Second-order structural optimization

A. F. M. Azevedo
Faculty of Engineering, University of Porto, 4099 Porto Codex, Portugal

Abstract

This paper describes a second-order method that can be used to calculate the optimal solution of a nonlinear program with equality and inequality constraints. The functions that define the mathematical program are generalized polynomials, allowing for the interpretation and derivation of the objective function and constraints, in a fully automatic, exact and efficient manner. Slack variables are used to convert the inequality constraints into equality constraints. The optimal solution is calculated with the Lagrange-Newton method. Results are presented for the cost minimization of linear 3D trusses, where successful solutions have been achieved for problems with over 10^3 independent design variables and over 10^4 constraints.

1 Introduction

First-order methods are the most commonly used in structural optimization [4] [10] and are often combined with sensitivity analysis [7]. These algorithms are usually employed to solve problems where the displacement method is used and the behavior of the structure is considered to be linear. However, in many problems this formulation is inaccurate or not applicable and the advantages of the traditional approach vanish. General purpose optimization methods could be used in these circumstances, but their inability to deal with large scale problems and some difficulties associated with the calculation of derivatives make this choice less attractive.

The use of second-order information has been referred by many authors as a mean of increasing both the convergence rate and the accuracy of the optimization methods [1] [11]. However, the calculation and storage of second derivatives are considered too demanding and are usually avoided. Some

alternative techniques try to collect second-order information without explicitly calculating second derivatives [6]. The algorithm described in this paper is not a general purpose one, but is applicable to a wide variety of structural optimization problems, with the advantages of being a true second-order method.

2 Nonlinear programming

The minimization of a function subject to equality and inequality constraints has the following general formulation

$$\textit{Minimize} \quad f\left(\underset{\sim}{x}\right) \qquad \underset{\sim}{x} = \left(x_1,\ldots,x_n\right) \tag{1}$$

subject to

$$\underset{\sim}{g}\left(\underset{\sim}{x}\right) \le \underset{\sim}{0} \qquad \underset{\sim}{g} = \left(g_1,\ldots,g_m\right) \tag{2}$$

$$\underset{\sim}{h}\left(\underset{\sim}{x}\right) = \underset{\sim}{0} \qquad \underset{\sim}{h} = \left(h_1,\ldots,h_p\right) \tag{3}$$

In the algorithm presented in this paper, all the variables can assume positive or negative real values and the functions f, g_i and h_i have the following form

$$\sum_{t=1}^{T} c_t \prod_{i=1}^{n} x_i^{a_{ti}} \tag{4}$$

Expression (4) describes a generalized polynomial that is exemplified by the following function

$$f\left(\underset{\sim}{x}\right) = 5.9x_1^2x_4^{-3} - 3.1x_2 + 2.7x_1^{-1}x_3x_5^2 - 1.8 \tag{5}$$

These kinds of expressions are stored in vectors where only the term coefficients, the variable indexes and the exponents are present. The expressions of the first and second derivatives can be easily obtained and evaluated at the current solution. These operations are performed by the computer in a very accurate and efficient manner, avoiding problems associated with the symbolic derivation of complex expressions or the selection of a suitable finite difference step [8].

Squared slack variables are added to the inequality constraints, allowing for a generic treatment of all the constraints as equalities [1].

$$g_i\left(\underset{\sim}{x}\right) \le 0 \quad \rightarrow \quad g_i\left(\underset{\sim}{x}\right) + s_i^2 = 0 \qquad \left(i = 1,\ldots,m\right) \tag{6}$$

This operation adds *m* variables to the nonlinear program. However, as will be shown later, this inconvenience is not significant.

3 Lagrange-Newton method

The Lagrangian of the nonlinear program described in the previous section is given by [7]

$$L\left(\underset{\sim}{X}\right) = f\left(\underset{\sim}{x}\right) + \sum_{k=1}^{m} \lambda_k^g \left[g_k\left(\underset{\sim}{x}\right) + s_k^2 \right] + \sum_{k=1}^{p} \lambda_k^h h_k\left(\underset{\sim}{x}\right) \tag{7}$$

In equation (7), $\underset{\sim}{\lambda}^g$ and $\underset{\sim}{\lambda}^h$ are the Lagrange multipliers associated with the inequality and equality constraints respectively. The total number of variables is $2m+n+p$. Vector $\underset{\sim}{X}$ contains all these variables ordered in the following way

$$\underset{\sim}{X} = \left(\underset{\sim}{s}, \underset{\sim}{\lambda}^g, \underset{\sim}{x}, \underset{\sim}{\lambda}^h \right) \tag{8}$$

The solution of the nonlinear program (1)-(3) is a saddle point of the Lagrangian (7) and the following necessary conditions can be used in its calculation

$$\nabla L\left(\underset{\sim}{X}\right) = \underset{\sim}{0} \Rightarrow \left| \begin{array}{ll} 2 s_i \lambda_i^g = 0 & \left(i = 1,\ldots,m\right) \quad (9) \\ g_i + s_i^2 = 0 & \left(i = 1,\ldots,m\right) \quad (10) \\ \dfrac{\partial f}{\partial x_i} + \sum_{k=1}^{m} \lambda_k^g \dfrac{\partial g_k}{\partial x_i} + \sum_{k=1}^{p} \lambda_k^h \dfrac{\partial h_k}{\partial x_i} = 0 & \left(i = 1,\ldots,n\right) \quad (11) \\ h_i = 0 & \left(i = 1,\ldots,p\right) \quad (12) \end{array} \right.$$

The Newton method can be used to calculate the solution of the system of $2m+n+p$ nonlinear equations (9)-(12). When this technique is employed, the optimization method is termed Lagrange-Newton [5]. For each Newton iteration (q) the following system of linear equations must be solved

$$\underset{\sim}{H}\left(\underset{\sim}{X}^{q-1}\right) \Delta \underset{\sim}{X}^{q} + \nabla L\left(\underset{\sim}{X}^{q-1}\right) = \underset{\sim}{0} \tag{13}$$

Matrix $\underset{\sim}{H}$ is the Hessian of the Lagrangian and its terms can be obtained by derivation of equations (9)-(12) in order to the variables indicated in (8). Matrix $\underset{\sim}{H}$ is symmetric and is composed of 16 submatrices, whose expressions are

	(m)	(m)	(n)	(p)
(m)	$\underset{\sim}{H}_{11}$	$\underset{\sim}{H}_{12}$	$\underset{\sim}{0}$	$\underset{\sim}{0}$
(m)		$\underset{\sim}{0}$	$\underset{\sim}{H}_{23}$	$\underset{\sim}{0}$
(n)			$\underset{\sim}{H}_{33}$	$\underset{\sim}{H}_{34}$
(p)	SYM.			$\underset{\sim}{0}$

(14)

$$\underset{\sim}{H}_{11} \rightarrow H_{ij} = 2\lambda_i^g, \; if \;\; i = j; \;\; H_{ij} = 0, \; if \;\; i \neq j \tag{15}$$

$$\underset{\sim}{H}_{12} \rightarrow H_{ij} = 2 s_i, \; if \;\; i = j; \;\; H_{ij} = 0, \; if \;\; i \neq j \tag{16}$$

$$\underset{\sim}{H}_{23} \rightarrow H_{ij} = \frac{\partial g_i}{\partial x_j} \tag{17}$$

$$\underset{\sim}{H}_{33} \rightarrow H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} + \sum_{k=1}^{m} \lambda_k^g \frac{\partial^2 g_k}{\partial x_i \partial x_j} + \sum_{k=1}^{p} \lambda_k^h \frac{\partial^2 h_k}{\partial x_i \partial x_j} \tag{18}$$

$$\underset{\sim}{H}_{34} \rightarrow H_{ij} = \frac{\partial h_j}{\partial x_i} \tag{19}$$

In most problems, the number of lines and columns of the Hessian matrix is very large ($2m+n+p$). Fortunately, the number of nonzero terms is usually small and the system of linear equations (13) can be solved using an algorithm that takes into account the sparsity of the matrix.

When Gaussian elimination is used, the algorithm spends almost all the time reducing the $(n+p)\times(n+p)$ symmetric submatrix. Submatrices $\underset{\sim}{H}_{21}$ and $\underset{\sim}{H}_{32}$ can be efficiently eliminated looping over their nonzero terms. Storage requirements are only significant for the $(n+p)\times(n+p)$ symmetric submatrix, because the other submatrices require only the storage of their nonzero terms. For these reasons, the number of inequality constraints (m) does not affect significantly the computation time and the storage requirements.

The solution of the system of linear equations (13) can also be calculated using a conjugate gradient algorithm [9]. Both members of equation (13) are multiplied by $\underset{\sim}{H}^T$, because $\underset{\sim}{H}$ is indefinite. The conjugate gradient method is thus applied to a linear system whose coefficient matrix is $\underset{\sim}{H}^T \underset{\sim}{H}$. This procedure increases the condition number significantly and the number of CG iterations becomes unacceptably large. Despite its low storage requirements, the conjugate gradient method is rarely advantageous, due to significantly longer computation times when compared to Gaussian elimination [2].

In each Newton iteration, vector $\Delta \underset{\sim}{X}^q$ is used to update the current solution. In order to increase the reliability of the process and the rate of convergence, $\Delta \underset{\sim}{X}^q$ is multiplied by a scalar value α. A standard line search algorithm is used to calculate the value of α that minimizes the norm of the error $\left(\|\nabla L\|\right)$ in the direction $\Delta \underset{\sim}{X}^q$ [11].

4 Scaling

Newton method based algorithms are much more reliable when the initial solution is close to a minimum and when the problem is not ill-conditioned. These conditions are more easily fulfilled if the original nonlinear program is subjected to suitable transformations. The following variable transformation is

performed by the computer program on each term of the generalized polynomials that define the nonlinear program (4).

$$x_i = Z_i \, \bar{x}_i \qquad (i = 1, \ldots, n) \tag{20}$$

In equation (20), x_i are the original variables of the nonlinear program (1)-(3), $\bar{x}_i$ are the scaled variables and Z_i are fixed scaling parameters. The initial solution values x_i^0 are typically good candidates to be used as scaling parameters. In these circumstances a unit initial solution is an obvious choice for the application of the algorithm to the scaled nonlinear program.

According to transformation (6), the slack variables are treated as decision variables and are equally scaled. A feasible initial solution usually leads to a more reliable optimization process and it is desirable to use in the scaled problem a unit initial solution for all the variables. In these circumstances, the values of the fixed scaling parameters $\left(Z_i\right)$ associated with the slack variables are

$$Z_i = \sqrt{\left| g_i\left(\underset{\sim}{x}^0\right) \right|} \qquad (i = 1, \ldots, m) \tag{21}$$

For numerical reasons, small absolute values of Z_i must be avoided. Equation (20) can be used to calculate the values of the decision variables and slack variables as if no scaling has occurred.

When the Newton method is employed to solve the system of nonlinear equations (9)-(12), the values of the decision variables, slack variables and Lagrange multipliers are simultaneously calculated during the iteration process. For this reason, the values of the Lagrange multipliers should also be as close to unity as possible. This requirement is approximately fulfilled with a constraint normalization, i.e., a multiplication of all the constraints by a factor chosen so that the norm of the gradient of each constraint is unitary at the initial solution. This operation is easily performed because all the functions are generalized polynomials (4) and their derivation, evaluation and multiplication by a factor are trivial. The most suitable initial value for each Lagrange multiplier becomes also the unity. The Lagrange multipliers that correspond to the original nonlinear program can be easily calculated from the scaled values [2].

5 NEWTOP computer program

The algorithm described in the previous sections was coded in ANSI C and called NEWTOP [2]. The nonlinear program data is described using a perceivable syntax, based in keywords, variable names and constraint titles. The computer program parses and validates all the input data. Elementary equality constraints (e.g., $x_i = c$ or $x_i = c\, x_j$) are substituted in the original nonlinear program and occasional simplifications are then automatically

performed. This feature greatly simplifies the task of generating a mathematical program for a new type of problem.

Some graphical output is interactively available during the iteration process. The information that can be displayed consists in the variation of the variables and the evolution of the objective function, error and line search parameter. Deficiencies in the initial solution or in the selection of some parameters can be immediately diagnosed and corrected with no need to restart the iteration process.

The solution of a wide variety of structural optimization problems indicates that the NEWTOP program is reliable, accurate and reasonably efficient in modern computational platforms. In the next section a large scale truss optimization problem is presented.

6 Optimization of a 4096 - bar truss

The characteristics of the program NEWTOP were evaluated by means of a set of parameterized trusses [2]. All these trusses have similar characteristics, except the fact that the number of design variables progressively increases. Only the largest example is presented here. In the structural optimization problem the displacements are treated as decision variables $\underset{\sim}{x}$ (integrated formulation) [3] and there is no variable linking.

The problem consists in the minimization of the volume of a building roof. The structure is a 3D truss subjected to its self weight and to a vertical live load. The truss is simply supported by the building walls with the exception of a large opening. The objective function is the volume of the 4096 truss bars. As the structure has 3135 degrees of freedom, the number of decision variables is increased to 7231 (4096+3135). It was previously referred that the number of inequality constraints does not affect significantly the CPU time and the storage requirements, whereas equality constraints do. In these circumstances the substitution of each equilibrium equation $(h = 0)$ by a pair of inequality constraints $(h < 0; -h < 0)$ is advantageous. The following inequality constraints are also considered: 4096 minimum area constraints, 2×4096 tension/compression stress constraints in the midpoint of each bar and 480 displacement constraints. The total number of inequality constraints is 19038 ($2 \times 3135+3 \times 4096+480$). These constraints are always present in the mathematical program, i.e., no active set strategy is implemented.

A few days of CPU time were required to solve this problem in a desktop workstation with 256 MBytes of RAM and $\approx$ 40 MFlops. The optimal solution can be visualized in Figure 1. Each bar is represented by a cylinder whose base area is proportional to the area of the bar in the optimal solution.

4 096 independent design variables
3 135 degrees of freedom
19 038 inequality constraints

Error (scaled NLP) = 5.5E-4

CONSTRAINTS:
- minimum cross sectional area
- stress (tension/compression)
- maximum nodal displacement

NOTES:
- no variable linking
- no active set strategy

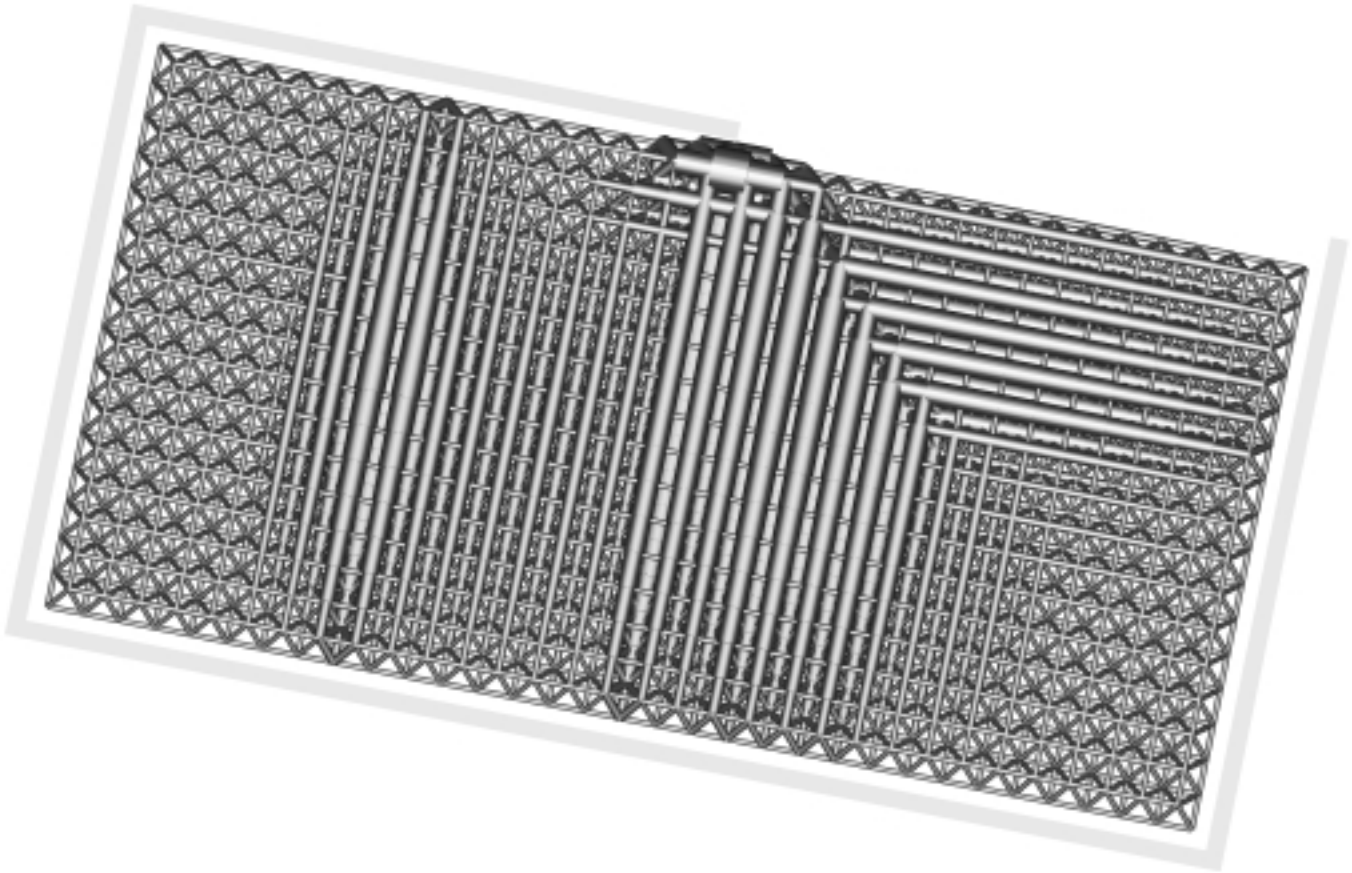

Figure 1: Visualization of the optimal solution of a 4096-bar truss.

7 Conclusions

In this paper an algorithm based in the Lagrange-Newton method was described. Some difficulties usually associated with this method were overcome, allowing for the solution of large scale structural optimization problems. Quadratic convergence is the main advantage of this method, leading to very accurate solutions in a small number of iterations. The inclusion of the displacements in the vector of the decision variables can be problematic when the structure has a large number of degrees of freedom and several load cases. A wide variety of optimization problems can be handled, due to the versatility and user-friendliness of the computer program.

References

1. Arora, J.S. *Introduction to Optimum Design*, McGraw-Hill, 1989.
2. Azevedo, A.F.M. *Structural Optimization with Linear and Nonlinear Behavior*, Ph.D. thesis (in Portuguese), Faculty of Engineering, University of Porto, Portugal, 1994.
3. Barthelemy, B., Haftka, R.T., Madapur, U. & Sankaranarayanan, S. Integrated Structural Analysis and Design Using 3D Finite Elements, *Proceedings of the 30th Structures, Structural Dynamics and Materials Conference*, pp. 1305-1310, Mobile, Alabama, 1989.
4. Fleury, C. & Braibant, V. Structural Optimization: A New Dual Method Using Mixed Variables, *International Journal for Numerical Methods in Engineering*, 1986, **23**, pp. 409-428.
5. Fleury, C. A Convex Linearization Method Using Second Order Information, in FEMCAD 88 (proceedings), IITT Int., Paris, France, 1988.
6. Fleury, C. First and Second Order Convex Approximation Strategies in Structural Optimization, *Structural Optimization*, 1989, **1**, pp. 3-10.
7. Haftka, R.T. & Gurdal, Z. *Elements of Structural Optimization*, Kluwer Academic Publishers, 1992.
8. Kodiyalam, S. Sensitivity Analysis in Static and Dynamic Problems, *Advanced Techniques in the Optimum Design of Structures*, ed. S. Hernandez, Computational Mechanics Publications, 1993.
9. Luenberger, D.G. *Linear and Nonlinear Programming*, 2nd Edition, Addison-Wesley, 1984.
10. Svanberg, K. The Method of Moving Asymptotes - A New Method for Structural Optimization, *International Journal for Numerical Methods in Engineering*, 1987, **24**, pp. 359-373.
11. Vanderplaats, G.N. *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, 1984.