

# Structural Optimization

William R. Spillers · Keith M. MacBain

# Structural Optimization

 Springer

William R. Spillers  
Department of Civil and Environmental  
Engineering  
New Jersey Institute of Technology  
University Heights  
Newark, NJ 07102  
USA  
spillers@njit.edu

Keith M. MacBain  
Geiger Engineers  
P.C. NY Office  
2 Executive Blvd.  
Suffern, NY 10901  
Suite 410  
USA  
k\_macbain@yahoo.com

ISBN 978-0-387-95864-4 e-ISBN 978-0-387-95865-1  
DOI 10.1007/978-0-387-95865-1  
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009920688

© Springer Science+Business Media, LLC 2009

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

*for our wives*

# Preface

Contemporary structural optimization has its roots in the 1960s with Lucien Schmidt's seminal paper.<sup>1</sup> Prior to that time there were no texts on nonlinear programming and if you wanted to do optimization you were relegated to using linear programming. Once mathematical programming was discovered by designers it was thought that engineering design, as an area of study, was over since all you had to do was formulate your design as a nonlinear programming problem and invoke some canned solver. That turned out not to be the case.

While the 1960s and 1970s were characterized by difficulties in solving even small optimization problems (forgetting for the moment optimality criteria methods), the 1990s were characterized by discussions of mathematical programming methods for solving large systems. The flavor of these discussions can be found in the fact that workers in linear programming were then solving *large* systems (Bixby et al., 1991) with, for example, 12 million variables. In fact, today (2007) the web site of Jacek Gondzio describes solving a nonlinear programming problem with 353 million rows and 1010 million columns.

These capabilities of mathematical programming solvers are part of the driving force behind this text: Surely this technology should offer hope to the structural engineer who must commonly deal with large systems. The other factor driving this work is context: We believe that the use of sequential linear programming together with the use of the incremental equations of structures can serve as a focal point about which a quite general structural optimization solver can be developed.

Then there is the question of the availability of software today. In view of the Solver package included in Microsoft EXCEL, an argument can be made that optimization software is now available to everyone. That package is used in this text along with the IMSL routines available with Digital FORTRAN (now sold as Intel FORTRAN). And clearly, all the work described in this text could have as well been done using Matlab. The reader will find a mix of this software used here. The point is that tools are now available for the solution of optimization problems. This includes some freeware available on the Internet that will be discussed later.

With regard to content, this book includes many computer programs most of which are written in FORTRAN. The excuse for focusing so heavily on computing is that contemporary structural optimization is about computing. Parenthetically, we try in this book to do justice to the history of structural optimization but surely have left things out and apologize to those authors whose work has not been properly referenced here.

---

<sup>1</sup> The history of structural optimization has been developed carefully by Wasiutynski and Brandt (1963).

The attempt in writing this book is to help bring the methods of structural optimization into common usage like those of the finite element method. That is, when the structural engineer sits down to design something, he/she should not only have analysis tools available but should also have optimization tools available. In fact, a good case can be made for including these tools in one package since analysis can always be performed as an optimization problem. The engineer could then, for example, automatically change the structural shape as he/she attempts to satisfy some allowable stress constraints. The analysis/redesign cycle could then simply be replaced by optimization steps.

Historically, there has been a tension between proponents of classical optimization methods who claimed that the users of optimality criteria methods were lacking in theory and the users of heuristic schemes such as optimality criteria methods who at the same time claimed that the classical methods were incapable of solving real (large) structures. In view of the tools now available to the engineer, these arguments can be seen to diminish in importance although the optimality criteria methods still have enormous physical appeal.

We attempt to deal fairly with optimality criteria methods but clearly the focus is on the use of sequential linear programming and the incremental equations of structures that comprise a classical approach. The text begins with an Introduction to simple problems of optimization. Then some available tools are discussed. (This second chapter is more formal than the remainder of the text and the first-time reader might treat it lightly.) Chapter 3 introduces our central topic which is structural optimization approached via the incremental equations of structures and sequential linear programming. Chapter 4 then discusses some problems solved using optimality criteria methods. The remainder of the text offers what we see as an overview of the field of structural optimization. This includes beams and plates, dynamic systems, multicriteria methods, and a brief discussion of some ongoing work. The text closes with an Appendix containing three reprints that are regarded by the authors to be basic to the historical development of structural optimization.

Finally, it has been a pleasure to work Springer Science+Business Media and Steven Elliot our editor. For us, Springer has managed to bring the capabilities and competence of an enormous organization to bear without losing the personal touch of staff contacts. You could not ask for more than that.

Keith M. MacBain  
Suffern, NY  
William Spillers  
Newark, NJ

# Contents

<b>1. Introduction .....</b>	<b>1</b>
1.1 Problem Statement.....	1
1.2 An Optimization Problem.....	2
1.3 Elementary Calculus.....	5
1.4 Optimal Slope for Truss Bars .....	6
1.5 An Arch Problem.....	7
1.6 The Gradient of a Function.....	8
1.7 The Lagrange Multiplier Rule .....	11
1.8 Newton's Method .....	12
1.9 Solving Linear Equations.....	14
1.10 Linear Systems Versus Optimization.....	14
1.11 Equations of Structures.....	15
1.12 Plastic Analysis.....	18
1.13 A Beam Problem.....	19
1.14 Quadratic Programming.....	20
1.15 Embedding.....	21
1.16 Geometric Programming.....	21
1.17 Plastic Design of Plane Frames .....	22
1.18 Problems .....	24
<b>2. Some Tools of Optimization .....</b>	<b>29</b>
2.1 The Lagrange Multiplier Rule .....	29
2.2 The Kuhn-Tucker Conditions .....	31
2.3 Calculus of Variations .....	34
2.4 Newton's Method .....	36
2.4.1 An Example of Newton's Method.....	37
2.5 Linear Programming.....	40
2.5.1 An Example of the Simplex Method.....	41
2.5.2 Interior Point Methods .....	43
2.6 Sequential Linear Programming .....	44
2.7 Other Methods of Mathematical Programming .....	46
2.8 Genetic Algorithms.....	47
2.9 Problems .....	48
<b>3. Sequential Linear Programming and the Incremental Equations of Structures.....</b>	<b>49</b>
3.1 Introduction .....	49
3.2 The Incremental Equations of Structures.....	49
3.3 Application to Structural Optimization.....	51
3.4 An Example with a Displacement Constraint.....	52
3.5 Adding Stress Constraints.....	56
3.6 The 25-Bar Truss .....	57

3.7 A Frame Example .....	60
3.8 A Buckling Example.....	63
3.9 The Incremental Equations when Shape Change is Allowed.....	67
3.10 A Beam Example .....	72
3.11 A Plate Bending Problem.....	74
3.12 Problems .....	76
<b>4. Optimality Criteria Methods.....</b>	<b>77</b>
4.1 Introduction.....	77
4.2 The Most Simple Optimality Criteria Problem. ....	78
4.3 Monotone Behavior .....	79
4.4 An Application.....	80
4.5 Sandwich Beams.....	84
4.6 A Generalization of the Truss problem.....	84
4.7 Plastic Design of Frames .....	86
4.8 Sandwich Plate Design. ....	88
4.9 Truss Design .....	89
4.10 A Plane Stress Problem.....	91
4.11 Prager and His Co-workers .....	95
4.11.1 The Axisymmetric Circular Plate .....	95
4.12 A Plate Problem .....	97
4.13 Problems .....	100
<b>5. Some Basic Optimization Problems .....</b>	<b>103</b>
5.1 Multiple Loading Conditions .....	103
5.1.1 An Example.....	105
5.1.2 Realizability .....	106
5.1.3 Three Loading Conditions.....	108
5.2 Deflection Constraints .....	109
5.2.1 Funaro's Example.....	112
5.2.2 Optimality Criteria Approach.....	117
5.2.3 An Example.....	121
5.3 Optimal Shape.....	122
5.3.1 Frame Problems.....	124
5.3.2 Code Design for a Truss Problem.....	126
5.4 Generating New Designs Automatically.....	128
5.4.1 Algebraic Methods .....	130
5.4.2 Linguistic Methods.....	135
5.5 Problems .....	137
<b>6. Beams and Plates: The Work of Rozvany .....</b>	<b>139</b>
6.1 Introduction.....	139
6.1.1 Examples .....	140
6.1.2 Reaction Costs.....	143
6.1.3 Optimal Segmentation.....	144

6.1.4 A More Realistic Beam Model.....	145
6.2 Design of Plates .....	146
6.3 Problems .....	149
<b>7. Some Problems of Dynamic Structural Optimization.....</b>	<b>151</b>
7.1 Introduction .....	151
7.2 Optimization for Transient Vibrations.....	152
7.2.1 The Static Case.....	154
7.2.2 The Dynamic Case .....	154
7.2.3 The Work of Connor .....	160
7.2.3.1 Tuned Mass Dampers .....	161
7.3 Steady-State Problems .....	162
7.3.1 A Truss Problem.....	165
7.3.2 An Algorithm for the Truss Problem.....	166
7.3.3 An Incremental Solution of the Truss Problem .....	171
<b>8. Multicriteria Optimization .....</b>	<b>175</b>
8.1 Introduction .....	175
8.2 Solving Multicriteria Optimization Problems.....	177
<b>9. Practical Matters: The Work of Farkas and Jarmai.....</b>	<b>179</b>
9.1 Introduction .....	179
9.2 Sizing Member Cross Sections .....	179
9.2.1 Effect of Fabrication Costs.....	181
9.3 Tubular Trusses .....	182
9.3.1 The Effect of Shape.....	184
9.4 Problems .....	186
<b>10. On Going Work .....</b>	<b>187</b>
10.1 Design of Tall Buildings.....	187
10.1.1 Wind Loads on Tall Buildings.....	187
10.1.2 Tuned Mass Dampers .....	188
10.1.3 Stochastic Processes .....	189
10.2 Heuristic Algorithms .....	192
10.3 Extending the Design Process.....	193
10.4 Design Theory .....	193
10.4.1 Robust Design .....	194
10.4.2 Creativity in Design .....	194
10.4.3 Design Ontologies .....	195
10.4.4 Architectural Design .....	195
10.5 Available Computational Algorithms.....	195
<b>A. Using the Computer .....</b>	<b>197</b>
A.1 Using Computer Languages and Programs.....	197
A1.1 Fortran .....	197

A1.2 Perl .....	199
A1.3 Makefiles .....	199
A.2 Matlab .....	199
A.3 Microsoft Excel.....	200
A.3.1 The Solver Routine.....	200
A.3.2 Visual Basic.....	203
A.5 Freeware.....	205
A.6 Graphical Interface Applications .....	205
<b>B. The Node Method for Trusses .....</b>	<b>207</b>
B.1 Introduction .....	207
B.2 A Formal Description of the Truss Problem .....	208
B.3 A Decomposition .....	212
<b>C. Convex Sets and Functions: Homogeneous Functions .....</b>	<b>219</b>
C.1 Convex Sets and Functions.....	219
C.2 Homogeneous Functions.....	220
<b>D. Structural Optimization Classics .....</b>	<b>223</b>
D.1 Michell Trusses.....	223
D.2 Keller’s Optimal Column.....	233
D.3 The Paper of Venkayya, Khot, and Berke.....	245
<b>References .....</b>	<b>289</b>
<b>Index .....</b>	<b>299</b>

# Contents of the CD

## Readme

The readme.txt file contains some basic information as to how to use the CD. More detailed information is given in the text and in the Tools folder.

## Computer Programs

This folder contains source code for all programs discussed in the text. A list of these programs is given below.

### List of Programs

- Program 1. (Prog01.xls) This Excel worksheet is used in Chapter 1 to solve the two-bar truss problem.
- Program 2. (Prog02.xls) This Excel worksheet is used in Chapter 1 to solve the frame design.
- Program 3. (Prog03.for) This FORTRAN program is used in Section 2.6 of the text as an example of the sequential linear programming solver.
- Program 4. (Prog04.xls) This Excel worksheet uses Solver check of the problem solved by Program 3.
- Program 5. (Prog05.for) This FORTRAN program is used in Appendix C as a sequential linear programming solver.
- Program 6. (Prog06.xls) This Excel worksheet is the Solver solution used in Appendix C.
- Program 7. (Prog07.m) This Word file contains a Matlab script file and a Matlab function. They must be separated before they are used. This program is a Matlab version of sequential linear programming.
- Program 8. (Prog08.xls) This Excel worksheet is a study of the use of visual basic for structural optimization.
- Program 9. (Prog09.for) This FORTRAN program solves a truss with a displacement constraint from Section 3.4 in the text. It is to be used with the data file Prog09.dat.
- Program 10. (Prog10.for) This is Program 9 with stress constraints added. It is to be used with the data file Prog10.dat.
- Program 11. (Prog11.for) This FORTRAN program solves the classic 25-bar truss problem. It is to be used with the data file Prog11.dat.
- Program 12. (Prog12.for) This FORTRAN program solves a plane frame problem with a displacement constraint as discussed in Section 3.7. It is to be used with the data file Prog12.dat.

- Program 13. (Prog13.for) This FORTRAN program solves Keller's problem of the optimal shape for a column.
- Program 14. (Prog14.for) This FORTRAN program solves the geometric optimization problem discussed in Section 3.9 in the text. It is to be used with the data file Prog14.dat.
- Program 15. (Prog15.for and Prog15.dat) This is another version of Program 14.
- Program 16. (Prog16.m) This Matlab program solves the problem of an optimal fixed beam as discussed in Section 3.10.
- Program 17. (Prog17.m) This Matlab program solves the problem of an optimal fixed plate as discussed in Section 3.11.
- Program 18. (Prog18.for) This FORTRAN program solves the most simple iterative design problem for trusses discussed in Section 4.4 of the text. It uses Prog18.dat as a data file.
- Program 19. (Prog19.for) This FORTRAN program solves the iterative design problem for plane frames discussed in Section 4.7 of the text. It uses Prog19.dat as a data set.
- Program 20. Program 20 comprises three programs: Prog20.for, a basic finite element program; Prog21.for, a mesh generator; and Prog22.for, an iterative design program for plane stress problems. The use of these programs is discussed in Section 4.10 in the text.
- Program 23. (Prog23.for) This FORTRAN program solves the iterative design problem for a fixed plate as discussed in Section 4.12 of the text.
- Program 24. (Prog24.m and Prog25.m) These Matlab programs comprise a program for the solution of the transient response of a tapered beam as discussed in Section 7.2 of the text.
- Program 26. (Prog26.for) This FORTRAN program solves the problem of the optimal steady-state vibrations of a truss as discussed in Section 7.3 of the text. It is run with Prog26.dat as a data file.
- Program 27. (Prog27.for) This is an incremental solution of the problem solved in Program 26. It uses the same data file.
- Program 28. (Prog28.for) This FORTRAN program resolves an example of Farkas from Section 9.3 using AISC stresses. It is to be run with the data file Prog28.dat.
- Program 29. (Prog29.for) This FORTRAN program solves the three-dimensional truss. It is used in Problem 3 of Chapter 1. It is to be run with the data file Prog29.dat.
- Program 30. (Prog30.m) This Matlab program executes the simplex method. It is an educational program not intended for serious applications.
- Program 31. (Prog31.for) This FORTRAN program executes the primal interior point linear programming algorithm described by Arbel. It is to be run with Prog31.dat as a data file.

- Program 32. (Prog32.for) This FORTRAN program executes the dual interior point linear programming algorithm described by Arbel. It is to be run with Prog32.dat as a data file.
- Program 33. (Prog33.xls) This Excel worksheet is used in Section 9.3 to compute the properties of hollow sections.
- Program 34. (Prog34.xls) This Excel worksheet is used in Section 9.3 to compute the properties of angles.

## Tools

This folder contains some tools that are considered useful. A list of these tools is given below.

### List of Tools

tr2d.	This folder contains an example of using VB code in Excel to automate tasks and using the solver.
g77.zip	A freely available FORTRAN compiler.
MinGW	A freely available FORTRAN compiler. This is the compiler that was tested with the files in optools.zip.
optools	Files that contain IMSL routines used in the text for users without this library.
unixUtils	Unix-like utilities for a PC.
unzip	Executable file to unzip files on the disk.
View3d	Java-based graphical user interface (GUI) framework.