A note on exploiting structure when using slack variables

by A.R. Conn[1], Nick Gould[2], and Ph.L. Toint[3]

Report 92/18(revised) November 19, 1993

[1] IBM T.J. Watson Research Center, P.O.Box 218, Yorktown Heights, NY 10598, USA
Email : arconn@watson.ibm.com

[2] CERFACS, 42 Avenue Gustave Coriolis, 31057 Toulouse Cedex, France, EC
Email : gould@cerfacs.fr or nimg@directory.rl.ac.uk
Current reports available by anonymous ftp from the directory
"pub/reports" on camelot.cc.rl.ac.uk (internet 130.246.8.61)

[3] Department of Mathematics, Facultés Universitaires ND de la Paix, 61, rue de Bruxelles,
B-5000 Namur, Belgium, EC
Email : pht@math.fundp.ac.be
Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be (internet 138.48.4.14)

Keywords : Large-scale problems, nonlinear optimization, numerical algorithms, slack variables, augmented Lagrangian.

Mtahematics Subject Classifications: 65k05, 90C30

[0]This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No F49620-91-C-0079. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

A note on exploiting structure when using slack variables

A. R. Conn, Nick Gould and Ph. L. Toint

November 19, 1993

Abstract

We show how to exploit the structure inherent in the linear algebra for constrained nonlinear optimization problems when inequality constraints have been converted to equations by adding slack variables and the problem is solved using an augmented Lagrangian method.

1 Introduction

In this note, we consider solving the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad l_y \leq c(x) \leq u_y \quad \text{and} \quad l_x \leq x \leq u_x \tag{1.1}$$

by introducing slack variables y to create the equivalent problem

$$\underset{x \in \Re^n, \; y \in \Re^m}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) - y = 0, \quad l_y \leq y \leq u_y \quad \text{and} \quad l_x \leq x \leq u_x. \tag{1.2}$$

The use of slack variables is, of course, a well-established technique for dealing with general inequalities in linear programming (see, for example, Dantzig, 1963, page 86, Murtagh, 1981 or Chvátal, 1983, page 14) and more recently nonlinear programming (see, for example, Murtagh and Saunders, 1978). The conversion to "standard form" has a number of linear-algebraic advantages for large problems, see for example Gill *et al.* (1981), page 190.

We attempt to solve (1.2) by a sequential minimization of the *augmented Lagrangian function* (see Powell, 1969 and Hestenes, 1969)

$$\Phi(x, y, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i (c_i(x) - y_i) + \frac{1}{2\mu} \sum_{i=1}^{m} (c_i(x) - y_i)^2, \tag{1.3}$$

within a region defined by the simple bounds

$$l_y \leq y \leq u_y \quad \text{and} \quad l_x \leq x \leq u_x, \tag{1.4}$$

where the components λ_i are Lagrange multiplier estimates and μ is a positive penalty parameter. Notice that we *do not* include the simple bounds in the augmented Lagrangian function.

At the heart of any iterative algorithm to minimize (1.3) (for fixed values of λ and μ), it is usual to construct a quadratic model of the augmented Lagrangian function and (approximately) minimize this within the region defined by the simple bounds, and, perhaps, a trust-region, on x and y. A simple-minded approach to this — in fact the approach taken within the LANCELOT code SBMIN (see Conn *et al.*, 1992a)— is to treat all variables in the same way. Thus slack variables are *not* treated differently from the problem variables x. If there are many slack variables relative to the number of problem variables— for instance, as would be the case for problems where a parameterized (or semi-infinite) constraint is approximated by a large number of representatives at discrete values of the parameter — the linear algebra will typically involve matrices of $O(n+m)$. The exact order will be determined by the number of *free* variables — i.e., those which are allowed to move unhindered by their bounds — at any instant. However, if slack variables are handled explicitly, we shall show that the linear algebra need only involve matrices of order $O(n)$.

The exploitation of the structure of slack variables has apparently also been used to advantage in the MINOS package of Murtagh and Saunders (1980) although this does not appear to have been publicized in the open literature.

Throughout this note, we shall use the following notation. Let $\rho \stackrel{\text{def}}{=} 1/\mu$. For given x and y, we define Lagrange multiplier updates

$$\bar{\lambda} \stackrel{\text{def}}{=} \lambda + \rho(c(x) - y). \tag{1.5}$$

We let $g(x)$ denote the gradient of $f(x)$, $a_i(x)$ denote the gradient $\nabla_x c_i(x)$, $A(x)$ be the Jacobian matrix whose rows are $a_i(x)^T$ and $\bar{H}(x, \lambda) = \nabla_{xx} f(x) + \sum_{i=1}^m \lambda_i \nabla_{xx} c_i(x)$ be the Hessian matrix of the Lagrangian function $f(x) + \lambda^T c(x)$.

We need to consider the derivatives of (1.3) with respect to x and y. The gradient is

$$\begin{pmatrix} g(x) + A(x)^T \bar{\lambda} \\ -\bar{\lambda} \end{pmatrix} \tag{1.6}$$

and an appropriate approximation to the Hessian matrix is

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} + \rho \begin{pmatrix} A(x)^T A(x) & -A(x)^T \\ -A(x) & I \end{pmatrix}, \tag{1.7}$$

where B is a suitable approximation to $\bar{H}(x, \bar{\lambda})$. The Hessian approximation aims to mimic the structure of the exact second derivatives of (1.3), using known first derivative information, while allowing the possibility of approximating the second order terms. Indeed, we may use the exact Hessian matrix if that is convenient.

Finally, let $\mathcal{M} = \{1, 2, \cdots, m\}$ and $\mathcal{N} = \{1, 2, \cdots, n\}$. Then, if v is a vector with m components and $\mathcal{I} \subseteq \mathcal{M}$, $v_{\mathcal{I}}$ is the vector whose components are v_i, $i \in \mathcal{I}$. Furthermore, if A is an m by n matrix and $\mathcal{J} \subseteq \mathcal{N}$, $A_{\mathcal{I},\mathcal{J}}$ is the matrix whose components are $A_{i,j}$, $i \in \mathcal{I}$, $j \in \mathcal{J}$.

2 The model

Let us assume that, in the course of an iterative method to solve (1.2), we have obtained the estimate (x, y) of the minimizer of (1.3). We now wish to obtain an improved estimate (x^+, y^+)

of the required minimizer. For convenience, we shall often drop the arguments x and y; Unless otherwise stated, all functions are assumed to be evaluated at x or (x, y) as appropriate.

In a typical descent method, we construct a simplified model of (1.3) and use an (approximate) minimizer of this model to predict the (unknown) minimizer of (1.3). The minimizer of the model within the bounds

$$l_x \leq x^+ \leq u_x \quad \text{and} \quad l_y \leq y^+ \leq u_y. \tag{2.1}$$

and, if a trust-region method is intended, a trust-region

$$\left\| \begin{pmatrix} x^+ - x \\ y^+ - y \end{pmatrix} \right\| \leq \Delta, \tag{2.2}$$

for some positive scalar Δ, is sought. We shall refer to the intersection of the regions defined by (2.1) and, if required, (2.2) as the *model-feasible region* and any point which lies in this region is said to be *model-feasible*. It is particularly convenient when the infinity norm is chosen to define the trust-region as then the sides of the simple bound "box" and the trust-region align.

We shall be concerned with the case when the quadratic model

$$\begin{aligned} \Phi_m(x^+, y^+) &\stackrel{\text{def}}{=} \Phi(x, y, \lambda, \rho) + ((x^+ - x)^T \quad (y^+ - y)^T) \begin{pmatrix} g + A^T \bar{\lambda} \\ -\bar{\lambda} \end{pmatrix} + \\ &\tfrac{1}{2}((x^+ - x)^T \quad (y^+ - y)^T) \begin{pmatrix} B + \rho A^T A & -\rho A^T \\ -\rho A & \rho I \end{pmatrix} \begin{pmatrix} x^+ - x \\ y^+ - y \end{pmatrix} \end{aligned} \tag{2.3}$$

is chosen to predict improvements x^+ and y^+ to x and y. This model is, of course, just a second-order Taylor series approximation using the approximate Hessian (1.7).

We allow the possibility that a model-feasible correction x^b and y^b for which $\Phi_m(x^b, y^b) \leq \Phi_m(x, y)$ has already been computed and that we are interested in computing model-feasible x^+ and y^+ for which $\Phi_m(x^+, y^+) \leq \Phi_m(x^b, y^b)$. In trust-region methods, the point (x^b, y^b) might be the (generalized) Cauchy point (see, for example, Moré, 1988 or Conn *et al.*, 1988). In linesearch methods, it is more normal for (x^b, y^b) to be (x, y).

We refer to (x^b, y^b) as the *base* point. We note that the gradient of the model at the base point is

$$\begin{pmatrix} g^b + A^T \lambda^b \\ -\lambda^b \end{pmatrix}, \tag{2.4}$$

where the multiplier estimates λ^b satisfy

$$\lambda^b = \bar{\lambda} + \rho A(x^b - x) - \rho(y^b - y) \tag{2.5}$$

and g^b is given by

$$g^b \stackrel{\text{def}}{=} g + B(x^b - x); \tag{2.6}$$

the quantity $g^b + A^T \lambda^b$ is nothing other than a first-order prediction of the gradient of the Lagrangian function at the point $x = x^b$ and $\lambda = \lambda^b$. We also note that the multiplier estimates (2.5) are first-order predictions of the multiplier updates (1.5) evaluated at the base point.

3 Linear Algebra

We assume without loss of generality that the first n_a problem variables, indexed by $\mathcal{A}_x$, and the first m_a slack variables, indexed by $\mathcal{A}_y$, are *active* (i.e., lie on one of their bounds) at the base point and are to be *fixed* during the current iteration. We denote the indices of the $n_i \equiv n - n_a$ free problem variables and $m_i \equiv m - m_a$ free slack variables by $\mathcal{I}_x = \mathcal{N} \setminus \mathcal{A}_x$ and $\mathcal{I}_y = \mathcal{M} \setminus \mathcal{A}_y$ respectively. Here *free* is intended to mean both variables which are *inactive* (i.e., lie away from their bounds) and variables which, although active, are free to move off their bounds. We will not concern ourselves here with how to decide which variables are free or fixed; the reader should consult, for example, Gill *et al.* (1981) for details of explicit active/working set strategies (Chapters 5 and 6) and Bertsekas (1982), page 248, for implicit ones.

3.1 Direct methods

We first consider direct methods, that is methods which rely solely on matrix factorizations, to solve the model problem. We note that, although we attempt to solve the model starting from (x^b, y^b), all the derivative information that we use is calculated at (x, y).

If we are using direct methods, the Newton correction (p, q) to the iterate (x^b, y^b) satisfies

$$\left(\begin{array}{cccccc} \multicolumn{2}{c}{B + \rho A^T A} & -\rho A^T_{\mathcal{A}_y,\mathcal{A}_x} & -\rho A^T_{\mathcal{I}_y,\mathcal{A}_x} & I & 0 \\ & & -\rho A^T_{\mathcal{A}_y,\mathcal{I}_x} & -\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} & 0 & 0 \\ -\rho A_{\mathcal{A}_y,\mathcal{A}_x} & -\rho A_{\mathcal{A}_y,\mathcal{I}_x} & \rho I & 0 & 0 & I \\ -\rho A_{\mathcal{I}_y,\mathcal{A}_x} & -\rho A_{\mathcal{I}_y,\mathcal{I}_x} & 0 & \rho I & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \end{array}\right) \cdot \left(\begin{array}{c} p_{\mathcal{A}_x} \\ p_{\mathcal{I}_x} \\ q_{\mathcal{A}_y} \\ q_{\mathcal{I}_y} \\ \pi_x \\ \pi_y \end{array}\right) = - \left(\begin{array}{c} g^b_{\mathcal{A}_x} + A^T_{\mathcal{M},\mathcal{A}_x} \lambda^b \\ g^b_{\mathcal{I}_x} + A^T_{\mathcal{M},\mathcal{I}_x} \lambda^b \\ -\lambda^b_{\mathcal{A}_y} \\ -\lambda^b_{\mathcal{I}_y} \\ 0 \\ 0 \end{array}\right), \tag{3.1}$$

where π_x and π_y are Lagrange multipliers associated with the active variables. Using the last two block equations of (3.1) to eliminate the variables

$$p_{\mathcal{A}_x} = 0 \quad \text{and} \quad q_{\mathcal{A}_y} = 0, \tag{3.2}$$

and extracting the second and fourth block equations, we obtain

$$\left(\begin{array}{cc} B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{M},\mathcal{I}_x} A_{\mathcal{M},\mathcal{I}_x} & -\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} \\ -\rho A_{\mathcal{I}_y,\mathcal{I}_x} & \rho I \end{array}\right) \left(\begin{array}{c} p_{\mathcal{I}_x} \\ q_{\mathcal{I}_y} \end{array}\right) = - \left(\begin{array}{c} g^b_{\mathcal{I}_x} + A^T_{\mathcal{M},\mathcal{I}_x} \lambda^b \\ -\lambda^b_{\mathcal{I}_y} \end{array}\right). \tag{3.3}$$

We may factorize the coefficient matrix of (3.3) to obtain

$$\begin{aligned} &\begin{pmatrix} B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{M},\mathcal{I}_x} A_{\mathcal{M},\mathcal{I}_x} & -\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} \\ -\rho A_{\mathcal{I}_y,\mathcal{I}_x} & \rho I \end{pmatrix} \\ = &\begin{pmatrix} I & -\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} \\ 0 & \rho I \end{pmatrix} \begin{pmatrix} B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{A}_y,\mathcal{I}_x} A_{\mathcal{A}_y,\mathcal{I}_x} & 0 \\ -A_{\mathcal{I}_y,\mathcal{I}_x} & I \end{pmatrix}. \end{aligned} \tag{3.4}$$

The important point here is that when we zero the upper right block of the matrix on the left-hand-side of (3.4), this also zeros the $A_{\mathcal{I}_y,\mathcal{I}_x}$ terms in its upper left block. Thus we may solve (3.3) by successively solving the pair of intermediate equations

$$\begin{pmatrix} I & -\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} \\ 0 & \rho I \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = - \begin{pmatrix} g^b_{\mathcal{I}_x} + A^T_{\mathcal{M},\mathcal{I}_x} \lambda^b \\ -\lambda^b_{\mathcal{I}_y} \end{pmatrix} \tag{3.5}$$

and

$$\begin{pmatrix} B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{A}_y,\mathcal{I}_x} A_{\mathcal{A}_y,\mathcal{I}_x} & 0 \\ -A_{\mathcal{I}_y,\mathcal{I}_x} & I \end{pmatrix} \begin{pmatrix} p_{\mathcal{I}_x} \\ q_{\mathcal{I}_y} \end{pmatrix} = \begin{pmatrix} v \\ w \end{pmatrix}. \tag{3.6}$$

The first of these, (3.5), yields that

$$w = \mu \lambda^b_{\mathcal{I}_y} \quad \text{and} \quad v = -g^b_{\mathcal{I}_x} - A^T_{\mathcal{A}_y,\mathcal{I}_x} \lambda^b_{\mathcal{A}_y}, \tag{3.7}$$

where once again we see complete cancellation between the $A_{\mathcal{I}_y,\mathcal{I}_x}$ terms in the second equation in (3.7), and both solutions are obtained without any inversions. Then we can obtain $p_{\mathcal{I}_x}$ from (3.6) by solving

$$\left(B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{A}_y,\mathcal{I}_x} A_{\mathcal{A}_y,\mathcal{I}_x} \right) p_{\mathcal{I}_x} = -(g^b_{\mathcal{I}_x} + A^T_{\mathcal{A}_y,\mathcal{I}_x} \lambda^b_{\mathcal{A}_y}) \tag{3.8}$$

and thus recover, again without inversion, $q_{\mathcal{I}_y}$ from

$$q_{\mathcal{I}_y} = \mu \lambda^b_{\mathcal{I}_y} + A_{\mathcal{I}_y,\mathcal{I}_x} p. \tag{3.9}$$

Hence, the only system of equations that needs an explicit solution, (3.8), requires the factorization of an n_i by n_i matrix. Note that, if the Newton equations are to correspond to the minimizer of a convex model, one needs to ensure that $B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A^T_{\mathcal{A}_y,\mathcal{I}_x} A_{\mathcal{A}_y,\mathcal{I}_x}$ is positive definite.

If the Newton step lies outside the model-feasible region, one can perform a linesearch for the model along the piecewise linear path obtained by projecting the arc $(x^b + \alpha p, y^b + \alpha q)$, $\alpha \geq 0$, back into the model-feasible region (see Bertsekas, 1982, page 76, or Conn *et al.*, 1988). Furthermore, additional Newton steps may be performed, using reduced or (perhaps) increased sets of free variables and starting from the newly calculated point, if a more accurate solution of the model problem is required (see, for instance, Conn *et al.*, 1992a, Section 3.2.3).

The authors are aware that other derivations of the above result are possible. The description given here is intended to emphasize the reduction in computational effort possible when the structure inherent in the linear system (3.3) is properly exploited.

3.2 Iterative methods

If we wish to use an iterative method to (approximately) solve the model problem, we need to ensure that the resulting search direction is a descent direction for the model at the base point. We now show that we may guarantee this merely by finding a vector $p_{\mathcal{I}_x}$ for which

$$p_{\mathcal{I}_x}^T (g_{\mathcal{I}_x}^b + A_{\mathcal{A}_y,\mathcal{I}_x}^T \lambda_{\mathcal{A}_y}^b) < 0. \tag{3.10}$$

For then, having obtained such a $p_{\mathcal{I}_x}$, we use (3.9) to find $q_{\mathcal{I}_y}$. It now follows from (3.2), (3.9) and (3.10) that, as

$$\begin{aligned} (p^T \quad q^T) \begin{pmatrix} g^b + A^T \lambda^b \\ -\lambda^b \end{pmatrix} &= p_{\mathcal{I}_x}^T (g_{\mathcal{I}_x}^b + A_{\mathcal{M},\mathcal{I}_x}^T \lambda^b) - q_{\mathcal{I}_y}^T \lambda_{\mathcal{I}_y}^b \\ &= p_{\mathcal{I}_x}^T (g_{\mathcal{I}_x}^b + A_{\mathcal{A}_y,\mathcal{I}_x}^T \lambda_{\mathcal{A}_y}^b) - \mu \lambda_{\mathcal{I}_y}^{bT} \lambda_{\mathcal{I}_y}^b < 0, \end{aligned} \tag{3.11}$$

the overall search direction is a descent direction for the model at the base point.

So long as $B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A_{\mathcal{A}_y,\mathcal{I}_x}^T A_{\mathcal{A}_y,\mathcal{I}_x}$ is positive definite, we can find a $p_{\mathcal{I}_x}$ which satisfies (3.10) by applying a descent method to (approximately) minimize the model

$$\Psi(p_{\mathcal{I}_x}) \stackrel{\text{def}}{=} \tfrac{1}{2} p_{\mathcal{I}_x}^T (B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A_{\mathcal{A}_y,\mathcal{I}_x}^T A_{\mathcal{A}_y,\mathcal{I}_x}) p_{\mathcal{I}_x} + p_{\mathcal{I}_x}^T (g_{\mathcal{I}_x}^b + A_{\mathcal{A}_y,\mathcal{I}_x}^T \lambda_{\mathcal{A}_y}^b) \tag{3.12}$$

starting from the estimate $p_{\mathcal{I}_x} = 0$. Hence, using the convexity of Ψ, we have that

$$p_{\mathcal{I}_x}^T (g_{\mathcal{I}_x}^b + A_{\mathcal{A}_y,\mathcal{I}_x}^T \lambda_{\mathcal{A}_y}^b) \le \Psi(p_{\mathcal{I}_x}) < \Psi(0) = 0, \tag{3.13}$$

which gives (3.10). An appropriate descent method is, for instance, the truncated conjugate-gradient method (see, for example, Dembo *et al.*, 1982 or Toint, 1981)

Once again, the new iterate may have to be projected back into the model-feasible region to maintain model-feasibility and, if desired, the appropriate iterative process can be restarted.

4 Discussion

A number of the options within the software package SBMIN require that the matrix (3.4) is formed and factorized. This has several disadvantages:

- The matrix is of dimension $n_i + m_i$ rather than the dimension n_i of

$$B_{\mathcal{I}_x,\mathcal{I}_x} + \rho A_{\mathcal{A}_y,\mathcal{I}_x}^T A_{\mathcal{A}_y,\mathcal{I}_x}. \tag{4.1}$$

  When there are many inequality constraints present, this implies that considerable extra work will be performed.

- When a direct method is used, even if the pivot sequence is chosen to eliminate the slack variables first, no account is taken of the fact that the Schur complement after m_i pivots (that is, the matrix which remains to be factorized after m_i pivots in the factorization of (3.4)) is precisely the matrix (4.1). That is, as we have already mentioned, there would be exact cancellation of the term $\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} A_{\mathcal{I}_y,\mathcal{I}_x}$ in the Schur complement in exact arithmetic. Failure to exploit this exact cancellation can be harmful in two ways. Firstly, the pivot ordering is based on a symbolic factorization, which would not recognize such a cancellation. Thus significantly more space may be reserved for the factorization than is in fact warranted, and indeed it is even conceivable that the factorization may fail for lack of space when in reality there is sufficient room to hold the nonzeros in the factors. Secondly, if no account is taken of the cancellation which should occur in the positions once occupied by the $\rho A^T_{\mathcal{I}_y,\mathcal{I}_x} A_{\mathcal{I}_y,\mathcal{I}_x}$ terms, unnecessary floating point operations may be performed to calculate values that we are entitled to treat as zero, but that may be "small" when calculated with finite precision.

- An iterative method may suffer for three main reasons. Firstly, the work per iteration will be larger as the system is both bigger and contains more nonzeros. Secondly, if (4.1) is positive definite, the spectrum of (3.4) will be larger than that of (4.1) (because of the interlacing eigenvalue property of the matrices following a block-elimination; see, for instance, Wilkinson, 1965, page 103), thereby decreasing the theoretical rate of convergence of many iterative methods including the conjugate gradient method. Finally, and perhaps not so importantly in practice, a finite convergence result would occur after at most $n_i + m_i$ rather than n_i iterations if infinite precision arithmetic were used.

We believe that the performance of our optimization package LANCELOT will improve considerably if the structure of the slack variables is properly exploited, especially when $m \gg n$. This will place the work per iteration at the same level as is possible for methods, such as those based on the sequential minimization of barrier or Lagrangian barrier functions (see, for example, Wright, 1992 or Conn *et al.*, 1992b), which treat inequality constraints directly.

5 Acknowledgement

The authors would like to thank an anonymous referee whose report was almost as long as the original manuscript and whose comments have greatly improved the final presentation.

References

[Bertsekas, 1982] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[Chvátal, 1983] V. Chvátal. *Linear Programming.* W. H. Freeman, New York, USA, 1983.

[Conn *et al.*, 1988] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25:433–460, 1988. See also same journal 26:764–767, 1989.

[Conn *et al.*, 1992a] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A).* Number 17 in Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

[Conn *et al.*, 1992b] A. R. Conn, Nick Gould, and Ph. L. Toint. Convergence properties of minimization algorithms for convex constraints using a structured trust region. Technical Report 92-069, Rutherford Appleton Laboratory, Chilton, England, 1992.

[Dantzig, 1963] G. B. Dantzig. *Linear Programming and Extensions.* Princeton Uinversity Press, Princeton, USA, 1963.

[Dembo *et al.*, 1982] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact-Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[Gill *et al.*, 1981] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London and New York, 1981.

[Hestenes, 1969] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[Moré, 1988] J. J. Moré. Trust regions and projected gradients. In M. Iri and K. Yajima, editors, *System Modeling and Optimization*, volume 113, pages 1–13, Berlin, 1988. Springer Verlag. Lecture Notes in Control and Information Sciences.

[Murtagh and Saunders, 1978] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical Programming*, 14:41–72, 1978.

[Murtagh and Saunders, 1980] B. A. Murtagh and M. A. Saunders. MINOS/Augmented user's manual. Technical Report SOL80-14, Department of Operations Research, Stanford University, Stanford, USA, 1980.

[Murtagh, 1981] B. A. Murtagh. *Advanced Linear Programming.* McGraw-Hill Book Co, New York, 1981.

[Powell, 1969] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298, London and New York, 1969. Academic Press.

[Toint, 1981] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, London, 1981. Academic Press.

[Wilkinson, 1965] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, UK, 1965.

[Wright, 1992] M. H. Wright. Interior methods for constrained optimization. *Acta Numerica*, 1:341–407, 1992.