A note on exploiting structure when using slack variables

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

FUNDP 92/17

November 16, 1992

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.

[1] Mathematical Sciences Department,
IBM T.J. Watson Research Center,
PO Box 218, Yorktown Heights, NY 10598, USA
[2] Central Computing Department,
Rutherford Appleton Laboratory,
Chilton, Oxfordshire, OX11 0QX, England
[3] Department of Mathematics,
Facultés Universitaires ND de la Paix,
B-5000 Namur, Belgium

Keywords : Large-scale problems, nonlinear optimization, numerical algorithms.

Mathematics 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 16, 1992

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.

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

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

We attempt to solve (1.2) by a sequential minimization of the *augmented Lagrangian function* (ALF)

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

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 ALF.

For given x and y, we define Lagrange multiplier updates

$$\bar{\lambda} = \lambda + (c(x) - y)/\mu \tag{1.5}$$

We let $g(x)$ denote the gradient $f(x)$, $a_i(x)$ 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.

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

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{\mu} \begin{pmatrix} A(x)^T A(x) & -A(x)^T \\ -A(x) & I \end{pmatrix}, \tag{1.7}$$

where B is a suitable approximation to $\bar{H}(x, \bar{\lambda})$.

At the heart of any iterative algorithm to minimize (1.3) (for fixed values of λ and μ), one normally constructs a quadratic model of the ALF and (approximately) minimizes 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.*, 1992b)— 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 $O(m) + O(n)$. The exact order will be determined by the number of *free* variables — i.e., those which lie away from 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 $O(n)$.

Throughout this note, we shall use the following notation. 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

At each iteration of an iterative method to solve (1.2) a simplified model of (1.3) will typically be (approximately) minimized within a region, henceforth called the *feasible* region, defined as the intersection of the simple bounds (1.4) and, perhaps, a trust region. 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

$$\begin{aligned} \Psi_m(\bar{x}, \bar{y}) \stackrel{\text{def}}{=} \Psi(x, y, \lambda, \mu) + ((\bar{x} - x)^T \quad (\bar{y} - y)^T) \begin{pmatrix} g(x) + A(x)^T \bar{\lambda} \\ -\bar{\lambda} \end{pmatrix} + \\ \tfrac{1}{2}((\bar{x} - x)^T \quad (\bar{y} - y)^T) \begin{pmatrix} B + \frac{1}{\mu} A(x)^T A(x) & -\frac{1}{\mu} A(x)^T \\ -\frac{1}{\mu} A(x) & \frac{1}{\mu} I \end{pmatrix} \begin{pmatrix} \bar{x} - x \\ \bar{y} - y \end{pmatrix} \end{aligned} \tag{2.1}$$

is chosen to predict improvements $\bar{x}$ and $\bar{y}$ 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 feasible correction $\hat{x}$ and $\hat{y}$ for which $\Psi_m(\hat{x}, \hat{y}) \leq \Psi_m(x, y)$ has already been computed and that we are interested in computing feasible $\bar{x}$ and $\bar{y}$ for which $\Psi_m(\bar{x}, \bar{y}) \leq \Psi_m(\hat{x}, \hat{y})$. We refer to $(\hat{x}, \hat{y})$ as the *current* iterate. We note that the gradient of the model at the current iterate is

$$\begin{pmatrix} \hat{g}(x) + A(x)^T \hat{\lambda} \\ -\hat{\lambda} \end{pmatrix}, \tag{2.2}$$

where the multiplier estimates $\hat{\lambda}$ satisfy

$$\hat{\lambda} = \bar{\lambda} + \frac{1}{\mu} A(x)(\hat{x} - x) - \frac{1}{\mu}(\hat{y} - y) \tag{2.3}$$

and $\hat{g}$ is given by

$$\hat{g}(x) = g(x) + B(\hat{x} - x). \tag{2.4}$$

We also note that the multiplier estimates (2.3) are predictions of the first-order updates (1.5) evaluated at the current iterate.

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 current iterate. We denote the indices of the $n_f \equiv n - n_a$ inactive problem variables and $m_f \equiv m - m_a$ inactive slack variables by $\mathcal{I}_x = \mathcal{N} \setminus \mathcal{A}_x$ and $\mathcal{I}_y = \mathcal{M} \setminus \mathcal{A}_y$ respectively.

3.1 Direct methods

If we are using direct methods, the Newton correction (p_x, p_y) to the iterate $(\hat{x}, \hat{y})$ satisfies

$$\begin{pmatrix} B + \frac{1}{\mu}A(x)^T A(x) & & -\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{A}_x]}(x)^T & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{A}_x]}(x)^T & I & 0 \\ & & -\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T & 0 & 0 \\ -\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{A}_x]}(x) & -\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x) & \frac{1}{\mu}I & 0 & 0 & I \\ -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{A}_x]}(x) & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x) & 0 & \frac{1}{\mu}I & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \end{pmatrix}.$$

$$\begin{pmatrix} (p_x)_{[\mathcal{A}_x]} \\ (p_x)_{[\mathcal{I}_x]} \\ (p_y)_{[\mathcal{A}_y]} \\ (p_y)_{[\mathcal{I}_y]} \\ \lambda_x \\ \lambda_y \end{pmatrix} = - \begin{pmatrix} \hat{g}_{[\mathcal{A}_x]}(x) + A_{[\mathcal{M},\mathcal{A}_x]}(x)^T \hat{\lambda} \\ \hat{g}_{[\mathcal{I}_x]}(x) + A_{[\mathcal{M},\mathcal{I}_x]}(x)^T \hat{\lambda} \\ -\hat{\lambda}_{[\mathcal{A}_y]} \\ -\hat{\lambda}_{[\mathcal{I}_y]} \\ 0 \\ 0 \end{pmatrix}, \tag{3.1}$$

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

$$(p_x)_{[\mathcal{A}_x]} = 0 \quad \text{and} \quad (p_y)_{[\mathcal{A}_y]} = 0, \tag{3.2}$$

and extracting the second and fourth block equations, we obtain

$$\begin{pmatrix} B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu}A_{[\mathcal{M},\mathcal{I}_x]}(x)^T A_{[\mathcal{M},\mathcal{I}_x]}(x) & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T \\ -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x) & \frac{1}{\mu}I \end{pmatrix} \begin{pmatrix} (p_x)_{[\mathcal{I}_x]} \\ (p_y)_{[\mathcal{I}_y]} \end{pmatrix}$$

$$= - \begin{pmatrix} \hat{g}_{[\mathcal{I}_x]}(x) + A_{[\mathcal{M},\mathcal{I}_x]}(x)^T \hat{\lambda} \\ -\hat{\lambda}_{[\mathcal{I}_y]} \end{pmatrix}. \tag{3.3}$$

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

$$\begin{pmatrix} B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu}A_{[\mathcal{M},\mathcal{I}_x]}(x)^T A_{[\mathcal{M},\mathcal{I}_x]}(x) & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T \\ -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x) & \frac{1}{\mu}I \end{pmatrix}$$

$$= \begin{pmatrix} I & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T \\ 0 & \frac{1}{\mu}I \end{pmatrix} \begin{pmatrix} B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x) & 0 \\ -A_{[\mathcal{I}_y,\mathcal{I}_x]}(x) & I \end{pmatrix}. \tag{3.4}$$

The important point here is that when we zero the upper right block of the matrix, 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

$$\begin{pmatrix} I & -\frac{1}{\mu}A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T \\ 0 & \frac{1}{\mu}I \end{pmatrix} \begin{pmatrix} (q_x)_{[\mathcal{I}_x]} \\ (q_y)_{[\mathcal{I}_y]} \end{pmatrix} = - \begin{pmatrix} \hat{g}_{[\mathcal{I}_x]}(x) + A_{[\mathcal{M},\mathcal{I}_x]}(x)^T\hat{\lambda} \\ -\hat{\lambda}_{[\mathcal{I}_y]} \end{pmatrix} \tag{3.5}$$

and

$$\begin{pmatrix} B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x) & 0 \\ -A_{[\mathcal{I}_y,\mathcal{I}_x]}(x) & I \end{pmatrix} \begin{pmatrix} (p_x)_{[\mathcal{I}_x]} \\ (p_y)_{[\mathcal{I}_y]} \end{pmatrix} = \begin{pmatrix} (q_x)_{[\mathcal{I}_x]} \\ (q_y)_{[\mathcal{I}_y]} \end{pmatrix}. \tag{3.6}$$

The first of these, (3.5), yields that

$$(q_y)_{[\mathcal{I}_y]} = \mu\hat{\lambda}_{[\mathcal{I}_y]} \quad \text{and} \quad (q_x)_{[\mathcal{I}_x]} = -\hat{g}_{[\mathcal{I}_x]}(x) - A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T\hat{\lambda}_{[\mathcal{A}_y]}, \tag{3.7}$$

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

$$\left[B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)\right](p_x)_{[\mathcal{I}_x]} = -(\hat{g}_{[\mathcal{A}_y]}(x) + A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T\hat{\lambda}_{[\mathcal{A}_y]}) \tag{3.8}$$

and thus recover, again without inversion, $(p_y)_{[\mathcal{I}_y]}$ from

$$(p_y)_{[\mathcal{I}_y]} = \mu\hat{\lambda}_{[\mathcal{I}_y]} + A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)p_x. \tag{3.9}$$

Hence, the only system of equations that needs an explicit solution, (3.8), requires the factorization of an n_f by n_f matrix. Note that, if the Newton equations are to correspond to the minimizer of a convex model, one needs to ensure that $B_{[\mathcal{I}_x,\mathcal{I}_x]}+\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)$ is positive definite.

If the Newton step lies outside the feasible region, one can perform a linesearch along the piecewise linear path obtained by projecting the arc $(x + \alpha p_x, y + \alpha p_y)$, $\alpha \geq 0$, back into the feasible region (see Bertsekas, 1982 or Conn *et al.*, 1988). Furthermore, additional Newton steps may be performed using reduced sets of free variables if so required (see, for instance, Conn *et al.*, 1992b, Section 3.2.3).

3.2 Iterative methods

If we wish to use an iterative method, we merely need to find a vector $(p_x)_{[\mathcal{I}_x]}$ for which

$$(p_x)^T_{[\mathcal{I}_x]}(\hat{g}_{[\mathcal{A}_y]}(x) + A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T\hat{\lambda}_{[\mathcal{A}_y]}) < 0. \tag{3.10}$$

We might achieve this by, for instance, applying a CG truncated-Newton method (see, for example, Dembo *et al.*, 1982 or Toint, 1981) to approximately minimize the model

$$\tfrac{1}{2}(p_x)^T_{[\mathcal{I}_x]}[B_{[\mathcal{I}_x,\mathcal{I}_x]}+\frac{1}{\mu}A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)](p_x)_{[\mathcal{I}_x]}+(p_x)^T_{[\mathcal{I}_x]}(\hat{g}_{[\mathcal{A}_y]}(x)+A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T\hat{\lambda}_{[\mathcal{A}_y]}). \tag{3.11}$$

Using such a $(p_x)_{[\mathcal{I}_x]}$, we use (3.9) to find $(p_y)_{[\mathcal{I}_y]}$. It then follows from (3.2), (3.9) and (3.10) that, as

$$\begin{aligned} &(p_x^T \quad p_y^T)\begin{pmatrix} \hat{g}(x) + A(x)^T\hat{\lambda} \\ -\hat{\lambda} \end{pmatrix} \\ &\quad = (p_x)^T_{[\mathcal{I}_x]}(\hat{g}_{[\mathcal{I}_x]}(x) + A_{[\mathcal{M},\mathcal{I}_x]}(x)^T\hat{\lambda}) - (p_y)^T_{[\mathcal{I}_y]}\hat{\lambda}_{[\mathcal{I}_y]} \\ &\quad = (p_x)^T_{[\mathcal{I}_x]}(\hat{g}_{[\mathcal{I}_x]}(x) + A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T\hat{\lambda}_{[\mathcal{A}_y]}) - \mu\hat{\lambda}^T_{[\mathcal{I}_y]}\hat{\lambda}_{[\mathcal{I}_y]} < 0, \end{aligned} \tag{3.12}$$

the overall search direction is a descent direction for the ALF from the current iterate. Once again, the new iterate may have to be projected back into the feasible box to maintain feasibility and, if desired, the CG process can be restarted. As the number of free variables strictly decreases each time a restart is undertaken, only a finite number of restarts are possible.

4 Discussion

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

- The matrix is of dimension $n_f + m_f$ rather than the dimension n_f of $B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu} A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)$. 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_f pivots is $B_{[\mathcal{I}_x,\mathcal{I}_x]} + \frac{1}{\mu} A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{A}_y,\mathcal{I}_x]}(x)$. That is, as we have already mentioned, there would be exact cancellation of the term $\frac{1}{\mu} A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)$ in the Schur complement in exact arithmetic. This is bad as a) the ordering is based on a symbolic factorization, which would not recognize such a cancellation, and b) it is likely that there will be "small" rounding numbers in the positions once occupied by the $\frac{1}{\mu} A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)^T A_{[\mathcal{I}_y,\mathcal{I}_x]}(x)$ term when finite precision arithmetic is used. This may lead to bad orderings of the variables for the factorization and unnecessary fill-ins.

- An iterative method will suffer because a) the work per iteration will be larger, b) the spectrum is likely to be stretched as the matrix is bigger and c) (for what its worth) a finite convergence result would occur after at most $n_f + m_f$ rather than n_f iterations.

It seems to the authors that the performance of their optimization package LANCELOT can be considerably improved if the structure of the slack variables is properly exploited, especially when $m \gg n$. This should improve the work per iteration to 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.*, 1992a), which treat inequality constraints directly.

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.

[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. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. Technical Report 92/07, Department of Mathematics, FUNDP, Namur, Belgium, 1992.

[Conn *et al.*, 1992b] 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.

[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.

[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.

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