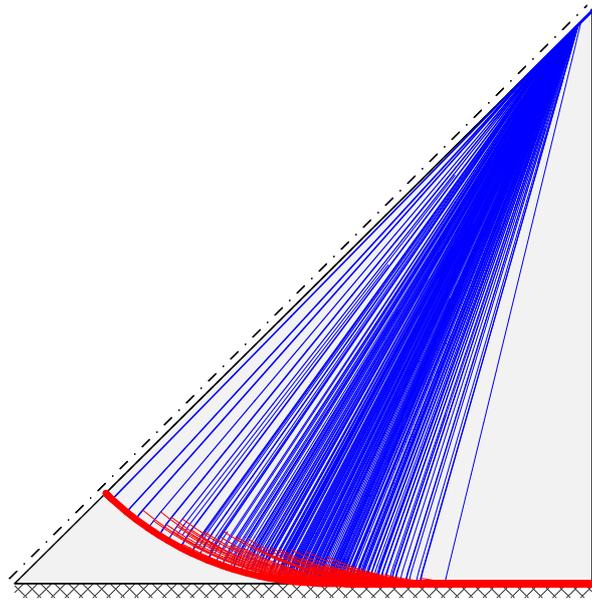


Figure 5. Square slab with fixed supports: DLO yield-line pattern (320 nodal divisions). (Online version in colour.)

Table 2. Square slabs with known exact solutions: numerical versus analytical solutions.

support type	analytical $\lambda(m_p/qL^2)$	nodal divisions ^a	numerical $\lambda(m_p/qL^2)$	error%	CPU (s)
simple	24.0	1	24.000	0.000	<1
fixed	42.851 [6]	1	48.000	12.016	<1
		20	43.055	0.476	2
		40	42.934	0.194	66
		60	42.908	0.133	278
		80	42.887	0.085	1105
		100	42.879	0.064	1704
		120	42.874	0.054	4835
		140	42.870	0.045	15 655
		160	42.868	0.040	54 949
		180	42.865	0.033	71 420
		200	42.863	0.028	276 301
		220	42.862	0.025	594 702
		240	42.861	0.023	855 442
		260	42.860	0.021	1 299 532
280	42.859	0.018	985 247		
300	42.858	0.016	1 695 220		
320	42.857	0.015	912 559		
		∞^b	42.851	0.000	—

^aNumber of divisions along each leg of the right-angled triangle domain analysed.

^bExtrapolated value (see appendix A for extrapolation procedure).

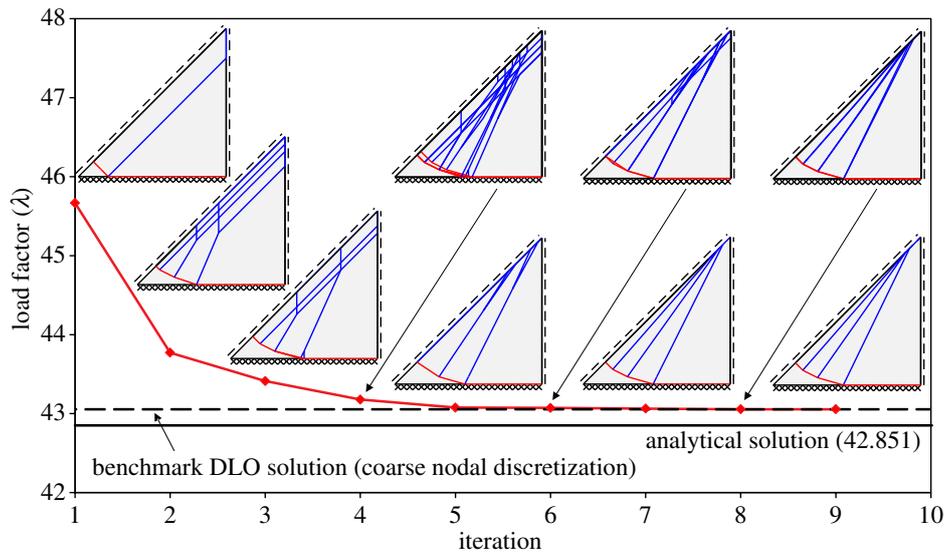


Figure 6. Square slab with fixed supports: numerical solution versus iteration when using adaptive nodal connection scheme (20 nodal divisions). (Online version in colour.)

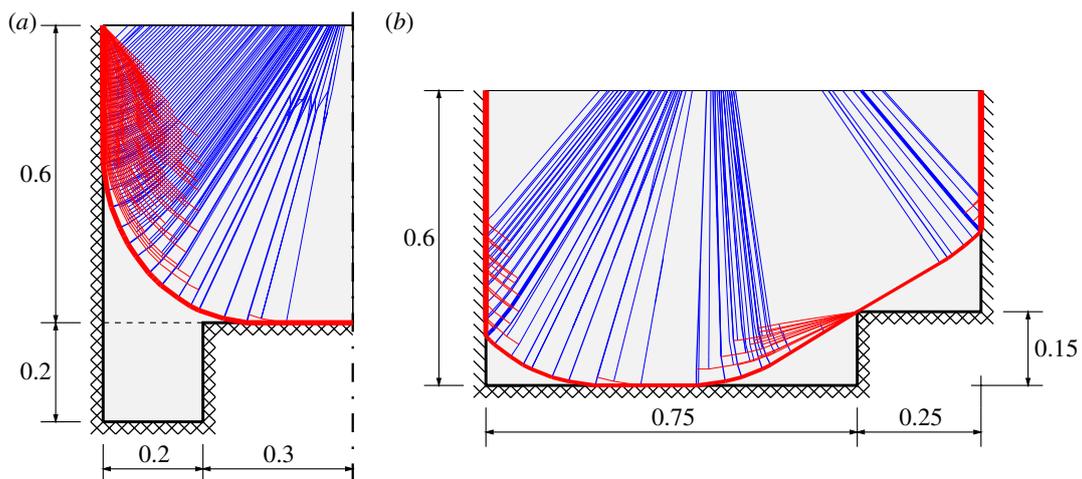


Figure 7. Regan and Yu's irregular slabs: (a) with alcoves and (b) indented, showing geometries and DLO solutions (120 nodal divisions). Simple and fixed supports are denoted, respectively, by single and cross hatches. (Online version in colour.)

Finally, figure 6 shows how the computed collapse load and associated mechanism changes as the adaptive nodal connection procedure employed proceeds, here using a coarse nodal discretization involving 20 nodal divisions for sake of clarity.

(c) Regan and Yu's irregular slabs

The next two slab problems were originally included in the book by Regan & Yu [38] and are somewhat more complex, with varying support conditions and non-convex geometries. Both the 'slab with alcoves' and 'indented slab' problems involve a pressure load of unit intensity and unit plastic moment of resistance per unit length. The geometries of the slabs and sample DLO solutions are presented in figure 7.

Table 3. Regan and Yu's irregular slabs: literature solutions versus DLO solutions.

reference	bound	nodal divisions ^a	slab with alcoves λ	indented slab λ
Regan & Yu [38]	upper	—	41.6 ^c	33.3 ^c
Johnson [39]	upper	—	37.0	32.5
Thavalingham <i>et al.</i> [18]	upper	—	35.8	29.2
Jackson [21]	upper	—	35.8	29.2
	lower	—	35.1	28.5
DLO	upper	20	35.589	29.174
	upper	40	35.411	29.062
	upper	60	35.330	29.034
	upper	80	35.305	29.014
	upper	100	35.293	29.010
	upper	120	35.279	29.002
	upper	140	35.267	28.998
	upper	160	35.262	28.995
	upper	180	35.257	28.995
	upper	200	35.254	28.992
	upper	220	35.251	28.991
	upper	240	35.250	28.990
	upper	260	35.247	28.990
	upper	280	35.245	28.989
	upper	300	35.244	28.988
	upper	320	35.243	28.988
	upper	340	35.243	—
	upper	360	35.242	—
	upper	380	35.241	—
	—	∞^b	35.230	28.980

^aNumber of divisions per unit length (i.e. the total length of each of the slabs, neglecting symmetry).

^bExtrapolated values, obtained using the 16 most refined solutions (see appendix A for extrapolation procedure).

^cComputed using the yield-line patterns shown in Regan & Yu [38]; these values are slightly lower than the simplified finite-element mesh derived solutions quoted by Johnson [39].

In table 3, solutions obtained by previous workers are presented alongside new DLO results. It is clear that even the coarsest DLO solutions presented (involving 20 nodal divisions) improve upon (i.e. are lower than) previously obtained upper-bound solutions. This is despite the fact that some of the previously obtained numerical solutions benefitted from the use of problem-specific element meshes, tailored to yield the best possible solutions. The DLO solutions are also bracketed by the upper and lower bound solutions computed by Jackson [21].

(d) Slab with hole

The final example considered comprises the irregular polygonal slab containing a hole previously analysed by Olsen [40], Krabbenhøft *et al.* [10] and others. Here, the slab is assumed to be isotropic with unit plastic moment of resistance per unit length and is subjected to a pressure load of unit intensity. The slab geometry and DLO solution are shown in figure 8. The computed

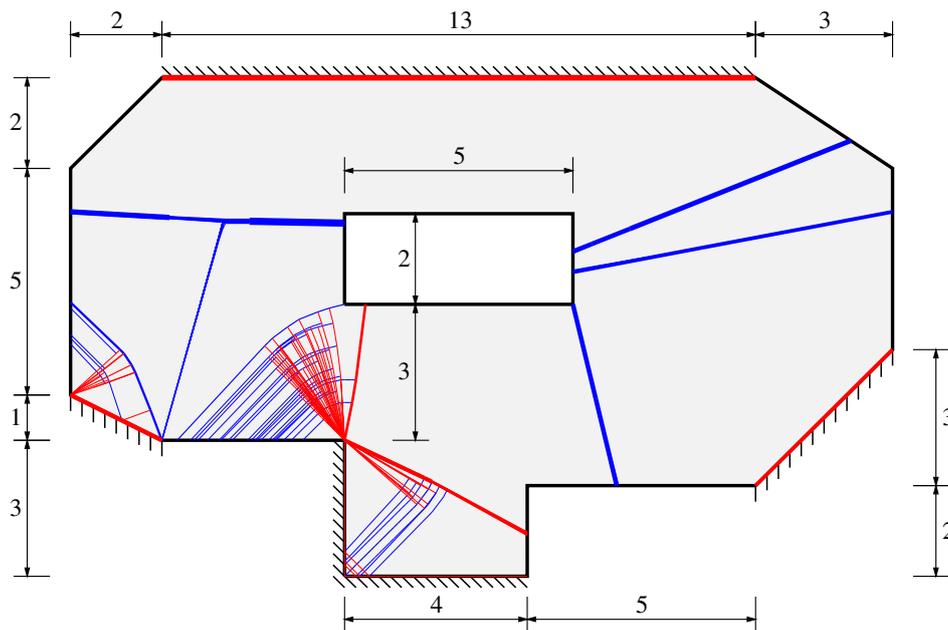


Figure 8. Slab with hole: geometry (dimensions in metre) and DLO solution (120 nodal divisions). (Online version in colour).

Table 4. Slab with hole: literature versus DLO solutions.

reference	bound	nodal divisions ^a	solution λ
Jackson [21]	upper	—	0.137
	lower	—	0.132
Krabbenhøft <i>et al.</i> [10]	lower (approx.)	—	0.135 ^b
DLO	upper	120	0.13554

^aNumber of divisions per 10 m slab length.

^bCalculated by dividing the quoted pressure load (6.75) by the quoted plastic moment of resistance (50).

DLO load factor was found to be 0.13554, which is bracketed by the upper and lower bound solutions reported by Jackson [21], as indicated in table 4. Also, the solution is 0.4% higher than the approximate lower bound solution reported by Krabbenhøft *et al.* [10]. This example demonstrates that the DLO procedure can be applied to problems with realistic geometries, something that is essential for industrial application.

(e) Simplified solutions

It is evident from the preceding examples that many of the DLO solutions identified are rather complex, and distinctly different to the ‘textbook’ yield-line solutions most practicing engineers are familiar with (for reasons which will be briefly discussed in the next section). However, by using the procedure described in §4b, simpler, more familiar looking, yield-line patterns can be generated. Sample simplified solutions for each of the examples considered are shown in figure 9; values for the simplification factor k were chosen on a case-by-case basis to provide the desired level of simplification. Figure 10 shows how the value of k influences the yield-line pattern for Regan and Yu’s indented slab example.

It is evident that simplified yield-line patterns can successfully be generated, and, although the corresponding load factors are somewhat less accurate than calculated using the standard DLO

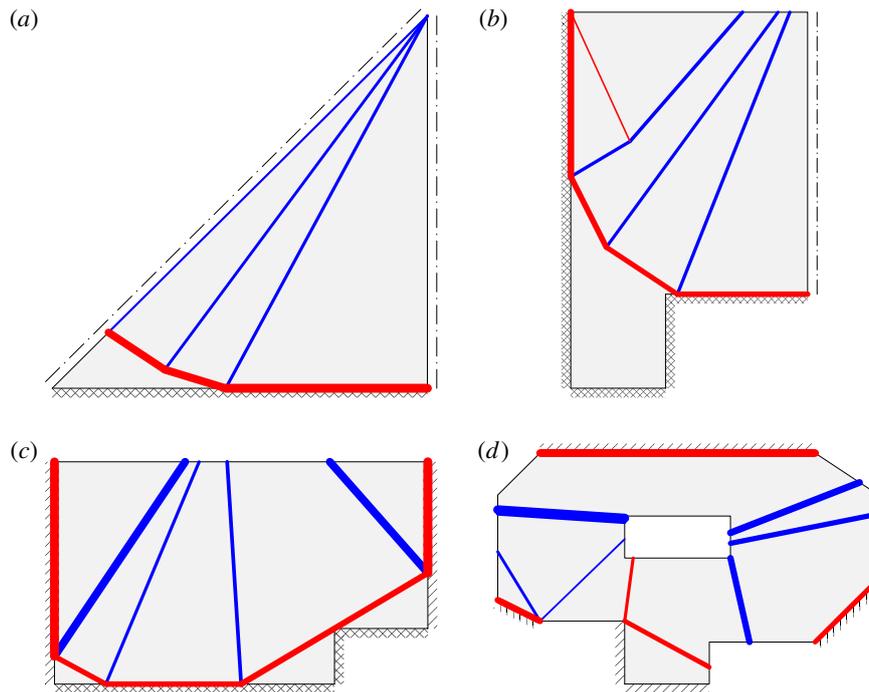


Figure 9. Simplified failure mechanisms: (a) fixed square slab (40 nodal divisions, $k = 0.005$, $\lambda = 43.080$ (diff: 0.53%)); Regan & Yu's (b) slab with alcoves (40 nodal divisions, $k = 0.02$, $\lambda = 35.852$ (diff: 1.77%)) and (c) indented slab (40 nodal divisions, $k = 0.05$, $\lambda = 29.293$ (diff: 1.08%)); (d) slab with hole (50 nodal divisions, $k = 0.5$, $\lambda = 0.13640$ (diff: 0.63%)). (Differences relative to (a) analytical solution given in table 2, (b), (c) extrapolated DLO solutions given in table 3, and (d) numerical DLO solution given in table 4.) (Online version in colour.)

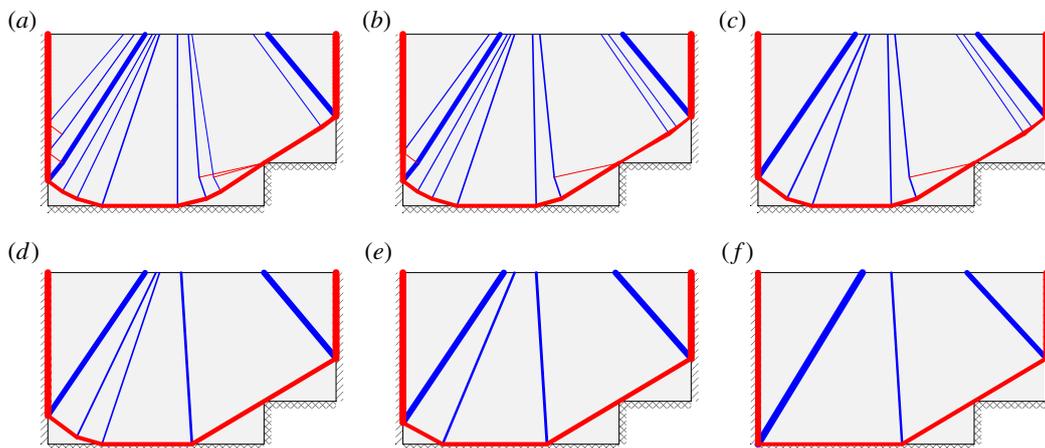


Figure 10. Regan and Yu's indented slab: influence of simplification factor k on DLO solution, using 40 nodal divisions. (a) $k = 0$, $\lambda = 29.062$ (diff: 0.28%), (b) $k = 0.001$, $\lambda = 29.067$ (diff: 0.30%), (c) $k = 0.002$, $\lambda = 29.104$ (diff: 0.43%), (d) $k = 0.005$, $\lambda = 29.205$ (diff: 0.78%), (e) $k = 0.05$, $\lambda = 29.293$ (diff: 1.08%) and (f) $k = 0.1$, $\lambda = 29.965$ (diff: 3.40%). (Differences relative to extrapolated DLO solution given in table 3.) (Online version in colour.)

procedure, they are mostly very similar, demonstrating that the load factor is often relatively insensitive to the precise form of the collapse mechanism. Also, the efficacy of the simplification technique is likely to depend on the type of problem being considered.

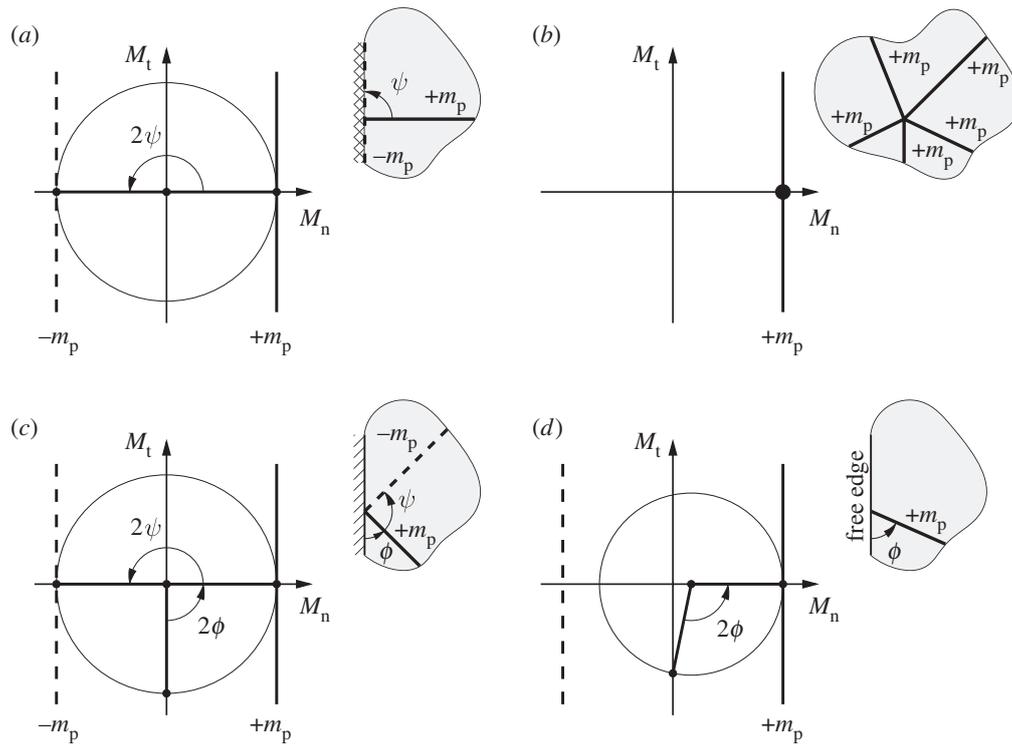


Figure 11. Use of Mohr's circles in normal moment (M_n)–torque (M_t) space to illustrate characteristic features of critical yield-line patterns in isotropic slabs: (a) orthogonal intersection of yield-lines of opposite sign, here at a fixed edge; (b) intersection of yield-lines of the same sign at arbitrary angles; (c) intersection of yield-lines of opposite sign at simple support (where $\phi = 45^\circ$ if $m_p = m_p^+ = m_p^-$) and (d) yield line intersecting a free edge (at $45^\circ \leq \phi \leq 135^\circ$ if $m_p = m_p^+ = m_p^-$).

6. Discussion

Developing a procedure to automatically identify upper bound limit analysis solutions has been of interest to researchers for many decades. In the case of slabs, a number of researchers have proposed procedures designed to improve upon the solution obtained using an initial rigid finite-element analysis (e.g. obtained using the method described by Munro & Da Fonseca [15]), by refining this in a subsequent iterative nonlinear optimization phase (e.g. [17,18]). In fact, when the adaptive nodal connection scheme described in §5a is employed, the initial solution obtained using the DLO procedure will be precisely the same as that obtained using rigid finite elements (assuming nodes are identically positioned in both cases, and assuming nearest neighbour connectivity in the case of DLO). What is new here is that when DLO is used the form of the yield-line pattern is permitted to change completely at subsequent iterations (e.g. to a fan mechanism). Additionally, the convex nature of the underlying mathematical optimization problem is preserved, and, even when the adaptive nodal connection procedure is used, the solution obtained will be globally optimal for the prescribed nodal discretization. This demonstrates that the widely held belief that recourse to nonlinear, non-convex, mathematical optimization procedures is necessary in order to directly identify critical yield-line patterns is misplaced. The DLO procedure also appears to retain much of the elegant simplicity of the original yield-line analysis method. Compared with more conventional finite-element limit analysis methods (e.g. [10]), the underlying formulation is simpler and involves only linear constraints when Johansen's square yield criterion is involved. Furthermore, visual interpretation of the output is straightforward as discrete yield-lines can clearly be identified.

High-resolution DLO solutions also allow a number of characteristic features of critical yield-line patterns in isotropic slabs to be observed, which can readily be confirmed via the use of Mohr's Circles. For example:

- yield-lines of opposite signs should intersect at 90° , whether in the interior of a slab or at a fixed support, as indicated in figure 11a;
- yield-lines of the same sign can intersect at any angle, as indicated in figure 11b;
- yield-lines of opposite signs should intersect simple supports at 45° and 135° (when $m_p = m_p^+ = m_p^-$), as indicated in figure 11c; and
- yield-lines should intersect free edges at between 45° and 135° (when $m_p = m_p^+ = m_p^-$), figure 11d. (Note that, as pointed out by Nielsen & Hoang [41], Kirchhoff boundary conditions permit a torsional moment to exist along a free edge. Thus, it is not necessary for critical yield-lines to intersect free edges at 90° , as suggested by Quintas [42]).

These characteristic features are generally not enforced when postulating simple yield-line patterns, either by hand or when using low numbers of nodes with DLO, and strictly would only be fully enforced when using an infinite number of infinitesimally spaced nodes. Since solutions generated using high numbers of nodes will often lead to highly complex patterns, a simplification procedure has also been presented, which provides a pragmatic means of identifying less complex layouts. A potential practical advantage of such layouts is that they can be used to generate traditional engineering calculations, which can readily be checked by hand by a practitioner.

7. Conclusion

- (i) In this paper, it has been demonstrated that the problem of identifying critical yield-line patterns can be formulated as a simple, albeit relatively large-scale, LP problem. This overturns the widely held belief that recourse to complex nonlinear, non-convex, mathematical optimization procedures is necessary in order to directly identify critical yield-line patterns.
- (ii) The analogy between approximate-discretized formulations for truss layout optimization and yield-line layout optimization has been established. The DLO procedure used retains much of the inherent simplicity of the traditional hand-based yield-line analysis method. Excellent agreement with known exact solutions has been obtained and improved solutions to a number of problems described in the literature have been obtained.
- (iii) Unlike previously proposed upper bound computational limit analysis methods, the DLO procedure presented can identify 'fan-type' yield-line mechanisms, as well as mechanisms of any other geometry. The procedure therefore appears to be the first *truly systematic* analysis tool capable of directly identifying yield-line patterns to have been developed to date.
- (iv) The yield-line patterns identified by the DLO procedure are often observed to be complex, containing numerous closely spaced yield-lines. However, it is shown that such complex failure mechanisms can be simplified if required (e.g. to facilitate hand-checking), albeit at the expense of some accuracy.

Acknowledgements. The authors thank Wael Darwich, Iain Haslam and Thomas Pritchard for contributing to the development of the cross-platform software framework used to solve the example problems presented, and also Will Pearson for preparing a number of the figures contained herein. The first author would also like to thank David Johnson for originally inspiring him to enter this field of study.

Funding statement. Initial work in the area of this study was undertaken while the first author was in receipt of an EPSRC Advanced Research Fellowship, grant no. GR/S53329/01.

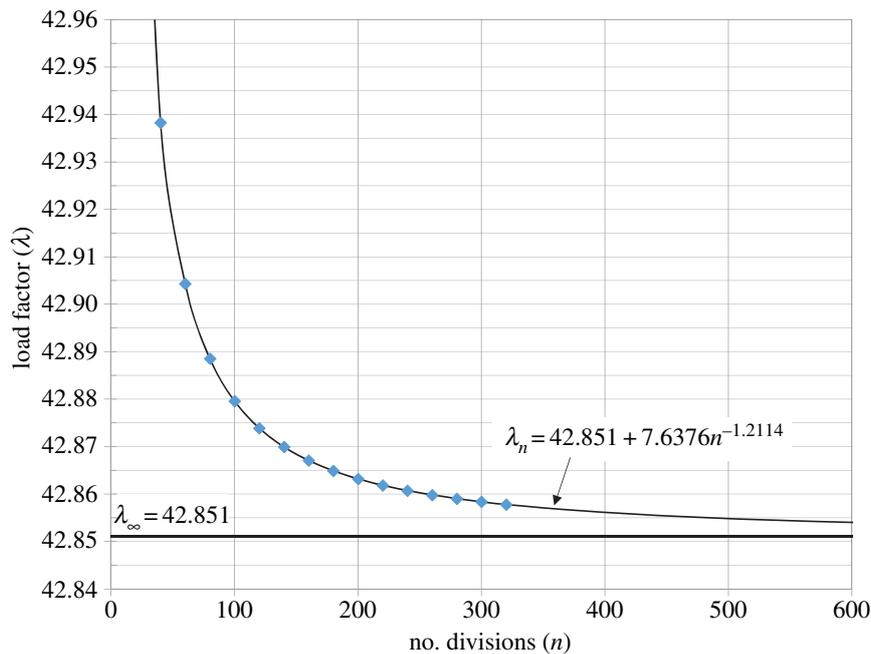


Figure 12. Square slab with fixed supports: computed load factor versus nodal refinement. (Online version in colour.)

Appendix A. Computing extrapolated load factors

In common with truss layout optimization problems (e.g. [43]), the solutions obtained using the proposed layout optimization procedure appear to follow a relation of the form:

$$\lambda_n = \lambda_\infty + kn^{-\alpha}, \quad (\text{A } 1)$$

where λ_n is the numerically computed load factor for n equally spaced nodal divisions, λ_∞ is the load factor when $n \rightarrow \infty$, and k and α are positive constants. Using (A 1), a weighted nonlinear least-squares approach can be used to find best-fit values for λ_∞ , k and α , with the weighting coefficient taken as n . For example, the resulting trend line and value for λ_∞ for the fixed edge square slab are shown in figure 12.

References

- Ingerslev A. 1923 The strength of rectangular slabs. *Struct. Eng.* **1**, 3–14.
- Johansen KW. 1962 *Yield-line theory*. London, UK: Cement and Concrete Association.
- Gvozdev A. 1960 The determination of the value of the collapse load for statically indeterminate systems undergoing plastic deformation. *Int. J. Mech. Sci.* **1**, 322–335. (doi:10.1016/0020-7403(60)90051-5)
- Save M. 1967 A consistent limit-analysis theory for reinforced concrete slabs. *Mag. Conc. Res.* **19**, 3–12. (doi:10.1680/mac.1967.19.58.3)
- Save MA, Massonnet C, Saxce Gd. 1997 *Plastic limit analysis of plates, shell and disks*, vol. 43. North-Holland Series in Applied Mathematics and Mechanics. Amsterdam, The Netherlands: Elsevier.
- Fox EN. 1974 Limit analysis for plates: the exact solution for a clamped square plate of isotropic homogeneous material obeying the square yield criterion and loaded by uniform pressure. *Phil. Trans. R. Soc. Lond. A* **227**, 121–155. (doi:10.1098/rsta.1974.0047)
- Hillerborg A. 1975 *Strip method of design*. Slough, UK: Cement and Concrete Association of Great Britain.

8. Anderheggen A, Knöpfel H. 1972 Finite element limit analysis using linear programming. *Int. J. Solids Struct.* **8**, 1413–1431. (doi:10.1016/0020-7683(72)90088-1)
9. Krabbenhøft K, Damkilde L. 2003 A general non-linear optimization algorithm for lower bound limit analysis. *Int. J. Numer. Meth. Eng.* **56**, 165–184. (doi:10.1002/nme.551)
10. Krabbenhøft K, Lyamin A, Sloan S. 2007 Formulation and solution of some plasticity problems as conic programs. *Int. J. Solids Struct.* **44**, 1533–1549. (doi:10.1016/j.ijsolstr.2006.06.036)
11. Maunder E, Ramsay A. 2012 Equilibrium models for lower bound limit analyses of reinforced concrete slabs. *Comput. Struct.* **108–109**, 100–109. (doi:10.1016/j.compstruc.2012.02.010)
12. Bleyer J, Buhan P. 2013 On the performance of non-conforming finite elements for the upper bound limit analysis of plates. *Int. J. Numer. Meth. Eng.* **94**, 308–330. (doi:10.1002/nme.4460)
13. Le CV, Gilbert M, Askes H. 2010 Limit analysis of plates and slabs using a meshless equilibrium formulation. *Int. J. Numer. Meth. Eng.* **83**, 1739–1758. (doi:10.1002/nme.2887)
14. Chan H. 1972 The collapse load of reinforced concrete plates. *Int. J. Numer. Meth. Eng.* **5**, 57–64. (doi:10.1002/nme.1620050106)
15. Munro J, Da Fonseca A. 1978 Yield line method by finite elements and linear programming. *Struct. Eng.* **56B**, 37–44.
16. Balasubramanyam KV, Kalyanaraman V. 1988 Yield line analysis by linear programming. *J. Struct. Eng.* **114**, 1431–1437. (doi:10.1061/(ASCE)0733-9445(1988)114:6(1431))
17. Johnson D. 1995 Yield-line analysis by sequential linear programming. *Int. J. Solids Struct.* **32**, 1395–1404. (doi:10.1016/0020-7683(94)00200-G)
18. Thavalingam A, Jennings A, Sloan D, McKeown J. 1999 Computer-assisted generation of yield-line patterns for uniformly loaded isotropic slabs using an optimisation strategy. *Eng. Struct.* **21**, 488–496. (doi:10.1016/S0141-0296(97)00228-9)
19. Wüst J, Wagner W. 2008 Systematic prediction of yield-line configurations for arbitrary polygonal plates. *Eng. Struct.* **30**, 2081–2093. (doi:10.1016/j.engstruct.2008.01.005)
20. Johnson D. 2007 Collapse analysis of reinforced concrete slabs: Are the up and down roads one and the same? In *Solid Mechanics and its applications: advances in engineering structures, mechanics and construction* (eds M Pandey, W-C Xie, L Xu), vol. 140, pp. 823–831. Berlin, Germany: Springer.
21. Jackson A. 2010 Modelling the collapse behaviour of reinforced concrete slabs. PhD thesis, Department of Engineering, University of Cambridge, UK.
22. Jackson AM, Middleton CR. 2013 Closely correlating lower and upper bound plastic analysis of real slabs. *Struct. Eng.* **91**, 34–40.
23. Middleton C. 1998 Concrete bridge assessment. In *Proc. Surveyor Bridge Conf.*, London, March 1998. Available at <http://www-civ-eng.cam.ac.uk/brg/papers/brgsrv98/>.
24. Kennedy G, Goodchild C. 2004 *Practical yield line design*. Crowthorne, UK: The Concrete Centre.
25. Smith C, Gilbert M. 2007 Application of discontinuity layout optimization to plane plasticity problems. *Proc. R. Soc. A* **463**, 2461–2484. (doi:10.1098/rspa.2006.1788)
26. Denton S. 2001 Compatibility requirements for yield-line mechanisms. *Int. J. Solids Struct.* **38**, 3099–3109. (doi:10.1016/S0020-7683(00)00230-4)
27. Dorn W, Gomory R, Greenberg H. 1964 Automatic design of optimal structures. *J. de Mécanique* **3**, 25–52.
28. Hemp W. 1973 *Optimum structures*. Oxford, UK: Clarendon.
29. Gilbert M, Tyas A. 2003 Layout optimisation of large-scale pin-jointed frames. *Eng. Comput.* **20**, 1044–1064. (doi:10.1108/02644400310503017)
30. Snelson K. 1965 *Continuous tension, discontinuous compression structures*. U.S. patent 3,169,611.
31. Michell A. 1904 The limits of economy of material in frame-structures. *Phil. Mag.* **8**, 589–597. (doi:10.1080/14786440409463229)
32. Strang G, Kohn R. 1983 Hencky-Prandtl nets and constrained Michell trusses. *Comput. Meth. Appl. Mech. Engng.* **36**, 207–222. (doi:10.1016/0045-7825(83)90113-5)
33. Neilson M. 1962 Theory of plasticity for reinforced slabs. PhD thesis, Technical University of Denmark.
34. Smith C, Gilbert M. 2012 *Data processing system and method*. U.S. patent 8,140,175.
35. Vanderbei R. 2008 *Linear programming: foundations and extensions*. Berlin, Germany: Springer.
36. Parkes E. 1975 Joints in optimum frameworks. *Int. J. Solids Struct.* **11**, 1017–1022. (doi:10.1016/0020-7683(75)90044-X)
37. Mosek 2009 *The MOSEK optimization tools manual*. See <http://www.mosek.com>, version 6.0 edn, Mosek ApS.

38. Regan PE, Yu CW. 1973 *Limit state design of structural concrete*. London, UK: Chatto and Windus.
39. Johnson D. 1994 Mechanism determination by automated yield-line analysis. *Struct. Eng.* **72**, 323–327.
40. Olsen PC. 1998 The influence of the linearisation of the yield surface on the load-bearing capacity of reinforced concrete slabs. *Comput. Method. Appl. M.* **162**, 351–358. (doi:10.1016/S0045-7825(97)00352-6)
41. Nielsen MP, Hoang LC. 2010 *Limit analysis and concrete plasticity*. Boca Raton, FL: Taylor & Francis.
42. Quintas V. 2003 Two main methods for yield line analysis of slabs. *J. Eng. Mech.* **129**, 223–231. (doi:10.1061/(ASCE)0733-9399(2003)129:2(223))
43. Darwich W, Gilbert M, Tyas A. 2010 Optimum structure to carry a uniform load between pinned supports. *Struct. Multidiscip. Optimiz.* **42**, 33–42. (doi:10.1007/s00158-009-0467-0)

Appendix B

The yield-line method for concrete slabs: automated at last^[i]

^[i]Gilbert, M. He, L and Pritchard, T. (2015), 'The yield-line method for concrete slabs: automated at last', *The Structural Engineer*, **93**, 44-48.

The yield-line method for concrete slabs: automated at last

Matthew Gilbert BEng, PhD, CEng, MICE, MASCE
University of Sheffield

Linwei He BEng, MSc
University of Sheffield

Thomas Pritchard MEng, PhD
LimitState Ltd

Synopsis

The yield-line method of analysis provides a powerful means of identifying the ultimate load-carrying capacity of reinforced concrete slabs. Benefits of the yield-line method are that it will often identify additional reserves of strength when applied to the analysis of existing slabs, and to highly economic slabs when used in design. Traditionally a hand-based method, the yield-line method is easy to apply to problems involving simple slab geometries and loading regimes. However, when these become more complex it can be difficult to identify the critical yield-line pattern.

To address this, the method has now been systematically automated.

The automated method quickly identifies the critical mechanism (or a close approximation of this) and corresponding load-carrying capacity, providing engineers with a powerful new computer-based tool for the analysis and design of concrete slabs. In this article, the discontinuity layout optimisation (DLO) procedure which has been used to automate the yield-line method is briefly described and then applied to various example problems.

Introduction

Reinforced concrete slabs are a feature of many modern buildings and bridges. When designing or assessing a reinforced concrete slab, elastic analysis methods have become popular in recent years, largely due to the availability of efficient computer-based implementations (e.g. using grillage analysis or finite element analysis techniques). Elastic methods are therefore now often used both to estimate slab deflections under service loads (to establish the serviceability limit state (SLS))

and to analyse a slab at failure (to establish the ultimate limit state (ULS)).

However, a standard elastic analysis does not take account of the redistribution of moments that takes place after yielding of the reinforcement in a slab. This means that an elastic analysis may provide a grossly over-conservative estimate of ULS capacity. In cases when the ULS is critical, this is likely to lead to more material (i.e. more concrete and/or steel reinforcement) being specified in a design than is necessary. To address this, a non-linear analysis (e.g. a

non-linear finite element analysis) could be performed; however, this type of analysis tends to be demanding in terms of operator expertise and computer resources, and is generally not considered suitable for routine use. Alternatively, a much simpler plastic analysis method, such as the yield-line method, could be used. However, the lack of an efficient computer-based implementation of the yield-line method has reduced its popularity in recent years.

The term 'yield-line' was first coined by Ingerslev¹ in the very first article to appear in *The Structural Engineer* in 1923. Subsequently, Johansen² developed the theory underpinning the general yield-line method, later shown to be an 'upper bound' plastic analysis method, within the context of the then emerging plastic theorems³. Since a reinforced concrete slab generally contains a low percentage of reinforcement, the section will generally yield in flexure in a ductile manner, thereby justifying the use of plastic methods. Benefits of the yield-line method are that it will often identify additional reserves of strength when applied to the analysis of existing slabs, and to highly economic slabs when used in design⁴.

The traditional hand-based method involves postulating a yield-line pattern (failure mechanism) and then using the work method to compute the corresponding load-carrying capacity. However, due to the upper-bound nature of the yield-line method, a range of yield-line patterns will often need to be explored, which can be time-consuming. Furthermore, there is often the concern that the critical pattern may have been missed, and consequently that an unsafe estimate of load-carrying capacity has been computed. This has prompted many practitioners to turn to computer-based elastic methods, which provide demonstrably safe, albeit frequently over-conservative, ULS predictions.

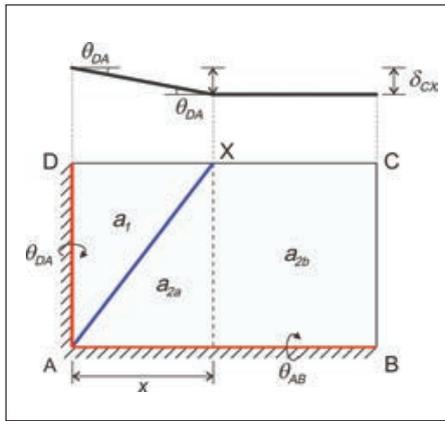


Figure 1 Example yield-line pattern for rectangular slab with simple supports and free edges

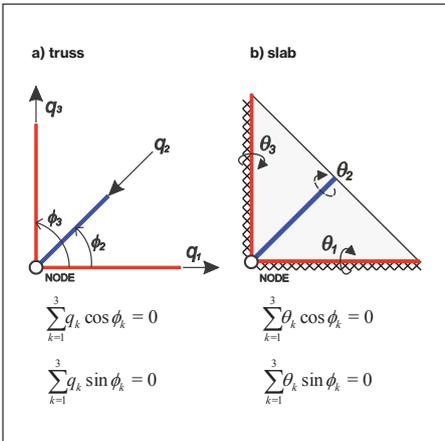


Figure 2 Analogy between a) truss equilibrium and b) slab compatibility at node

However, the yield-line analysis method has recently been systematically automated, thereby allowing the critical yield-line pattern (or a very close approximation of this) to be reliably found. This means that practitioners can now apply the yield-line method with confidence, even when slabs with complex geometries and/or loading regimes are involved. However, since the yield-line method considers only flexural failure, additional checks (e.g. for punching shear failure and/or SLS deflections) will still be required.

This article briefly describes the traditional process of yield-line analysis and then explains how it has been possible to systematically automate this. The new automated method is then applied to various practical example problems.

Traditional hand-based yield-line analysis

The yield-line method can straightforwardly be applied to problems involving simple slab geometries and loading regimes. The first step is to postulate a yield-line pattern, following basic rules to ensure that this is geometrically compatible (see, for example, Kennedy and Goodchild⁴). Figure 1 shows a sample yield-line pattern for a reinforced rectangular slab with two simple supports and two free edges, and subject to uniform pressure loading.

The second step involves performing calculations to determine the load or load factor required to cause collapse (or, in the case of design, the moment capacity required to support the applied load). The work method is commonly used to do this; in this case the mechanism is perturbed and external work done by applied loads is

equated to the internal work done along yield-lines:

External work (E) = Internal work (dissipation D)

$$\lambda \times \sum_{\text{all slab regions}} q \times a \times \delta(\theta) = \sum_{\text{all yield-lines}} m_p \times l \times |\theta| \tag{1}$$

Where λ is a load factor, to be determined, q is the specified pressure loading per unit area, a is the area of a given rigid slab region, $\delta(\theta)$ is the displacement of the centroid of this slab region, which can be expressed as a function of yield-line rotations θ . Also m_p is the plastic moment of resistance per unit length of the slab, and l is the length of a given yield-line.

For the example shown in Fig. 1, assuming lengths $AB = 9\text{m}$, $BC = 6\text{m}$, and isotropic moment of resistance m_p , sample calculations are given as follows:

External work:

$$E = \lambda q (a_1 \delta_1 + a_{2a} \delta_{2a} + a_{2b} \delta_{2b})$$

$$= \lambda q \theta_{AB} l_{BC} (a_1 / 3 + a_{2a} / 3 + a_{2b} / 2)$$

$$= \lambda q \theta_{AB} (162 - 6x) \tag{2}$$

Internal work:

$$D = m_p l_{DA} \theta_{DA} + m_p x \theta_{AB} = m_p \theta_{AB} (36 / x + x) \tag{3}$$

To determine the minimum load factor λ , the critical value of x must be found. For small problems this can be done by calculus or by trial and error (e.g. for this example x can be found to be 4.813m and, when $m_p = 20\text{kNm/m}$ and $q = 1\text{kN/m}^2$, the computed load factor can be calculated to be 1.847). For larger problems, mathematical optimisation can be used; in this case it is usual to pose the problem in a slightly different way, setting the unfactored external work done by the external loads to unity, giving the following mathematical optimisation problem:

minimise $\lambda = \sum m_p l (\theta^+ + \theta^-)$

subject to $\sum q a \delta(\theta) = 1$ (4)

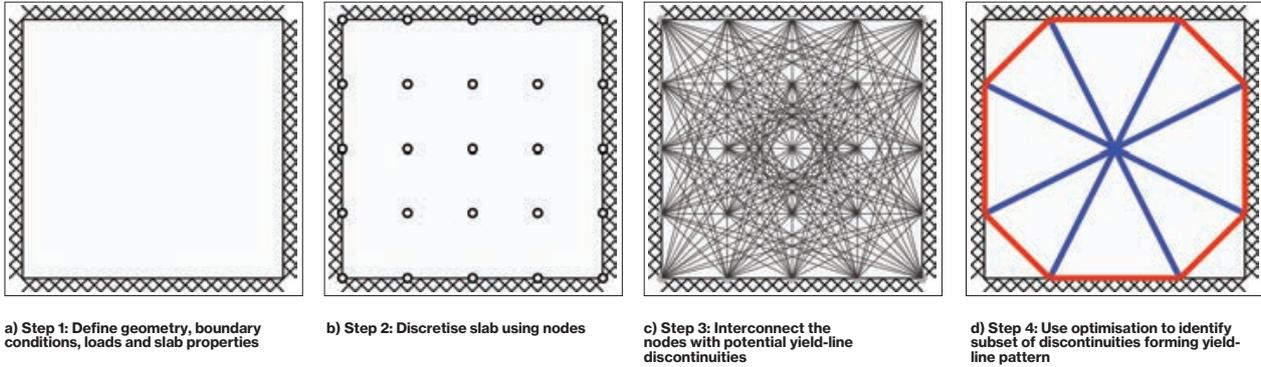
Equation 4 is equivalent to Equation 1, but with the additional stipulation that the value of λ is to be minimised. Also, to ensure positive dissipation along yield-lines, new variables θ^+ , θ^- have been introduced, where $\theta = \theta^+ - \theta^-$, and where θ^+ , $\theta^- > 0$. This modified form will be used in the next section.

New automated method

In the simple yield-line analysis problem considered in the previous section, a geometrically compatible yield-line pattern was predefined, and all that was required was to adjust the geometry of the mechanism (i.e. the distance x) to determine the critical case. However, in general, the critical yield-line pattern will not be known in advance, and the challenge is to identify this (or a good approximation of this) from a large set of possible geometrically compatible patterns.

One possible approach is to discretise the slab using rigid finite elements, with potential yield-lines lying along element boundaries^{5,6}. However, with this approach the set of geometrically

Figure 3
Steps in automated yield-line analysis procedure



compatible patterns from which the most critical can be chosen will be relatively small. Also, specially tailored meshes must be used in order to, for example, identify fan type mechanisms, which is clearly unsatisfactory.

An alternative approach involves considering the yield-line discontinuities directly, and enforcing the geometric compatibility requirement at the end points of yield-lines (nodes). Denton⁷ demonstrated that compatibility can be enforced for a yield-line mechanism in essentially the same way as equilibrium is enforced in a truss (Figure 2). Gilbert *et al.*⁹ then showed that the long-established 'layout optimisation' technique, used to identify the optimum topology of a truss, could also be used to identify the critical yield-line pattern, and corresponding collapse load (or load factor) of a slab. Steps in the process are shown in Figure 3.

For a problem comprising n nodes and m potential yield-lines, the resulting 'discontinuity layout optimisation' (DLO) formulation can be written as:

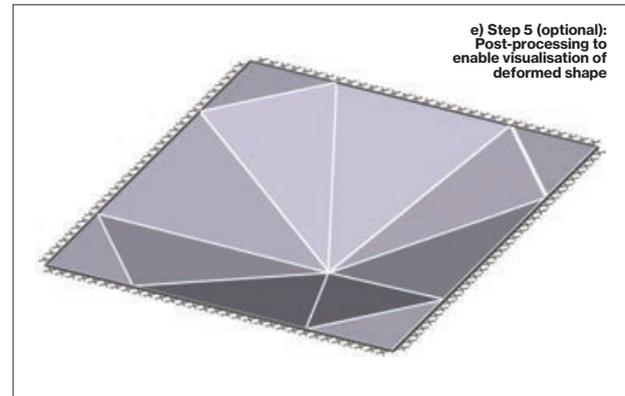
$$\text{minimise } \lambda = \sum_{j=1}^m m_j (\theta_j^+ + \theta_j^-) \quad (5a)$$

$$\text{subject to } \left. \begin{aligned} \sum_{k=1}^m \theta_k \cos \phi_k &= 0 \\ \sum_{k=1}^m \theta_k \sin \phi_k &= 0 \end{aligned} \right\} \text{ for each node } i = 1, \dots, n \quad (5b)$$

$$\sum_{j=1}^m q a_j \delta_j(\theta_j) = 1 \quad (5c)$$

Where Equation 5b enforces for each node the geometric compatibility constraint shown in Fig. 2 (assuming there are m_i yield-line connections at node i). Also, Equation 5c enforces the unit external work constraint (see Gilbert *et al.*⁹ for further details). This is a linear optimisation problem for which highly efficient solvers exist. This means that problems involving thousands of nodes can be solved in a matter of seconds on a modern desktop PC.

An interesting feature of the DLO procedure is that, at points where potential yield-line discontinuities crossover one another (see instances of this in Fig. 3c), compatibility requirements are



implicitly enforced. Also, since the location of each potential yield-line is known in advance of the optimisation process, it is possible to locally ascribe m_p values, making it straightforward to model slabs with orthotropic or skew reinforcement.

Example 1: Benchmark square slabs

To demonstrate the effectiveness of the automated method it has first been applied to various benchmark square slab problems, for which known solutions are available. In each case, the commercially available LimitState:SLAB software⁹, which implements the DLO formulation already outlined, was employed. Although some of these problems have been found to be difficult to solve when using rigid finite elements, here solutions well within 1% of the known values were obtained within a few seconds on a modern desktop PC.

Considering first the case of a slab with uniform pressure load and simple supports, Figure 4a shows that the familiar 'X' shaped yield-line pattern has been identified as being critical. In this case the exact load factor of 24 ($\times m_p / qL^2$, where L is the side length of the slab) is obtained even when very small numbers of nodes are employed. (Though note that a more complex pattern, incorporating corner fans, is identified when the slab is provided with only bottom reinforcement.)

Fig. 4b shows the identified yield-line pattern for the uniform pressure load with fixed supports case (also considered in Fig. 3). Here a load factor of 43.052 is obtained, which is quite close to the exact load factor¹⁰ of 42.851. Alternatively, an even closer value can

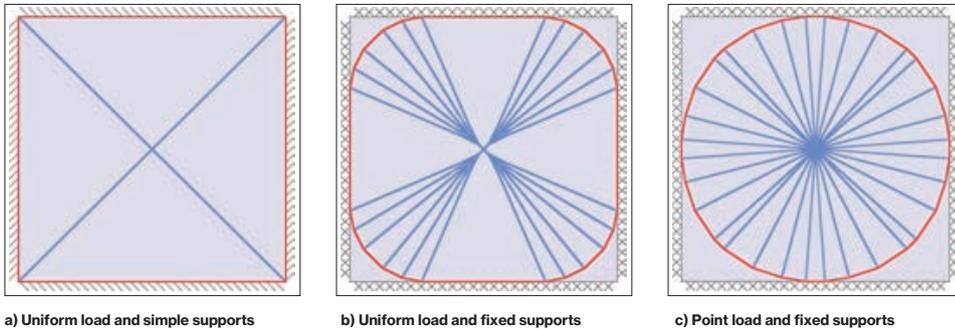


Figure 4 Identified yield-line patterns for square slabs

be obtained simply by using more nodes (e.g. a solution of 42.857 was reported by Gilbert *et al.*⁸). Note that in this case the identified yield-line pattern is somewhat more complex than the patterns typically considered in a hand analysis. This is partly because in a critical yield-line pattern positive and negative yield-lines will be orthogonal to each other, something that, for sake of simplicity, is often ignored in a hand analysis. In addition, the use of a fixed nodal grid in the DLO method means that a single yield-line in the true critical yield-line pattern may be approximately represented by several yield-lines in close proximity to one-another.

Finally, Fig. 4c shows the identified yield-line pattern for the central point load with fixed supports case; in this case the computed load factor of $12.624 (\times m_p/Q,$

where Q is the magnitude of the point load) is close to the exact load factor² of 4π .

Example 2: Building with irregular floor plate

Hand-based yield-line analysis becomes particularly problematic when complex slab geometries are involved. Kennedy and Goodchild⁴ provide useful advice on the types of mechanism that should be considered, though to account for the increased level of uncertainty involved they recommend that the moment capacity be increased by 15% for the purposes of design, rather than their normal recommended value of 10% (which itself has recently been challenged¹¹).

An example considered by Kennedy and Goodchild⁴ is the relatively complex floor plate of a London apartment block

(Figure 5). The irregular geometry requires that many possible yield-line patterns are considered by hand, which is a time-consuming process. Conversely, with the new automated method it is possible to quickly obtain a close approximation of the critical yield-line pattern and associated load factor (Fig. 5b).

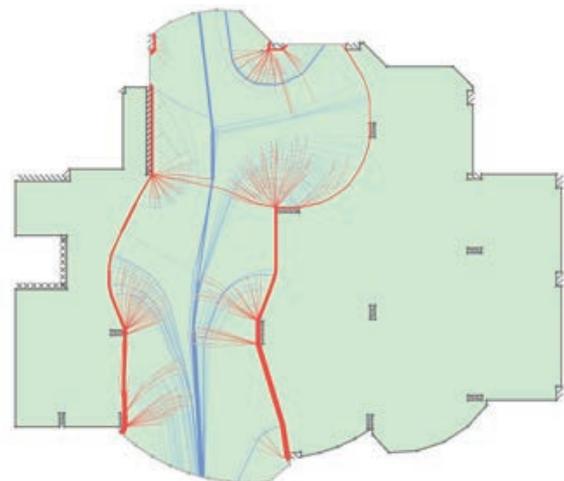
Example 3: Beam-and-slab bridge deck

Some years ago Middleton¹² suggested that many highway bridges have a low assessed load-carrying capacity, not because of inherent weakness, but due to the conservative nature of the elastic methods used to assess them. In a study of 21 local authority bridges initially assessed to have a capacity of less than 17t, it was found that over 80% had the capacity to carry at least

Figure 5 Building with irregular floor plate

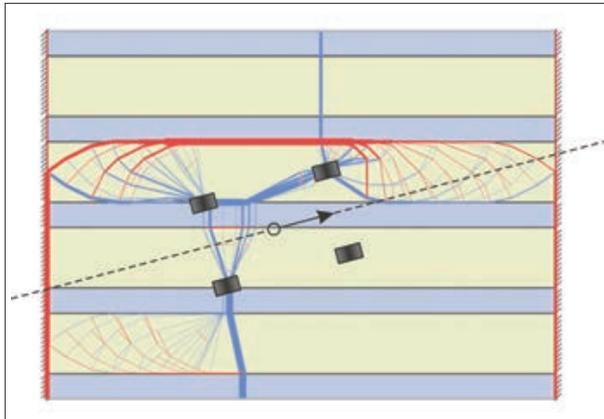


a) External view

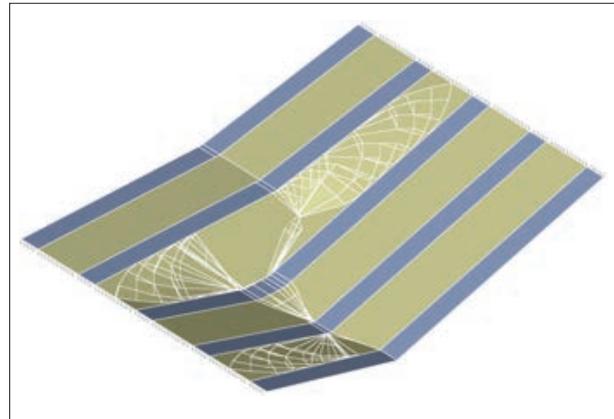


b) Identified yield-line pattern (obtained assuming simple supports at edges of blade columns)

Figure 6
Beam-and-slab bridge deck



a) Identified yield-line pattern at critical vehicle position



b) Corresponding deformed shape

38t vehicles when assessed using plastic (yield-line) methods. To facilitate rapid assessment of such bridges, a practical software tool, COBRAS¹³, was developed. However, although COBRAS considers a relatively large number of predefined yield-line patterns, there is still a concern that the critical mechanism may be missed. (To address this, Jackson and Middleton¹⁴ recently developed a more general plastic analysis procedure; however, obtaining the critical yield-line pattern necessitated a manual interpretation step.)

Figure 6 shows an example of a four-

wheeled vehicle traversing a beam-and-slab bridge deck, in this case assuming that the wheels act as point loads, and that both beams and slabs behave in a plastic manner. Using the new automated method, it is evident that a relatively complex yield-line pattern, of the sort that would be very unlikely to be found by hand, is identified as being critical.

Conclusions

The yield-line method provides a powerful means of analysing the ultimate (collapse) limit state. Benefits of the yield-line method

are that it will often identify additional reserves of strength when applied to the analysis of existing slabs, and to highly economic slabs when used in design. However, the lack of a general computer-based implementation has limited its popularity in recent years. To address this, the yield-line method has now been systematically automated, using DLO. This provides structural engineers with a viable alternative to the elastic analysis methods that have become prevalent for collapse analysis in recent years, which can yield excessively conservative results.

References

- ▶ 1 Ingerslev A. (1923) 'The strength of rectangular slabs', *The Structural Engineer*, 1 (1), pp. 3–14
- ▶ 2 Johansen K. W. (1943) *Brudlinieteorier*, Copenhagen, Denmark: Gjellerup Forlag (English translation: Johansen K. W. (1962) *Yield-Line Theory*, London, UK: Cement and Concrete Association)
- ▶ 3 Braestrup M. W. (1970) 'Yield-line theory and limit analysis of plates and slabs', *Mag. Concr. Res.*, 22 (71), pp. 99–106
- ▶ 4 Kennedy G. and Goodchild C. H. (2004) *Practical yield line design*, Camberley, UK: The Concrete Centre
- ▶ 5 Chan H. S. Y. (1972) 'The collapse load of reinforced concrete plates', *Int. J. Numer. Meth. Eng.*, 5 (1), pp. 57–64
- ▶ 6 Munro J. and Da Fonseca A. M. A. (1978) 'Yield-line method by finite elements and linear programming', *The Structural Engineer*, 56 (14), pp. 37–44
- ▶ 7 Denton S. R. (2001) 'Compatibility requirements for yield-line mechanisms', *Int. J. Solids Struct.*, 38, pp. 3099–3109
- ▶ 8 Gilbert M., He L., Smith C. C. and Le C. V. (2014) 'Automatic yield-line analysis of slabs using discontinuity layout optimization', *Proc. Roy. Soc. A*, 470 (2168)
- ▶ 9 LimitState Ltd (2015) *LimitState:SLAB, version 1.0* [Online] Available at: www.limitstate.com/slab (Accessed: September 2015)
- ▶ 10 Fox E. N. (1974) 'Limit analysis for plates: the exact solution for a clamped square plate of isotropic homogeneous material obeying the square yield criterion and loaded by uniform pressure', *Phil. Trans. Roy. Soc. A*, 277 (1265), pp. 121–155
- ▶ 11 Ramsay A., Maunder E. and Gilbert M. (2015) 'Yield-line analysis: is the 10% rule safe?', *NAFEMS Benchmark*, January, pp. 15–20
- ▶ 12 Middleton C. R. (1998) 'Concrete bridge assessment', *Proc. Surveyor Bridge Conf.*, London, UK, March
- ▶ 13 Cambridge Enterprise Ltd (2007) *COBRAS, version 1.7* [Online] Available at: www.cobras.co.uk/ (Accessed: September 2015)
- ▶ 14 Jackson A. M. and Middleton C. R. (2013) 'Closely correlating lower and upper bound plastic analysis of real slabs', *The Structural Engineer*, 91 (1), pp. 34–40

Appendix C

Second derivative terms in geometry optimization for slab

When the optimization variables are arranged as $[x_A, y_A, x_B, y_B, \theta_n, \theta_t, \delta, p^+, p^-]$, the Hessian matrix of the objective function $\mathbf{g}_i^T \mathbf{p}_i$ is derived as:

$$\mathbf{H}_{\mathbf{g}_i^T \mathbf{p}_i} = \begin{bmatrix} \frac{y_l^2 E_0}{l^4} & -\frac{x_l y_l E_0}{l^4} & -\frac{y_l^2 E_0}{l^4} & \frac{x_l y_l E_0}{l^4} & \vdots & -\frac{m_p^+ x_l}{l} & -\frac{m_p^- x_l}{l} \\ -\frac{x_l y_l E_0}{l^4} & \frac{x_l^2 E_0}{l^4} & \frac{x_l y_l E_0}{l^4} & -\frac{x_l^2 E_0}{l^4} & \vdots & -\frac{m_p^+ y_l}{l} & -\frac{m_p^- y_l}{l} \\ -\frac{y_l^2 E_0}{l^4} & \frac{x_l y_l E_0}{l^4} & \frac{y_l^2 E_0}{l^4} & -\frac{x_l y_l E_0}{l^4} & \vdots & \frac{m_p^+ x_l}{l} & \frac{m_p^- x_l}{l} \\ \frac{x_l y_l E_0}{l^4} & -\frac{x_l^2 E_0}{l^4} & -\frac{x_l y_l E_0}{l^4} & \frac{x_l^2 E_0}{l^4} & \vdots & \frac{m_p^+ y_l}{l} & \frac{m_p^- y_l}{l} \\ \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{4 \times 3} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{3 \times 4} & \vdots & \vdots \\ -\frac{m_p^+ x_l}{l} & -\frac{m_p^+ y_l}{l} & \frac{m_p^- x_l}{l} & \frac{m_p^- y_l}{l} & \vdots & \mathbf{0}_{5 \times 5} & \vdots \\ -\frac{m_p^- x_l}{l} & -\frac{m_p^- y_l}{l} & \frac{m_p^+ x_l}{l} & \frac{m_p^+ y_l}{l} & \vdots & \vdots & \vdots \end{bmatrix}, \quad (\text{C.1})$$

where $E_0 = l(m_p^+ p^+ + m_p^- p^-)$ is the internal energy dissipation associated with the given yield-line.

Considering the compatibility constraint (5.2), each yield-line contributes to six equality constraints, and the Hessian matrix for each can be derived separately. In addition, as \mathbf{B}_i was divided into two parts, \mathbf{B}_i^I and \mathbf{B}_i^{II} , the six Hessian matrices of the first part $\mathbf{B}_i^I \mathbf{d}_i$ contain only the following two terms: $\nabla^2(\theta_n \cos \phi)$ and $\nabla^2(\theta_n \sin \phi)$. The first term $\nabla^2(\theta_n \cos \phi)$ can be found to be:

$$\frac{\partial^2 \Gamma_\alpha}{\partial x_A^2} = -2 \frac{\partial \Lambda_\alpha}{\partial x_A} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial x_A^2} dx, \quad (\text{C.6a})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_A \partial x_A} = -\frac{\partial \Lambda_\alpha}{\partial y_A} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_A \partial x_A} dx, \quad (\text{C.6b})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_A^2} = \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_A^2} dx, \quad (\text{C.6c})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial x_B \partial x_A} = \frac{\partial \Lambda_\alpha}{\partial x_A} - \frac{\partial \Lambda_\alpha}{\partial x_B} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial x_B \partial x_A} dx, \quad (\text{C.6d})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial x_B \partial y_A} = \frac{\partial \Lambda_\alpha}{\partial y_A} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial x_B \partial y_A} dx, \quad (\text{C.6e})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial x_B^2} = 2 \frac{\partial \Lambda_\alpha}{\partial x_B} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial x_B^2} dx, \quad (\text{C.6f})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_B \partial x_A} = -\frac{\partial \Lambda_\alpha}{\partial y_B} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_B \partial x_A} dx, \quad (\text{C.6g})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_B \partial y_A} = \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_B \partial y_A} dx, \quad (\text{C.6h})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_B \partial x_B} = \frac{\partial \Lambda_\alpha}{\partial y_B} + \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_B \partial x_B} dx, \quad (\text{C.6i})$$

$$\frac{\partial^2 \Gamma_\alpha}{\partial y_B^2} = \int_{x_A}^{x_B} \frac{\partial^2 \Lambda_\alpha}{\partial y_B^2} dx. \quad (\text{C.6j})$$

$$(\alpha = x, y, z)$$

Note that Γ_z is not a function of the displacement variables. The Hessian matrix of Γ_α ($\alpha = x, y, z$) can now readily be obtained, as can the full expression for (C.4). For instance, considering (C.5), the fifth term $\nabla^2 (\Gamma_z \delta)$ can be written as:

$$\begin{aligned} \nabla^2 (\Gamma_z \delta) &= \Gamma_z \nabla^2 \delta + \nabla \delta \nabla^T \Gamma_z + \nabla \Gamma_z \nabla^T \delta + \delta \nabla^2 \Gamma_z \\ &= \nabla \delta \nabla^T \Gamma_z + \nabla \Gamma_z \nabla^T \delta + \delta \nabla^2 \Gamma_z \\ &= \left[\begin{array}{c|cc} \delta \nabla_{4 \times 4}^2 \Gamma_z & \mathbf{0}_{4 \times 2} & \nabla_{4 \times 1} \Gamma_z \quad \mathbf{0}_{4 \times 2} \\ \hline \mathbf{0}_{2 \times 4} & & \\ \nabla_{1 \times 4}^T \Gamma_z & & \mathbf{0}_{5 \times 5} \\ \mathbf{0}_{2 \times 4} & & \end{array} \right], \end{aligned} \quad (\text{C.7})$$

where $\nabla_{4 \times 4}^2 \Gamma_z$ is a 4×4 Hessian matrix of Γ_z with respect to $x_A, y_A, x_B,$ and y_B . In addition, $\nabla_{4 \times 1} \Gamma_z$ is the gradient of Γ_z calculated using (5.15).

Appendix D

Domain decomposition techniques for slab

Complex slab geometries are frequently encountered in practice. To treat these it is convenient to divide a slab domain into several simpler sub-domains. A particular example is shown in Fig. D.1. Here the slab is divided into three sub-domains, Ω_1 , Ω_2 , and Ω_3 . The approach taken will be to split any yield-line which intersects a sub-domain boundary into two. Also, each sub-domain can be deemed to be a separate slab, having the following characteristics:

- Matrices associated with each sub-domain are established locally (e.g., when considering live load effects, the geometry of the sub-domain is used).
- If a sub-domain boundary coincides with a boundary of the original slab, the original boundary condition is used.
- Internal boundaries are considered as free edges and their internal energy dissipation is thus not taken into account when setting up sub-problems.

However, clearly the sub-problems must be linked to properly represent the original problem; this is achieved by linking yield-lines on internal boundaries. Thus at an internal boundary yield-lines at the edge of each sub-domain are duplicated, coinciding in position, but belonging to different sub-problems. Although normally the displacements at yield-lines are relative, at the edges of a domain (or sub-domain) these are relative to the surrounding void domain, and hence can be

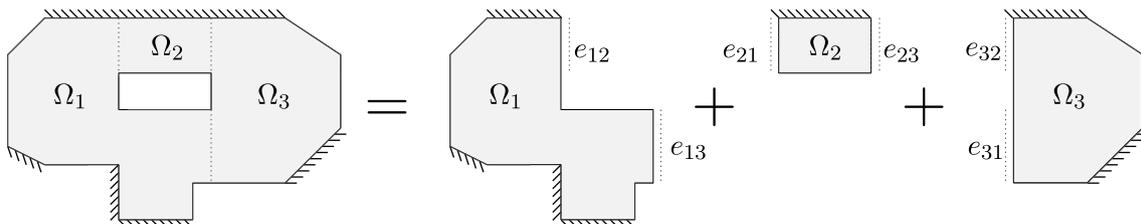


Figure D.1: Domain decomposition for a slab with hole, leading to three sub-domains Ω_1 , Ω_2 , and Ω_3 connected by six coupling boundaries (i.e., e_{12} , e_{21} , etc.)

considered to be absolute. Thus, assuming the absolute displacements at the edges of sub-domain Ω_1 and Ω_2 are denoted $\theta_n^{\Omega_1}$, $\theta_t^{\Omega_1}$, and δ^{Ω_1} , and at $\theta_n^{\Omega_2}$, $\theta_t^{\Omega_2}$, and δ^{Ω_2} respectively, the required compatibility condition can be written as follows:

$$\theta_n^{\Omega_1} + \theta_n^{\Omega_2} - \theta_n^B = 0, \quad (\text{D.1a})$$

$$\theta_t^{\Omega_1} + \theta_t^{\Omega_2} = 0, \quad (\text{D.1b})$$

$$\delta^{\Omega_1} + \delta^{\Omega_2} = 0. \quad (\text{D.1c})$$

Where θ_n^B is introduced to model the presence of a potential real yield-line at the boundary (supplemented by corresponding plastic multiplier terms). Note that, in order to avoid sign convention issues, all line directions are assumed to be identical. Thus if \mathbf{S} denotes the coefficient matrix for constraint D.1 then $\mathbf{Sd} = \mathbf{0}$. Hence the compatibility constraints for a problem where domain decomposition has been used can be written as:

$$\mathbf{B}^\alpha \mathbf{d} = \mathbf{0}, \quad \text{for all } \alpha \in \mathbb{S}, \quad (\text{D.2})$$

$$\mathbf{Sd} = \mathbf{0}, \quad (\text{D.3})$$

where \mathbf{B}^α is the compatibility matrix for sub-problem α , and where \mathbb{S} is the set of all sub-domains.

Appendix E

Extra considerations in geometry optimization

In geometry optimization, nodal positions (\mathbf{x} and \mathbf{y}) are considered as optimization variables, in addition to \mathbf{d} and \mathbf{p} in optimization problem (6.1). Therefore, the coefficient matrices and vectors in (6.1) contain the optimization variables, which are continuously updated during the optimization process. For example, whilst \mathbf{g} comprises constants coefficient values when slabs are isotropically reinforced, when orthotropic reinforcement is present the coefficient values are affected by the yield-line angles ϕ (see also Fig. 6.4 and yield-criterion (6.3)) that are determined by \mathbf{x} and \mathbf{y} ; hence \mathbf{g} is now a function of the optimization variables.

In addition, functions representing the load effect terms \mathbf{f}_L and \mathbf{f}_D can become non-smooth with respect to nodal positions, which can cause problems. In this paper extra constraints are added to prevent these functions from becoming non-smooth. Thus in Fig. E.1(a) node A is made non-movable since it coincides with a point load; in Fig. E.1(b), node B is restrained so as to only be able to move in the direction of the line load. For patch loads, domain decomposition (which divides a slab domain into several separate sub-domains; see He & Gilbert (2015b) for details) can be used, and a sub-domain can be created in the patch load area. Note that the above approaches restrict nodal movements and will therefore potentially somewhat reduce the accuracy of the numerical solutions obtainable using geometry optimization.

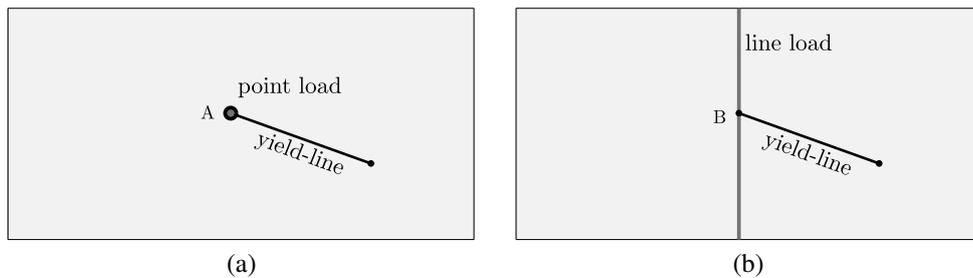


Figure E.1: Extra move limits imposed on nodes due to the presence of point and line loads: (a) node A is non-movable because it coincides with a point load; (b) node B can only move in the direction of the the line load

Appendix F

Geometric data of the real-world slab

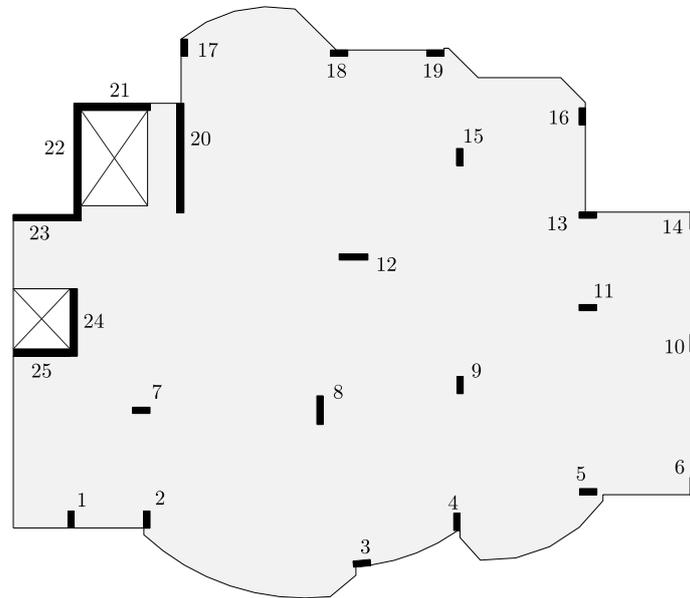


Figure F.1: Real-world slab: geometry and column ID

Table F.1: Real-world slab: vertices of the polygonal slab domain (units in metres)

ID	x	y												
1	-0.727	0.000	11	10.250	-2.400	21	14.748	-0.335	31	19.090	14.855	41	5.956	17.661
2	3.794	0.000	12	11.143	-1.650	22	15.456	-1.123	32	18.237	15.723	42	5.085	17.069
3	3.794	-0.238	13	11.143	-1.427	23	16.686	-1.046	33	15.376	15.723	43	5.085	14.826
4	4.466	-0.802	14	11.143	-1.359	24	17.855	-0.655	34	14.348	16.751	44	1.373	14.826
5	5.194	-1.291	15	11.657	-1.296	25	18.883	0.025	35	14.198	16.751	45	1.373	10.941
6	5.971	-1.700	16	12.460	-1.132	26	19.702	0.946	36	14.198	16.686	46	-0.727	10.941
7	6.786	-2.022	17	13.238	-0.874	27	19.702	1.161	37	10.463	16.686	47	-0.727	8.350
8	7.632	-2.256	18	13.980	-0.525	28	22.934	1.161	38	9.029	18.118	48	1.488	8.350
9	8.503	-2.398	19	14.675	-0.090	29	22.934	11.027	39	7.979	18.199	49	1.488	6.000
10	9.384	-2.446	20	14.748	-0.090	30	19.090	11.027	40	6.937	18.044	50	-0.727	6.000

Table F.2: Real-world slab: locations of holes (units in metres)

Description	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4
Stair	1.623	11.250	3.925	11.250	3.925	14.576	1.623	14.576
Core	-0.727	6.250	1.238	6.250	1.238000	8.350	-0.727	8.350

Table F.3: Real-world slab: information of column supports (units in metres)

Column ID	(Scenario 1) Support strength i	(Scenario 2) Support strength i	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4
1	1	0	1.165	0.000	1.380	0.000	1.380	0.592	1.165	0.592
2	1	0	3.794	0.592	3.794	-0.008	4.009	-0.008	4.009	0.592
3	1	0	11.040	-1.157	11.060	-1.367	11.658	-1.316	11.638	-1.099
4	1	0	14.533	-0.090	14.748	-0.090	14.748	0.510	14.533	0.510
5	1	0	18.891	1.161	19.491	1.161	19.491	1.376	18.891	1.376
6	1	0	22.934	1.761	22.719	1.761	22.719	1.161	22.934	1.161
7	1	0	3.410	4.000	4.010	4.000	4.010	4.215	3.410	4.215
8	1	0	9.788	3.613	10.003	3.613	10.003	4.613	9.788	4.613
9	1	0	14.644	4.690	14.859	4.690	14.859	5.290	14.644	5.290
10	1	0	22.719	6.148	22.934	6.148	22.934	6.748	22.719	6.748
11	1	0	18.875	7.586	19.475	7.586	19.475	7.801	18.875	7.801
12	1	0	10.560	9.354	11.560	9.354	11.560	9.569	10.560	9.569
13	1	0	18.875	10.812	19.475	10.812	19.475	11.027	18.875	11.027
14	1	0	22.719	10.427	22.934	10.427	22.934	11.027	22.719	11.027
15	1	0	14.627	12.650	14.842	12.650	14.842	13.250	14.627	13.250
16	1	0	18.875	14.069	19.090	14.069	19.090	14.669	18.875	14.669
17	1	0	5.085	16.469	5.300	16.469	5.300	17.069	5.085	17.069
18	1	0	10.260	16.471	10.860	16.471	10.860	16.686	10.260	16.686
19	1	0	13.598	16.471	14.198	16.471	14.198	16.686	13.598	16.686
20	0	0	4.935	11.000	5.185	11.000	5.185	14.826	4.935	14.826
21	0	0	1.373	14.576	4.025	14.576	4.025	14.826	1.373	14.826
22	0	0	1.623	10.726	1.623	14.576	1.373	14.576	1.373	10.726
23	0	0	1.373	10.941	-0.727	10.941	-0.727	10.726	1.373	10.726
24	1	1	1.238	6.000	1.488	6.000	1.488	8.350	1.238	8.350
25	1	1	1.238	6.250	-0.727	6.250	-0.727	6.000	1.238	6.000

Appendix G

Sign convention used in DLO (for the yield-line analysis of slabs)

G.1 Relative displacements

In DLO, kinematic variables represent the *relative* displacements between a pair of rigid blocks in contact. For example in Fig. G.1, assume two rigid blocks ABCD (block I) and ABEF (block II) are in contact at line AB. If this line has $\theta_n > 0$, then it describes sagging behaviour in a slab yield-line analysis problem, regardless of the predefined numbering order of this line (i.e., AB or BA).

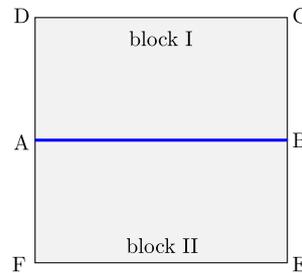


Figure G.1: Two blocks contacting at yield-line AB

To describe the relative displacements θ_n , θ_t and δ , both blocks need to be taken into account. To describe their displacements separately, now ‘split’ line AB to $A'B'$ and $A''B''$, belonging to blocks I and II respectively. As illustrated in Fig. G.2, $A'B'$ and $A''B''$ need to rotate in opposite directions to define relative rotations, and so does the shear displacement. The relative displacements (i.e., yield-line displacements) can now be written as:

$$\theta_n = \theta'_n + \theta''_n, \quad (\text{G.1})$$

$$\theta_t = \theta'_t + \theta''_t, \quad (\text{G.2})$$

$$\delta = \delta' + \delta''. \quad (\text{G.3})$$

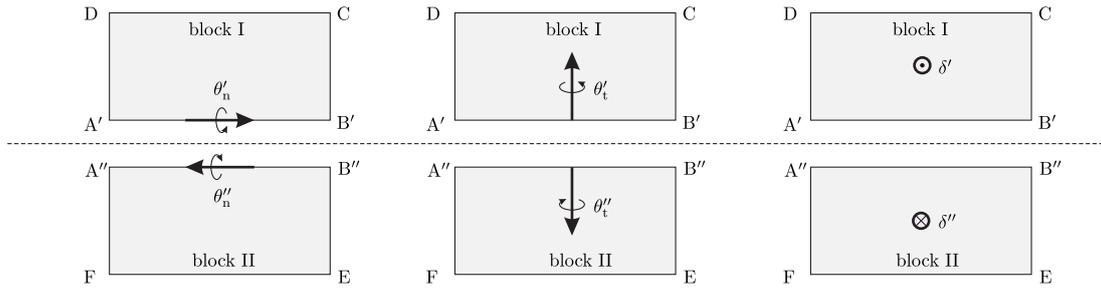


Figure G.2: Relative displacements between two contacting blocks

G.2 Local coordinate system

From a computational perspective, including both lines appears redundant - by assuming one block is stationary (e.g., let $\theta''_n = \theta''_t = \delta'' = 0$), then the displacement of the other correctly reflects the relative displacements. For this reason, DLO uses one line - it assumes line A'B' if it reads AB in Fig. G.1, or A''B'' if it reads BA - and establishes a local coordinate system on it to calculate parameters such as live load effects. If A'B' is used, it assumes block II is stationary and that block I moves in its local coordinate system $\vec{\theta}'_n, \vec{\theta}'_t, \vec{\delta}'$, as shown in Fig. G.3(a); if A''B'' is used, the local coordinate system $\vec{\theta}''_n, \vec{\theta}''_t, \vec{\delta}''$ in Fig. G.3(b) is used.

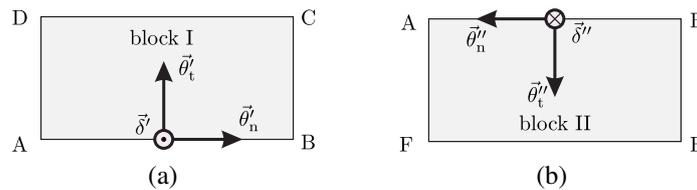


Figure G.3: Local coordinate systems established on the two contacting blocks

Because displacement variables in these two local coordinate systems describe the same relative movement behaviour (i.e., if one represents sagging, so does the other), *calculations involved in using the local coordinate system ($\vec{\theta}'_n, \vec{\theta}'_t, \vec{\delta}'$ or $\vec{\theta}''_n, \vec{\theta}''_t, \vec{\delta}''$) must obtain identical values for both*. For example, when calculating load effects using either of the two local coordinate systems, the calculated load effects (i.e., normal rotational and torsional moments, shear force) must be identical to those calculated using the other coordinate system. A detailed example is given in Fig. G.4. A point load P is acting downwards on block A'B'CD.

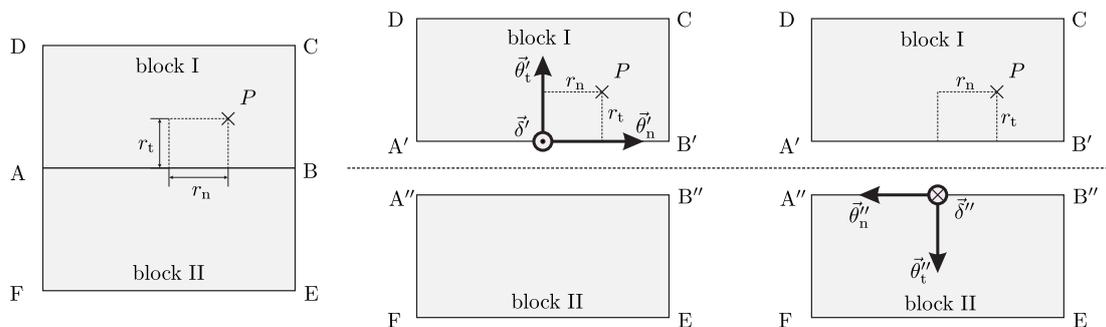


Figure G.4: Evaluating the load effect of a point load P on a yield-line that reads AB or BA

Assuming local coordinate system $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$, the load effect can readily be calculated as $[-r_t P, r_n P, -P]^T$. However, if taking $\vec{\theta}''_n \vec{\theta}''_t \vec{\delta}''$, it is incorrect to calculate the load effects directly in $\vec{\theta}''_n \vec{\theta}''_t \vec{\delta}''$, since P is not directly acting on block A''B''EF. In this situation, an equivalent load $-P$, instead of P , needs to be used, so that its load effect has an identical value of $-r_t P$, $r_n P$, and $-P$ in $\vec{\theta}''_n$, $\vec{\theta}''_t$, and $\vec{\delta}''$ directions respectively.

In addition, it is important to note that the local coordinate system $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ in Fig. G.3(a) obeys the right-hand rule, while $\vec{\theta}''_n \vec{\theta}''_t \vec{\delta}''$ in Fig. G.3(b) obeys the left-hand rule. Normally, right- and left-hand rules cannot coexist in a numerical procedure; for this reason, DLO uses the former but not the latter: by assuming that direction $\vec{\delta}'$ is consistent, $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ is established in block II, as shown in Fig. G.5.

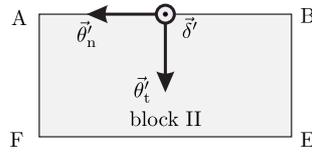


Figure G.5: Alternative local coordinate system for yield line that reads BA

In this case, rotational displacements described in $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ correctly reflect the relative displacements θ_n , and θ_t . However, out-of-plane translational displacement in $\vec{\delta}'$ needs to be reversed to reflect $\vec{\delta}''$. For example, the load effect of P is now given as $[-r_t P, r_n P, P]^T$. Nevertheless, a single local coordinate system has been used, and several points are given in the following:

- To represent relative displacements, $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ is always established on the moving block, meaning, the other contacting block is assumed to be still.
- $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ obeys the right-hand rule: directions of $\vec{\theta}'_n$ are determined by the numbering order of a yield-line (i.e., AB or BA); while the direction of $\vec{\delta}'$ is consistent, that of $\vec{\theta}'_t$ is obtained using the right-hand rule $\vec{\theta}'_t = \vec{\delta}' \times \vec{\theta}'_n$.
- Direction $\vec{\theta}'_t$ always points towards the moving block.
- If a load is applied on the stationary block, it can be transferred to $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ by reversing its direction. In addition, the sign of the calculated shear effect in $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ needs to be reversed.

G.3 Yield-lines on slab boundary

For boundary yield-lines, it is significant to identify the moving block based on the prescribed numbered order of yield-lines. Clearly, if $\vec{\theta}'_n \vec{\theta}'_t \vec{\delta}'$ is established on the slab but not the external supports, displacement variables directly reflect boundary movements. Because $\vec{\theta}'_t$ points towards the moving block, it then requires that yield-lines on boundaries are given in an anti-clockwise order (e.g., Fig. G.6(a)). Conversely, if they are given in a clockwise order (e.g., Fig. G.6(b)), the slab is the still block when considering relative movement, and the displacement variables reflect movements of the external supports. In this case, boundary displacements can be obtained by reversing the signs of the yield-line displacement variables.

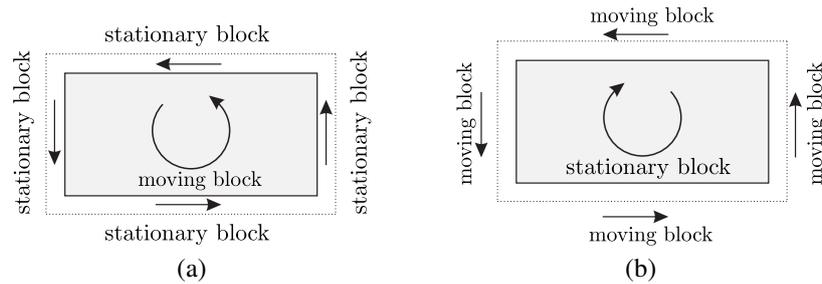


Figure G.6: Numbering order of boundary yield-lines, solid arrows indicate directions of yield-lines: (a) anti-clockwise; (b) clockwise

G.4 Conclusion

- Displacement variables in DLO reflect the relative displacements between two blocks.
- By assuming one block is moving and the other is stationary, the displacement of the moving block correctly represents the relative movement.
- A local coordinate system is established on the moving block.
- For a given yield-line, depending on which moving block is selected, two equivalent local coordinate systems exist: one obeys the right-hand rule, and the other the left-hand rule.
- In DLO, the local coordinate system that obeys the right-hand rule is used.
- If load is not applied on the moving block, it must be converted to the local coordinate system.
- If yield-lines on slab boundaries are given in anti-clockwise order, their displacement variables directly reflect boundary movements.