AMO - Advanced Modeling and Optimization Volume 1, Number 3, 1999

Numerical Experiments with Methods for Solving the KKT Equations

M. de F. G. Hernandez and M.C. Bartholomew-Biggs
Numerical Optimisation Centre, University of Hertfordshire

September 20, 1999

ABSTRACT

In many seqential quadratic programming algorithms for constrained optimization the calculation of an effective search direction depends on the (estimated) Hessian of the Lagrangian being positive definite on the null space of the active constraints. This paper reports some numerical experience with two techniques for checking the properties of the Hessian and, if necessary, modifying it during the solution of the KKT equations.

1 Introduction

In this paper we are concerned with sequential quadratic programming (SQP) techniques for solving the general nonlinear programming problem

$$\text{Minimize } F(x) \;\; \text{s.t. } c_i(x) = 0, \;\; i = 1,..,m; \quad c_i(x) \geq 0, \;\; i = m+1,..,m' \tag{1.1}$$

where x is a real n-vector. In particular we shall consider SQP methods which use exact second derivatives of the function and constraints to form an estimate of the Lagrangian Hessian

$$B = \nabla^2 F(x) - \sum_{i=1}^{m'} v_i \nabla^2 c_i(x) \tag{1.2}$$

where x is an approximate solution of (1.1) and $(v_1, ..., v_{m'})^T$ is a vector of approximate Lagrange multipliers.

As we shall explain more fully below, the calculation of an effective search direction away from x depends upon the properties of B. In particular, if B is positive definite on a certain subspace (the null space of *active* constraints) then a good serach direction will be obtained. But if B does not have this property then the search direction may not yield an improved solution estimate.

One way of avoiding unsuitable SQP search directions is to use a quasi-Newton approximation to B which is forced to be positive definite. This technique works quite well for small, dense problems; but when we deal with large sparse ones the exact second derivative approach may have some advantages. For instance, the additional information in $\nabla^2 F$ and the $\nabla^2 c_i$ may mean that fewer (expensive) iterations are needed. It may also be easier to exploit sparsity and structure in (1.1) when exact second derivatives are used because quasi-Newton techniques which seek to impose sparsity and positive definiteness on an approximate Hessian are often less effective than their dense counterparts. (It is worth mentioning, however, that Bartholomew-Biggs and Hernandez (1995) have reported some surprisingly good results with an update which forces B to be a diagonal matrix.)

Once we accept that we may have to deal with an indefinite B it follows that the SQP direction-finding procedure must incorporate some method for checking the current matrix (1.2) and for modifying it if necessary. In this paper we describe two such checking and correcting procedures, based on work by Gould (1986). These procedures are outlined in section 3; and we give some illustrative numerical results in section 4. First, however, in section 2 we describe the particular SQP algorithm which is the basis for our investigation.

2 The algorithm opalqp

The SQP technique **opalqp** was originally proposed by Bartholomew-Biggs (1987) and adapted for larger problems by Bartholomew-Biggs and Hernandez (1995). Each iteration of **opalqp** obtains a search direction p away from the current point, x, by solving the QP subproblem

$$\text{Minimize } \frac{p^T B p}{2} + \nabla F(x)^T p \tag{2.1}$$

subject to

$$\nabla c_i(x)^T p + c_i(x) = -r(u_i - v_i) \quad i = 1, \ldots, m \tag{2.2}$$

and

$$\nabla c_i(x)^T p + c_i(x) \geq -r(u_i - v_i) \quad i = m+1, \ldots, m' \tag{2.3}$$

where B is defined in (1.2). The penalty parameter r and the multiplier estimates v (also appearing in (1.2)) are specified at the start of each iteration; and the values $u_1, .., u_{m'}$ are obtained during the solution of (2.1) - (2.3) as new estimates of the Lagrange multipliers. Typically, r and v will remain constant for a number of iterations, but after a “successful” step (one which reduces the errors in the first order optimality conditions) r may be decreased and v replaced by u.

The vector p which solves (2.1) – (2.3) is such that the point $x + p$ is an approximation to the minimum of the augmented Lagrangian function

$$M(x,v,r) = F(x) + \frac{1}{2r}\sum_{i=1}^{m}(c_i(x) - rv_i)^2 + \frac{1}{2r}\sum_{i=m+1}^{m'}(\text{Min}(0,\ c_i(x) - rv_i))^2. \quad (2.4)$$

The method used to determine p and u to solve (2.1) - (2.3) is an active set technique which involves repeated solutions of systems of the form

$$Bp - A^T u = -\nabla F \quad (2.5)$$

$$-Ap - ru = g - rv \quad (2.6)$$

Here A is the $t \times n$ matrix whose rows are the vectors $\nabla c_i(x)$ corresponding to the t constraints that are currently active. (For the purpose of this paper it is sufficient to remark that the active set typically includes all the equality constraints along with the violated inequalities.) Similarly g denotes the t-vector of active constraint values. In (2.5), (2.6) and in what follows u and v denote t-vectors of approximate Lagrange multipliers associated with the active constraints.

The system (2.5),(2.6) may be solved several times on each iteration of **opalqp**, with different A and g and this usually accounts for most of the computational effort required by the algorithm. Since large-scale optimization problems often involve sparse constraint Jacobians and objective function Hessians, it may be advantageous to use a sparse data structure to represent the coefficient matrix

$$K = \begin{pmatrix} B & -A^T \\ -A & -rI \end{pmatrix} \quad (2.7)$$

A possible data structure for this purpose has been discussed by Hernandez (1995).

A suitable algorithm for solving (2.5), (2.6) is LDL^T factorization (see, for instance, Duff et al (1990)). An implementation of this approach is the routine **MA27** (Duff & Reid (1982)). This splits (a permuted form of) K into the product LDL^T where L is lower triangular and D consists of 1×1 and 2×2 diagonal elements. The pivot sequence is designed to minimize fill-in (which is important for the case when B and A are sparse) and is obtained initially during an analysis phase of **MA27**. This initial sequence may be modified later for stability reasons — e.g. if a scalar pivot is found to be unsuitably small. We shall take advantage of this existing mechanism for modifying pivots in constructing a revised form of **MA27** which allows certain negative pivot elements to be rejected or corrected, as described in section 3.

To force global convergence to a solution of (1.1), **opalqp** uses a linesearch to control the step along the search direction p. The new solution estimate is

$$x^+ = x + sp$$

where s is chosen to ensure that

$$M(x^+,\ v,\ r) < M(x,\ v,\ r).$$

The choice $s = 1$ is often suitable (and superlinear convergence depends upon this being the case near the solution); but otherwise a simple Armijo search can be used to obtain a new point which satisfies the Wolfe conditions. The presence of this linesearch phase in **opalqp** means that it is important for the vector p produced by (2.1) – (2.3) to be a descent direction with respect to the augmented Lagrangian M. It is this descent property that depends crucially on B being positive definite in the null space of A.

We now observe that there is strong connection between the search direction p and the Newton step for the augmented Lagrangian M. This is most easily explained if we consider the problem (1.1) with only equality constraints. In this case the active constraints g and their Jacobian A simply come from $c_i(x),\ i = 1,...,m$ and so

$$M = F + \frac{1}{2r}(g - rv)^T(g - rv) \text{ and } \nabla M = \nabla F + \frac{1}{r}A^T(g - rv).$$

Now it can be shown (Bartholomew-Biggs (1987)) that the vector p given by solving (2.5), (2.6) also satisfies

$$(B + \frac{1}{r}A^T A)p = -\nabla F - \frac{1}{r}A^T(g - rv) = -\nabla M. \tag{2.8}$$

For equality constrained problems the Hessian matrix of (2.4) is

$$\nabla^2 M = W + \frac{1}{r}A^T A \quad \text{where} \quad W = \nabla^2 F + \frac{1}{r}\sum_{i=1}^{m}(g_i - rv_i)\nabla^2 g_i. \tag{2.9}$$

Now the matrix W in (2.9) can be regarded as an estimate of the Hessian B as $g_i \to 0$ near a solution of (1.1). Hence the coefficient matrix in (2.8) is approximately $\nabla^2 M$ and p can be viewed as a Newton-like step towards a stationary point of M. Thus p will be a descent direction towards a minimum of M if the matrix

$$H = B + \frac{1}{r}A^T A \tag{2.10}$$

is positive definite. For this to be true it is sufficient but not necessary for B to have all positive eigenvalues. We shall say that a (possibly indefinite) matrix B is *sufficiently positive definite* if (2.10) is positive definite.

Although the above discussion deals with the equality constrained form of (1.1), there is a similar correspondence between the Newton vector for (2.4) and the solution p of (2.1) - (2.3) when inequality constraints are present.

We have already mentioned that **opalqp** uses (2.5), (2.6) to obtain p. This is preferable to using (2.8) both because it preserves any sparsity in B and A and also because the system (2.8) can be ill-conditioned if r is small. What we require, therefore, is a means of detecting, during the process of solving (2.5) and (2.6), whether or not B is sufficiently positive definite. It will then be possible to make changes to B to ensure that p is a descent direction.

In the next section we will show how to deduce the positive definiteness of H from information which is available during the factorization of K by **MA27**.

3 A modified factorization for the KKT matrix

We begin this section by quoting some results which use the concept of inertia.

Definition The *inertia* of the $n \times n$ matrix H (2.10) is the triple

$$\text{In } H = (h_+, h_-, h_0) \tag{3.11}$$

where h_+, h_-, h_0 are respectively the numbers of positive, negative and zero eigenvalues. Obviously, $h_+ + h_- + h_0 = n$.

In what follows we shall also use

$$\text{In } K = (k_+,\ k_-,\ k_0) \ \text{ with } \ k_+ + k_- + k_0 = n + t$$

and, assuming B is non-singular,

$$\text{In } B = (b_+,\ b_-,\ 0) \ \text{ with } \ b_+ + b_- = n.$$

Theorem (Gould (1986)) With the inertias defined above and providing $r > 0$,

$$(k_+,\ k_-,\ k_0) = (h_+,\ h_-, h_0) + (0, t, 0)$$

Corollary (Gould (1986)) H is positive definite if and only if $k_0 = 0$ and $k_- = t$.

Our aim is to modify K (if necessary) so as to ensure

$$\text{In } K = (n, t, 0). \tag{3.12}$$

The corollary then ensures that H is positive definite. Gould (1986) also shows that (3.12) guarantees that B is positive definite in the null space of the active constraint Jacobian, A.

As explained by Hernandez (1995), we can determine whether a scalar pivot element chosen by the routine **MA27** is associated with a B-row or an A-row from the original matrix K. Similarly we can tell whether a 2×2 block pivot is made up of two B-rows, two A-rows or an A-row and a B-row. During the factorization we need place no restrictions on the signs of scalar A-pivots or on the eigenvalues of 2×2 block A-pivots. However, some conditions must hold for the B-row pivots to ensure that the number of negative eigenvalues of K should not exceed t. Specifically, we want all scalar B-pivots to be positive and all 2×2 block B-pivots to be positive definite. Any 2×2 block pivots associated with A and B rows must be indefinite, but with the positive eigenvalue associated with the B-component.

With these requirements in mind, all B-pivots can be checked as **MA27** proceeds. If an unsuitable B-pivot is encountered there are two possible remedies. In the first, the pivot is corrected immediately (according to a procedure described in a little more detail below). The second approach, however, seeks to avoid modifications to B if possible; and therefore the initial response to an unsuitable B-pivot is to reject it and seek an alternative from the remaining rows of K. Specifically, the algorithm checks to see if any A-pivots are available that could be used instead. It is only when all A-pivots are exhausted that a modification to a B-pivot can occur.

In the experiments described below, the actual modification to a B-pivot is carried out as follows. Negative scalars are replaced by their absolute value (except that very small positive or negative values are replaced by a positive threshold value). For 2×2 pivots the eigenvalues are computed. Unacceptable negative eigenvalues are replaced by their absolute value (again subject to a small positive threshold) and the pivot block is then re-constructed with these new eigenvalues. This approach is not necessarily the most sophisticated or imaginative way of correcting the K matrix; but it will serve our purpose here, which is principally to compare the effectiveness of the two strategies mentioned in the previous paragraph — namely the immediate corrrection of unsuitable B-pivots (which we shall call Method 1) and their possibly deferred correction (Method 2). Clearly the motivation for Method 2 is to reduce the number of avoidable changes in the matrix B.

It should be noted, however, that in the implementation of Method 2 it may prove necessary to weaken the stability test which in **MA27**. This test prevents a pivot element being used if its value (or one of its eigenvalues in the case of a 2×2 block) is too close to zero. In order to allow the factorization to proceed even if several B-pivots are rejected at some stage, it may be necessary to allow a pivot to be used which would have failed the original stability requirement.

4 Numerical experiments

In this section we give some results obtained with **opalqp** using the **MA27** variants, Method 1 and Method 2, described in the previous section. We also quote results obtained by another version of **opalqp** discussed by Bartholomew-Biggs and Hernandez (1995). This involves a simple strategy for ensuring the descent property $p^T M(x, v, r) < 0$ by modifying B to be positive definite. This is done by replacing B by $B + \Lambda$ where Λ is a diagonal matrix with suitably large positive elements. The elements of Λ are chosen by using the Gershgorin disk theorem to estimate the number of negative eigenvalues of B and then assigning large enough values to the appropriate Λ_{ii} to ensure that $B + \Lambda$ has all positive eigenvalues. The approach has the merit of preserving sparsity in B; and it will sometimes leave B unchanged or, at least, make relatively few modifications. However it is clear that this strategy will usually lead to uneccessary changes, since no attempt is made to limit the effect of Λ to the null space of the constraints. We shall refer to this as Method 0 and it uses the standard form of **MA27** because (2.5), (2.6) have a positive definite matrix in the top left partition and hence K need not be modified to force (3.12) to hold.

Among the problems we consider are the optimal control and economic growth examples quoted by Murtagh and Saunders (1982). Both these problems involve calculations over T time steps, where T can be viewed as a parameter which determines the size of the problem. Using the notation given by Murtagh and Saunders, the starting points are

$$x_t = 0, \;\; y_t = -1, \;\; u_t = -0.2 \quad \text{for } t = 1, ..., T$$

for the optimal control problem and

$$C_t = C_0 = 0.95, \;\; K_T = K_0 = 3.0, \;\; I_t = I_0 = 0.05 \quad \text{for } t = 1, ..., T$$

for the growth problem.

We also consider a scheduling problem (Apampa, private communication) which is of the form

$$\text{Minimize} \;\; \sum_{i=1}^{n} \frac{b_i}{a_i x_i + x_i^{-1} \sum_{j \neq i}(a_j x_j + b_j)}$$

subject to

$$\sum_{i=1}^{n} a_i x_i + b_i \leq 6800 \quad \text{and} \;\; x_i \geq 0 \;\; i = 1, .., n$$

The values of the coefficients a_i, b_i are

$$a_i = 1 + \frac{i}{20}, \quad b_i = 2 + \frac{i}{5} \;\; \text{for} \;\; i = 1, .., n$$

The starting point used is $x_i = 1, \; i = 1, ..., n$.

We have solved each of these problems using an implementation of **opalqp** in Fortran 77 which calls either the standard routine **MA27** or one of the modified versions, Method 1 and Method 2, described in the previous section. The numerical tests were performed on a Sun Ultra Enterprise 3000 in the Learning Resources Centre at the University of Hertfordshire. For the purposes of these experiments we made no particular attempt to choose the convergence parameters of **opalqp** to maximize its performance, since our purpose here is not a comparison with other SQP methods but an investigation of the relative merits of different ways of handling an indefinite Hessian B.

We now summarise the results that were obtained. For each solution in the tables below we quote the execution time together with the number of iterations (function evaluations in brackets). We also give the total number of KKT systems (2.5), (2.6) that were solved (with a bracketed figure showing what percentage of these solutions involved modifications to B).

Results for the optimal control problem

Problem size n = 398, m' = 663

	time (sec)	iterations	KKT solutions
Method 0	21	39(40)	434(100%)
Method 1	23	39(40)	441(93%)
Method 2	23	40(41)	431(0%)

Problem size n = 794, m' = 1323

	time (sec)	iterations	KKT solutions
Method 0	42	39(40)	434(100%)
Method 1	46	39(40)	441(93%)
Method 2	56	40(41)	431(0%)

Results for the economic growth problem

Problem size n = 396, m' = 793

	time (sec)	iterations	KKT solutions
Method 0	100	42(110)	1928(100%)
Method 1	146	46(120)	2790(95%)
Method 2	110	48(115)	2314(29%)

Problem size n = 450, m' = 901

	time (sec)	iterations	KKT solutions
Method 0	210	59(151)	3645(100%)
Method 1	422	72(198)	6835(93%)
Method 2	295	71(176)	5287(39%)

Results for the scheduling problem

Problem size n= 150, m' = 151

	time (sec)	iterations	KKT solutions
Method 0	46	24(99)	167(100%)
Method 1	47	11(33)	100(26%)
Method 2	41	9(37)	52(63%)

Problem size n=200, m' = 201

	time (sec)	iterations	KKT solutions
Method 0	124	32(113)	194(100%)
Method 1	81	8(37)	82(22%)
Method 2	42	6(25)	25(56%)

A number of comments can be made about these results:

(a) It is only on the scheduling problem that **opalqp** with the modified **MA27** significantly outperforms the version which simply forces B to be positive definite. We see that the numbers of iterations and KKT solutions are significantly smaller for both Method 1 and Method 2 than for Method 0. On the optimal control problem all versions of **opalqp** require about the same number of iterations, but Methods 1 and 2 use more KKT solutions and hence more execution time. For the economic growth problem, Method 0 does best both in terms of iterations and KKT solutions.

(b) On the first two problems, Method 2 performs as expected in that it makes appreciably fewer modifications to B during the factorization than Method 1. On the scheduling problem, however, the *percentage* of KKT solutions involving changes to B is higher for Method 2 than Method 1. The Gershgorin disk based Method 0 makes changes to diagonal terms of B on every iteration in these examples. Results on the first two problems, however, show that such modifications do not necessarily inhibit convergence.

(c) On all the examples we observe that Method 2 needs fewer KKT solutions than Method 1. This suggests that it is better to solve (2.1) - (2.3) by a method which tries to avoid modifying B. However it also appears from the execution times for the optimal control example that there is a time penalty associated with Method 2's policy of trying and rejecting several pivots at certain stages.

(d) The fact that on four out of six examples Method 0 needs fewer KKT solutions indicates that the QP subproblem (2.1) – (2.3) is easier to solve if it involves a positive definite matrix B, rather than an indefinite one which may be implicitly modified during the solution process.

The overall effect of these observations is somewhat equivocal. Relative to one another, the modified factorization methods have performed more or less as one might have expected. The fact that they do not easily outperform the less sophisticated Method 0 is rather more surprising and disappointing. Additional tests carried out by Hernandez (1995) show a similar mix of performance.

5 Discussion and conclusions

The results we have presented make a modest contribution to a problem of current interest — namely the practical computation of effective search directions in SQP methods for constrained optimization when second derivative information indicates that the Lagrangian does not have positive curvature on the null space of the currently active constraints. We have investigated the performance of two modified forms of the factorization routine **MA27** used within the QP subproblem. Both make implicit changes to the elements of the Hessian matrix (1.2) as the factorization proceeds, although one of them aims to avoid such modifications as much as possible. Our computational results show that, while the new methods are successful, they do not necessarily perform better than a cruder strategy which forces the Hessian to be positive definite.

Among other approaches to the same problem we mention the work of Forsgren and Murray (1993) who discuss a particular pivoting strategy for the LDL^T factorization of a KKT matrix similar to K in (2.5), (2.6). This ensures that when a $(n-t)\times(n-t)$ lower right hand block remains to be factorized it will involve all the rows of B which relate to the null space of A. One can then deal with this bottom block by a modified Cholesky approach of the kind described by Gill, Murray and Wright (1981) or Schnabel and Eskow (1988). Such techniques implicitly modify the matrix, if necessary, to make it positive definite. We have not dealt with the work of Forsgren and Murray in the main part of the paper because it is related to a different SQP method; and unlike **opalqp** this uses a system of equations in which the bottom right hand partition of (2.7) is replaced by a zero matrix.

Another approach which we have not yet tried involves the use of a different factorization of the K matrix. Instead of finding the LDL^T factors we could use, for instance, a sequence of Householder transformations to express K as the product $K = QTQ$ where Q is a symmetric orthonormal matrix and T is tridiagonal. The solution to a system

$$Ky = z$$

can be found by solving

$$T\hat{y} = Qz$$

(which can be done very cheaply) and then setting $y = Q\hat{y}$. For a dense matrix K the QTQ factorization is more expensive than the LDL^T one; but for sparse problems there may not be a significant disadvantage. An advantage of the approach is that no pivoting strategy is needed. Moreover, the eigenvalues of T are the same as those of K; and we can use the Sturm sequence theorem very easily to determine the inertia of the tridiagonal matrix T. Hence any changes to the matrix B to make it sufficiently positive definite can be delayed until the factorization of K is complete and then carried out in terms of changes to some diagonal elements of T. Further research on this idea is in progress.

6 References

Bartholomew-Biggs, M.C., The development of recursive programming methods based on the augmented Lagrangian, Mathematical programming Study 31, 21-41 (1987)

Bartholomew-Biggs, M.C. and Hernandez, M de F. G., Using the KKT matrix in an augmented Lagrangian SQP method for sparse constrained optimisation, Journal of Optimization Theory and Applications, April 1995

Duff, I.S. and Reid, J.K., A set of Fortran subroutines for solving sparse symmetric sets of linear equations, Report R10533, AERE Harwell (1982)

Duff, I.S. Gould, N.I.M., Reid, J.K., Scott, J.A. and Turner, K., The factorization a sparse symmetric indefinite matrix, Report RAL-90-066, Rutherford Appleton Laboratory (1990)

Forsgren, A.L. & Murray, W., Newton methods for large-scale equality constrained minimization, SIAM Journal on Matrix Analysis and Applications, 15, 560-587 (1993)

Gill, P.E. Murray, W. & Wright, M., Practical Optimization, Academic Press, London and new York (1981)

Gould, N.I.M., On the accurate determination of search directions for simple differentiable penalty functions, IMA Journal of Numerical Analysis, 6, 357-372 (1986)

Hernandez, M. de F. G., Algorithms for large sparse constrained optimization, Ph D Thesis, University of Hertfordshire, (1995)

Murtagh, B.A. & Saunders, M.A., A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, Mathematical Programming Study 16, 84-117 (1982)

Schnabel, R. & Eskow, E., A new modified Cholesky factorization, Report CU-CS-415-88, University of Colorado (1988)