AMO - Advanced Modeling and Optimization, Volume 5, Number 3, 2003

A Newton Method with a Two-dimensional Line Search

M.C.Bartholomew-Biggs

Numerical Optimisation Centre, University of Hertfordshire

ABSTRACT

This paper deals with a modified version of Newton's method for unconstrained minimization in which a search direction of the form $s = -\alpha g + \beta p$ is obtained by fitting a quadratic model to the objective function in the plane of the Newton direction p and the steepest descent direction $-g$. This composite search direction can be used as a fall-back option when the standard Newton step proves ineffective – e.g., when the Hessian matrix is not positive definite and p represents a move towards a saddle point or maximum.

The method proposed for determining α and β resembles a two variable trust-region approach. Two prototype algorithms are discussed: one uses s in a trust-region framework and the other performs a line search. In both methods the computational overheads are comparable with those for a standard Newton method because only one linear system needs to be solved per iteration.

1 Introduction

In this paper we consider the unconstrained minimization of a continuous and twice-differentiable function $f(x)$ by Newton's method. We use the notation $G = \nabla^2 f(x)$ and $g = \nabla f(x)$. When G is positive definite, the basic Newton iteration generates a new point, x^+, from the current one, x, by calculating

$$p = -G^{-1}g; \quad x^+ = x + \gamma p$$

where γ is chosen by a line search to satisfy the Wolfe conditions. This iteration is usually quadratically convergent, provided the choice $\gamma = 1$ is acceptable near the solution. Practical difficulties arise, however, when x is far from the solution and the Hessian is not positive definite because, in this case, p may not be a descent direction. This is just one of the reasons why the Newton method has often been neglected in favour of quasi-Newton or conjugate-gradient approaches to the unconstrained minimization problem. Other practical disadvantages are associated *(a)* with the need to provide exact second derivatives and *(b)* with the costs of solving the linear system $Gp = -g$. The advent of efficient tools for automatic differentiation (see, e.g. Christianson (1992)) has helped alleviate the labour of getting second derivatives. It has also been noted that, for some problems at least, it is possible to compute the Newton direction for an n-variable problem in much less than $O(n^3)$ operations. This can be seen, for instance, in the Pantoja algorithm for optimal control (Pantoja (1988)). Hence it seems worthwhile to address, yet again, the question of providing a fall-back option, within a Newton-like minimization algorithm, for dealing with the situation when the Hessian matrix is not positive definite. Some strategies that have already been used are listed below. Fuller discussion of these techniques (and others) for dealing with nonconvex and/or highly non-quadratic problems is given by Fletcher(1980), Gill et al(1981) and Conn et al(2000).

(i) The trust region approach uses a search direction that can be calculated from

$$(\lambda I + G)p = -g$$

where λ is sufficiently large to ensure that $(\lambda I + G)$ is positive definite. It is well known that this gives p as the solution of

$$\text{Minimize} \;\; p^T Gp/2 + g^T p \quad \text{s.t.} \;\; p^T p \leq \Delta \tag{1.1}$$

where Δ is a trust region radius (that depends nonlinearly on λ). In most trust region method implementations, two or three trial values of λ must be used to get an acceptable approximation to a solution of (1.1). More details are given by, for instance, Moré (1983), Dennis et al (1991).

(ii) The Truncated Newton approach (Dembo & Steihaug (1983)) determines its search direction from a *partial* solution of $Gp = -g$ using the conjugate gradient

algorithm. The conjugate gradient iterations are terminated before the accurate Newton direction has been found if G is detected as having a direction of negative curvature. Truncated Newton algorithms which include some trust-region features are given by Steihaug (1983) and Dixon & Price (1986).

(iii) Modified Cholesky factorization is a technique which, if G is not positive-definite, obtains factors L, L^T corresponding to a perturbed matrix $\tilde{G} = G + E$ which is positive definite. The search direction is then obtained as

$$p = -L^{-T}L^{-1}g = -\tilde{G}^{-1}g.$$

More details of such techniques are given by Gill, Murray & Wright (1981) or Schnabel & Eskow (1988).

(iv) An unsuitable Newton vector can be replaced by a composite search direction

$$s = \alpha q + \beta p \tag{1.2}$$

where q is parallel to the steepest descent vector $-g$ and the coefficients α and β are found by considering the behaviour of f in the plane defined by p and q. One of the first proposals of this kind was the so-called “dog-leg” algorithm described by Powell (1970) where the choice of α and β is based on satisfying a trust-region limit on the size of $||s||$. Powell suggested (1.2) in the context of a quasi-Newton approach involving $p = -H^{-1}g$ where H is only an updated estimate of G; but his method of calculating α and β could also be used if p were the true Newton direction. The idea that we describe in this paper differs from Powell’s in a number of ways – not least in that it does not default simply to a steepest descent step when $g^T Gg < 0$.

The algorithm of Dennis & Mei(1979) uses a search direction like (1.2) when minimizing a convex function – although in this method p is obtained as a quasi-Newton direction. In the nonconvex case, however, the algorithm includes a third component in the determination of s (and hence it is known as the “double dog-leg” technique).

Use of a composite search direction like (1.2) has also been proposed by Byrd et al (1987) and by Zhang & Xu (1999). These proposals differ from the algorithms described in this paper by virtue of sometimes taking p as a direction of negative curvature, rather than the Newton step, when G is indefinite. Furthermore they are developed only as trust-region algorithms, whereas we shall also consider using (1.2) in a line search context.

A search direction like (1.2) has been used for for nonlinear least-squares problems (Bartholomew-Biggs & Forbes (2000)). In this case α and β are found by fitting a quadratic model to the objective function in the plane of the Gauss-Newton and steepest descent vectors. This composite search direction is used when the standard Gauss-Newton direction proves ineffective. Experimental results have shown that this can work well; and therefore the present paper considers an extension of a similar idea for general function minimization. The situation here is a little different from the least-squares case because the Gauss-Newton step is always downhill (except in the singular case) whereas the Newton iteration may give an uphill direction if the Hessian matrix is indefinite.

As a final note on the use of composite search directions, we mention briefly a suggestion made by Scolnik(1999) which involves combining search directions generated during the inner iterations of a Truncated Newton method – i.e., a combination of ideas from *(ii)* and *(iv)* of this section.

In the next section we shall show that the determination of suitable values of α and β can be viewed as a two variable trust region problem. In section three we outline some algorithms using (1.2) and in section four we consider some questions of convergence. A brief discussion of numerical experience and future work appears in section five.

2 The Two Dimensional search direction

We wish to determine a search direction of the form

$$s = \alpha q + \beta p \tag{2.1}$$

where p is the Newton correction obtained by solving $Gp = -g$ and q is a scaled steepest descent direction. Two possible choices are

$$q = -(\frac{g^T g}{|g^T G g|})g. \tag{2.2}$$

and

$$q = -\frac{||p||}{||g||}g \tag{2.3}$$

Scaling (2.2) implies that, when $g^T G g > 0$, a unit step along q will give a good approximation to the one-dimensional minimum. Choice (2.3) gives $||q|| = ||p||$.

The Newton step p locates a stationary point of the local quadratic model of $f(x)$

$$Q(p) = p^T g + p^T G p/2 \approx f(x+p) - f(x).$$

When G is non-singular but not positive definite, p will either be uphill towards a maximum of Q or else it will be a direction (of ascent or descent) towards a saddle point. Hence the composite direction (2.1) must be constructed in order to have more desirable properties for use in a minimization algorithm.

We consider a two-dimensional quadratic model of f in the plane defined by p and q, namely

$$\phi(\alpha,\beta) = c_1\alpha + c_2\beta + c_3\alpha\beta + c_4\alpha^2/2 + c_5\beta^2/2 \tag{2.4}$$

It is easy to see, from a Taylor series expansion of $Q(x+\alpha q+\beta p)$, that

$$c_1 = q^T g;\;\; c_2 = p^T g;\;\; c_3 = p^T Gq;\;\; c_4 = q^T Gq;\;\; c_5 = p^T Gp$$

where the definition of p implies

$$c_5 = -c_2 \;\text{ and }\; c_3 = -g^T q = -c_1. \tag{2.5}$$

With the scaling (2.2) we have

$$c_1 = -\frac{(g^T g)^2}{|g^T Gg|} \;\text{ and }\; c_4 = (\frac{(g^T g)^2}{|g^T Gg|^2})g^T Gg$$

which implies

$$c_4 = \begin{cases} -c_1 & \text{if } g^T Gg > 0 \\ c_1 & \text{if } g^T Gg < 0 \end{cases} \tag{2.6}$$

With the scaling (2.3) we get

$$c_1 = -\frac{||p||}{||g||} g^T g = -||p||||g|| \;\text{ and }\; c_4 = \frac{p^T p}{g^T g} g^T Gg$$

Since we want to deal with the nonconvex case, we cannot assume that $\phi(\alpha,\beta)$ has an unconstrained minimum with respect to α and β. Hence our method for calculating α, β is based on solving

$$\text{Minimize} \quad \phi(\alpha,\ \beta) \quad \text{subject to} \quad \alpha^2+\beta^2 = \rho^2. \tag{2.7}$$

The constraint in (2.7) means that ρ acts as a kind of trust-region radius in (p,q)-space. The choice of ρ will be considered later; but we observe that when $\rho = 1$ the trust region boundary is a curve which includes the Newton point ($\alpha = 0,\ \beta = 1$). In addition, if $g^T Gg > 0$ and scaling (2.2) is used then the trust-region boundary also includes the Cauchy point ($\alpha = 1,\ \beta = 0$).

2.1 Analytical solution of (2.7)

Values of α and β which solve (2.7) must satisfy

$$c_1 + c_4\alpha + c_3\beta + \alpha\lambda = 0$$

$$c_2 + c_3\alpha + c_5\beta + \beta\lambda = 0$$

$$\alpha^2 + \beta^2 = \rho^2.$$

The first two equations are stationarity conditions for the Lagrangian

$$L(\alpha,\ \beta) = \phi(\alpha,\ \beta) - \frac{\lambda}{2}(\rho^2 - \alpha^2 - \beta^2)$$

where λ is the Lagrange multiplier for the constraint in (2.7). Using (2.5) we get

$$(\lambda + c_4)\alpha - c_1\beta = -c_1 \tag{2.8}$$

$$-c_1\alpha + (\lambda - c_2)\beta = -c_2. \tag{2.9}$$

Hence the solution to (2.7) is

$$\alpha = \frac{-c_1\lambda}{(\lambda - c_2)(\lambda + c_4) - c_1^2}; \quad \beta = \frac{-c_1^2 - c_2(\lambda + c_4)}{(\lambda - c_2)(\lambda + c_4) - c_1^2} \tag{2.10}$$

where the constraint in (2.7) means λ must satisfy

$$\frac{c_1^2\lambda^2 + (c_1^2 + c_2(\lambda + c_4))^2}{((\lambda - c_2)(\lambda + c_4) - c_1^2)^2} = \rho^2. \tag{2.11}$$

It is easy to see that (2.11) re-arranges into a quartic equation in λ.

A consequence of (2.11) is that (2.10) can be re-written to show the relation between α, β and ρ. Since (2.11) implies

$$(\lambda - c_2)(\lambda + c_4) - c_1^2 = \frac{\pm\sqrt{(}c_1^2\lambda^2 + (c_1^2 + c_2(\lambda + c_4))^2)}{\rho}$$

we get

$$\alpha = \frac{\pm\rho c_1\lambda}{\sqrt{(}c_1^2\lambda^2 + (c_1^2 + c_2(\lambda + c_4))^2)}; \quad \beta = \frac{\pm\rho(c_1^2 + c_2(\lambda + c_4))}{\sqrt{(}c_1^2\lambda^2 + (c_1^2 + c_2(\lambda + c_4))^2)} \tag{2.12}$$

It is worth noting that, if $\lambda = 0$ then (2.12) gives

$$\alpha = 0; \quad \beta = \pm\rho.$$

Since (2.11) may have several roots, we need to consider the value of λ more carefully. The stationarity of the Lagrangian for (2.7) means that

$$\partial\phi/\partial\alpha = -\lambda\alpha; \quad \partial\phi/\partial\beta = -\lambda\beta.$$

Hence the directional derivative along the outward unit normal to the trust region constraint is

$$(\alpha/\rho, \ \beta/\rho)\begin{pmatrix} -\lambda\alpha \\ -\lambda\beta \end{pmatrix} = -\lambda\rho.$$

This implies that (to the first order) a change ε in the value of ρ in (2.7) will cause the solution value of φ to change by $-\lambda\varepsilon$. Now, in the nonconvex case (i.e. when $p^Tg > 0, \ p^TGp < 0$ and/or $g^TGg < 0$) it is clear that the optimal value of φ must decrease as the radius ρ increases. Hence we must have $\lambda > 0$ for all ρ. If, however, the local quadratic model is convex ($pTg < 0$, $p^TGp > 0$ and $g^TGg > 0$) then φ has an unconstrained minimum on the trust region boundary at the Newton point when

$\alpha = 0$ and $\beta = \rho = 1$. In this case, $\lambda = 0$ (and setting $\rho > 1$ will cause an increase in the optimal ϕ). However, if $\rho < 1$ and if scaling (2.2) is used then

$$\phi(\rho, 0) > \phi(1, 0) \quad \text{and} \quad \phi(0, \rho) > \phi(0, 1)$$

and so, as in the nonconvex case, the optimal value of ϕ will decrease as ρ increases. Summarising these observations we can say

$$\lambda > 0 \quad \text{for all } \rho > 0 \text{ when } \phi \text{ is nonconvex} \tag{2.13}$$

$$\lambda \geq 0 \quad \text{for } 0 < \rho \leq 1 \text{ when } \phi \text{ is convex} \tag{2.14}$$

We can place another restriction on the value of λ, using the second order optimality condition for (2.7), namely

$$z^T \nabla^2 L z > 0$$

where z is the tangential vector $(\beta,\ -\alpha)^T$ at the solution. Since $\nabla^2 L$ is the coefficient matrix in (2.8), (2.9), it follows that

$$(\lambda + c_4)\beta^2 - 2c_1\alpha\beta + (\lambda - c_2)\alpha^2 > 0$$

which leads to

$$\lambda > \frac{c_2\alpha^2 + 2c_1\alpha\beta - c_4\beta^2}{\rho^2}. \tag{2.15}$$

2.2 A variable transformation approach to (2.7)

By contrast with the above discussion, we can approach (2.7) in another, possibly simpler, way by defining

$$\alpha = \rho \sin\theta; \quad \beta = \rho \cos\theta \tag{2.16}$$

This ensures that the trust region constraint is automatically satisfied and therefore we wish to find θ^* as the unconstrained minimizer of

$$\psi(\theta) = \rho(c_1 \sin\theta + c_2 \cos\theta) + \frac{\rho^2}{2}(2c_3 \sin\theta\cos\theta + c_4 \sin^2\theta + c_5 \cos^2\theta)$$

which is obtained by substituting (2.16) in (2.4). Using (2.5) this simplifies to

$$\psi(\theta) = \rho(c_1 \sin\theta + c_2 \cos\theta) + \frac{\rho^2}{2}(-c_1 \sin 2\theta + c_4 \sin^2\theta - c_2 \cos^2\theta). \tag{2.17}$$

It then follows, on differentiating (2.17) with respect to θ, that θ^* satisfies

$$\rho(c_1 \cos\theta^* - c_2 \sin\theta^*) + \frac{\rho^2}{2}(-2c_1 \cos 2\theta^* + (c_2 + c_4) \sin 2\theta^*) = 0 \tag{2.18}$$

and

$$-\rho(c_1 \sin\theta^* + c_2 \cos\theta^*) + \frac{\rho^2}{2}(4c_1 \sin 2\theta^* + 2(c_2 + c_4)\cos 2\theta^*) > 0 \qquad (2.19)$$

In this paper we shall not pursue the question of getting λ, α and β by finding an appropriate solution of (2.11) and substituting in (2.12); and neither shall we consider the use of (2.18) and (2.19) to find θ^*. Instead, in the algorithms described in the next section, we shall adopt the simpler expedient of determining θ^* iteratively by applying the bisection technique to (2.17).

To conclude this section, we comment briefly on the relationship between our proposal using (2.7) and (2.17) and the method of Powell (1970). This is most easily done if we consider the case when q is defined using the scaling (2.3). If Δ denotes a trust-region radius on the step s then we simply set $\rho = \Delta/||p||$ in the subproblem (2.7). If, however, the same value of Δ were to be imposed within the context of the Powell "dogleg" algorithm then α, β would be found as follows.
If $g^T Gg \leq ||g||^3/\Delta$ then

$$\beta = 0, \;\; q = -g/||g||, \;\; \alpha = \Delta$$

otherwise

$$q = -g\frac{g^T g}{g^T Gg}, \quad \alpha = 1 - \beta$$

where β solves

$$\text{Minimize} \quad \frac{\beta^2 (p-q)^T G(p-q)}{2} + \beta g^T (p-q)$$

$$\text{subject to } \sqrt{(}q^T q + \beta^2 (p-q)^T (p-q)) \leq \Delta$$

The first point to observe is that a pure steepest descent step is used whenever $g^T Gg$ is negative, which seems to be a much weaker strategy than the one we propose. In the case where the Powell method does use a composite step, it is based upon searching the line between the Cauchy and Newton points for the least value of the local quadratic model (subject to being inside the trust region).

2.3 A worked example using search direction (2.1)

As an illustration of the behaviour of the optimally chosen search direction (2.1) obtained by minimizing (2.17) we consider the simple function

$$f(x) = x_1 x_2 + c(x)^2 \;\text{ where }\; c(x) = \text{Min}(0,\; 1 - x_1^2 - x_2^2) \qquad (2.20)$$

at the point $x = (-0.5,\ 0.25)$. Here $c(x) = 0$ and so $f(x) = -0.125$. Moreover,

$$g = \begin{pmatrix} 0.25 \\ -0.5 \end{pmatrix} \;\text{ and }\; G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The Newton step $p = (0.5,\ -0.25)^T$ is *uphill* towards the saddle point at the origin. Using (2.2) we get $q = (-0.3125,\ 0.625)^T$. After substituting for $c_1,..,c_5$ and taking $\rho = 1$ in (2.17) we find, on applying the bisection method in the initial range $0 \leq \theta \leq \pi$, that $\theta^* \approx 2.221$ and $\psi(\theta^*) \approx -0.82$. Thus the step s (2.1) is

$$s \approx 0.796q - 0.605p \approx (-0.5513,\ 0.6489)^T.$$

Hence the new point $x+s \approx (-1.0513,\ 0.8989)^T$ which gives $c(x+s) \approx -0.9132$ and $f(x+s) \approx -0.1111$. This does not agree with the quadratic prediction

$$f(x+s) = f(x) + \psi(\theta^*)$$

and, moreover, $f(x+s) > f(x)$. Hence we need to consider a smaller step. If we set $\rho = 0.5$ in (2.17) then ψ is minimised at $\theta^* \approx 2.198$ which leads to

$$s \approx 0.405q - 0.294p \approx (-0.2733,\ 0.3263)^T.$$

Now $x+s = (-0.7733,\ 0.5763)^T$ and $f(x+s) = -0.4457$. Hence we have obtained an improved point by a reduced step along the composite direction (2.1).

Next we consider the function (2.20) at $x = (0.5,\ 0.25)^T$, where $f(x) = 0.125$. Now $p = (-0.5,\ -0.25)^T$ and the scaling (2.2) yields $q = (-0.3125,\ -0.625)^T$. When $\rho = 1$ the quadratic model (2.17) has a minimum value of $\psi \approx -0.2205$ when $\theta^* \approx 1.883$. Thus the combined step (2.1) is

$$s \approx 0.952q - 0.307p \approx (-0.1437,\ -0.5179)^T.$$

Hence $x+s = (0.3563,\ -0.2679)^T$ and $f(x+s) = -0.0955$. It is important to note that – unlike the pure Newton correction – the combined step s achieves descent *past* the the saddle point at $x_1 = x_2 = 0$. Furthermore

$$f(x) + \psi(\theta^*) = 0.125 - 0.2205 = -0.0955 = f(x+s)$$

and so the actual change in f agrees with the quadratic prediction. We can therefore consider taking a larger step; and if we set $\rho = 1.5$ in (2.17) we find that ψ is minimized when $\theta^* \approx 2.07$ which gives

$$s \approx 1.32q - 0.717p \approx (-0.0529,\ -0.6439)^T$$

so that $x+s = (0.4471,\ -0.3939)^T$ and $f(x+s) = -0.1761$. Hence the increase in trust region radius produces a bigger decrease in f.

3 Algorithmic considerations

We now consider how the search direction (2.1) can be brought into the framework of a Newton-like method for minimization. We shall in fact propose two approaches: one which uses line searches and one which is of a trust-region type. First, however, we mention some other algorithmic details.

3.1 Scaling the steepest descent direction

We have mentioned two possible scalings for defining q in (2.1). Both these scalings will be useful and in order to implement the ideas outlined in the previous section we need to make a choice between them. This can be done by the following simple algorithm, involving a small positive constant m.

$$\text{If } |g^T Gg| \geq mg^T g \text{ then } q = -(\frac{g^T g}{|g^T Gg|})g; \text{ else } q = -\frac{||p||}{||g||}g. \tag{3.1}$$

Use of (3.1) will mean that a unit step along q will not overshoot the one-dimensional minimum (if there is one). Hence the discussion preceding (2.14) about the positivity of the Lagrange multiplier for (2.7) will still be valid.

3.2 Minimizing (2.17)

We consider next the minimization of the local quadratic model (2.17) with respect to θ. In order to use the bisection method we need to identify the range in which we expect θ^* to lie. If we evaluate

$$\psi_i = \psi(i\frac{\pi}{2}) \quad \text{for } i = 0,3$$

then $\psi_0, ..., \psi_3$ will, respectively be quadratic predictions of

$$f(x+p), \;\; f(x+q), \;\; f(x-p), \;\; f(x-q).$$

Now if $\psi_k = \text{Min}(\psi_0, ..., \psi_3)$ then the optimal value θ^* lies in the range

$$(k-1)\frac{\pi}{2} \leq \theta^* \leq (k+1)\frac{\pi}{2}$$

which gives a suitable starting bracket for the bisection method.

3.3 Solving the Newton equation

The search direction (2.1) involves the calculation of $p = -G^{-1}g$ even when G is indefinite. If, as is usual in Newton methods, we attempt the solution of $Gp = -g$ by the Cholesky method then we shall easily detect indefiniteness of G by a breakdown in the factorization. It will then, however, be necessary to start the solution of the Newton equation again by a different method (such as standard LU factorization). A better approach might be to compute the LBL^T factors of G (where B is a matrix with 1×1 or 2×2 blocks on the diagonal). We can then obtain

$$p = -L^{-T}B^{-1}L^{-1}g \tag{3.2}$$

using forward and backward substitution. Non positive-definiteness of G can be detected by the presence of negative terms or negative eigenvalues in the 2×2 blocks in the matrix B.

We must also consider the special case when G is singular because $Gp = -g$ is then not solvable by any standard technique. For dealing with this situation we suggest the use of a modified LBL^T factorization in which any diagonal element which is zero (to within some tolerance based on the working precision) is replaced by a prescribed small positive constant and any block with a zero eigenvalue is replaced by one whose smallest eigenvalue is the same threshold constant. The resulting factors will then correspond to some positive definite $\tilde{G} = G + E$, where E is a correction matrix. The vector p given by (3.2) can be regarded as an "almost-Newton" direction which can be used in the calculation of (2.1).

3.4 A linesearch algorithm using (2.1)

We can now present a typical iteration of an algorithm which performs a line search along the Newton direction whenever G is positive definite. When G is indefinite or singular, however, it calculates the composite direction s and performs a line search along this instead.

Algorithm 1

```
Perform a (possibly modified) $LBL^T$ factorization of $G$
Obtain $p$ from (3.2)
if $G$ is positive definite then
    obtain a new point $x^+ = x + \gamma p$ to satisfy the Wolfe conditions
else
    Calculate $q$ from (3.1)
    Calculate the coefficients $c_1, .., c_5$ appearing in (2.17)
    set $\rho = 1$ in (2.17)
    find $\theta^*$ to minimize $\psi(\theta)$ given by (2.17)
    set $s = \sin\theta^* q + \cos\theta^* p$
    obtain a new point $x^+ = x + \gamma s$ to satisfy the Wolfe conditions
end if
```

3.5 A trust region algorithm using (2.1)

An alternative approach offers a more radical departure from a standard Newton iteration. It replaces the line search with a 2D trust region strategy *even when G is positive definite*. The iteration described below features a step-size limit Δ which is revised on the basis of progress made along the previous search direction. (Δ is set equal to $||p||$ at the start of the first iteration.) In what follows, η_1, τ_1 and τ_2 are small positive constants used in comparing the behaviour of f with its local quadratic model; and $k_1(>1), k_2(<1)$ are scaling factors involved in the adjustment of Δ.

Algorithm 2

```
Find $p$ by (modified) $LBL^T$ factorization of $G$
Calculate $q$ and $c_1,..,c_5$ appearing in (2.17)
If $G$ is positive definite
    set $s = p$, $\theta^* = 0$, $\rho = 1$
    if $f(x+s) - f(x) \leq \eta_1\psi(\theta^*)$ then set $x^+ = x+s$ and exit
end if
if $G$ is not positive definite or $f(x+s) - f(x) > \eta_1\psi(\theta^*)$
    set $\rho = \min(1,\ \Delta/||p||)$
    Repeat
        find $\theta^*$ to minimize $\psi(\theta)$
        set $s = \rho\sin\theta^* q + \rho\cos\theta^* p$
        set $\rho = \rho/2$
    until $f(x+s) - f(x) \leq \eta_1\psi(\theta^*)$
    set $x^+ = x+s$ and exit
end if
compute $\sigma = (f(x+s) - f(x))/\psi(\theta^*)$
```

$$\Delta = \begin{cases} k_1||s|| & \text{if } 1-\tau_1 \leq \sigma \leq 1+\tau_1 \\ k_2||s|| & \text{if } \sigma \leq \tau_2 \\ ||s|| & \text{otherwise} \end{cases}$$

4 Computational experience

We shall now compare Algorithms 1 and 2 from Section 3 with two other Newton-type methods. In particular we consider a Truncated Newton method (**optnhp**, Dixon & Price (1986)) and an implementation of the **impbot** algorithm (Brown & Bartholomew-Biggs (1985)) which can be viewed as a trust region approach.

The subroutine **optnhp** uses conjugate gradient iterations to obtain an approximate solution to the Newton equation, stopping when the norm of the residual vector $||Gp+g||$ is less than some tolerance which is initially quite large and decreases on every iteration. The "inner" conjugate gradient iterations are also terminated if G is found to be non-positive definite or if $||p||$ exceeds a limit (which is adjusted at the end of each "outer" iteration).

The subroutine **impbot** is based on following the solution trajectory of the continuous steepest descent equation

$$\frac{dx}{dt} = -g(x)$$

and the step on each iteration is found by solving

$$(hG+I)p = -hg.$$

This method of calculating p comes from the application of the implicit Euler method with h as the step size in terms of the parameter t. Adjustment of this step size is based on the progress on each iteration, in broadly the same way as in Algorithm 2. We can see that the calculation of p in **impbot** is essentially the same form as the calculation of p in a trust-region method, but with $\lambda = h^{-1}$.

We consider the following test problems.

Problem 1 is a generalisation of the example used in section 2.

$$\text{Minimize} \quad x^T Gx + c(x)^2 \;\; \text{where} \;\; c(x) = \text{Min}(0, n-1-\sum_{i=1}^{n} x_i^2)$$

where G is the $n \times n$ matrix with $g_{ii} = 0,\ g_{ij} = 1$ when $i \neq j$. The starting point is $x_1 = 0.5, x_2 = 0.25, x_3 = ... = x_n = 0$.

Problem 2 involves the extended Rosenbrock function

$$\text{Minimize} \quad \sum_{i=1}^{n-1} 100(x_{i+1}^2 - x_i)^2 + (1-x_i)^2$$

The starting point is the non-standard one $(0,2,0,2,..)^T$.

Problem 3 is another penalty function-like example

$$\text{Minimize} \quad x^T Ax/2 + b^T x + c(x)^2 \;\; \text{where} \;\; c(x) = \text{Min}\ [0, n-1-\sum_{i=1}^{n} x_i^2]$$

where the elements of A and b are given by

$$a_{ij} = 1.0 \;\; \text{when} \;\; i \neq j; \;\; a_{ii} = 0.9^{i-1}; \quad b_i = 0.1$$

The starting point is taken as $x_i = 1/n$ for $i = 1, ..., n$.

Problem 4 is a barrier-function-like example

$$\text{Minimize} \quad x^T Ax/2 + b^T x + 0.001c(x)^{-1} \;\; \text{where} \;\; c(x) = 1 - \sum_{i=1}^{n} x_i^2.$$

The elements of A and b and the starting point are the same as for Problem 3.

Problem 5 is the extended Wood function

$$\text{Minimize} \quad \sum_{i=1}^{n-3} [100(x_{i+1} - x_i^2)^2 + (1-x_i)^2 + 90(x_{i+3} - x_{i+2}^2)^2$$

$$+10.1\{(x_{i+1}-1)^2+(x_{i+1}-1)^2\}+19.8(x_{i+1}-1)(x_{i+3}-1)]$$

The starting point is $x=(-3,-1,-1,-1,...)^T$

The following table shows the performance on these five test examples for various values of n. For all the methods we quote numbers of iterations and function evaluations needed for convergence. In addition, for the new algorithms, we show in brackets the number of iterations on which a non positive definite Hessian was encountered.

Problem	Alg 1		Alg 2		optnhp		impbot	
	itns	fns	itns	fns	itns	fns	itns	fns
1 n=2	5(2)	6	5(2)	6	6	44	9	14
1 n=4	6(3)	11	5(3)	6	6	45	10	16
1 n=8	7(4)	14	6(5)	11	7	79	11	20
2 n=2	14(1)	18	13(1)	18	16	27	16	22
2 n=12	37(7)	50	34(7)	45	20	73	37	53
2 n=24	59(13)	78	36(26)	42	19	34	62	87
3 n=5	27(9)	42	31(16)	48	18	98	37	53
3 n=10	33(11)	52	36(12)	52	27	118	41	66
3 n=20	59(12)	91	53(10)	80	49	201	53	79
4 n=15	46(11)	61	45(10)	52	37	101	42	54
4 n=20	50(10)	64	46(8)	53	46	198	45	55
4 n=25	59(10)	78	61(10)	75	63	454	55	70
5 n=4	40(3)	57	38(2)	53	***		39	63
5 n=12	17(1)	29	14(2)	17	17	33	18	34
5 n=20	19(1)	32	18(6)	21	18	172	19	36

*** Method fails near a saddle point

These preliminary tests suggest that the new algorithms using the composite search directions are competitive with existing techniques. We see from the bracketed figures in the first two columns that the direction (2.1) is used on a significant number of iterations. The new algorithms usually match or exceed the performance of the better of **optnhp** and **impbot**; and this remains true for the larger values of n even though the composite step can only explore a two-dimensional subspace. On the whole, the trust region strategy in Algorithm 2 seems slightly more effective than the linesearch-based Algorithm 1.

Generally speaking, the Truncated Newton method is very much less effcient in terms of function evaluations — presumably because it frequently uses an approximation to the Newton direction along which a unit step fails to produce an acceptable improvement. In terms of numbers of iterations, however, it often does better than the other methods; and this is important because it implies that fewer evaluations of the Hessian matrix will be needed. The performance of the two new algorithms is often quite similar to that of **impbot**. However, it should be remembered that **impbot** – unlike Algorithm 1 or Algorithm 2 – may have to solve more than one $n \times n$ linear system on each iteration in order to control the step-size.

5 Convergence issues

An intuitive justification of the algorithms has been outlined in section 3. When they are applied sufficiently close to a strong local minimum of $f(x)$ then they should both take pure Newton steps and have ultimately quadratic convergence. Global convergence properties should follow fairly easily from the fact that, to second order accuracy, the step (2.1) obtained by solving (2.7) is at least as effective as a steepest descent step; and results about global convergence of steepest descent methods are well-known.

We shall now put some flesh on the bones of the previous remarks, in the context of Algorithm 2. In order to do this we need to make an assumption about the non-singularity of G.

Assumption 1 G is a nonsingular matrix whose eigenvalues are bounded away from zero - i.e. there exist positive real numbers m and M such that any eigenvalue, λ, of G satisfies either

$$M > \lambda > m \quad \text{or} \quad -m > \lambda > -M.$$

Notice that, without loss of generality, we shall assume that m is the same small positive constant that appears in (3.1).

Lemma 1 Under Assumption 1 it follows that

$$g^T g/m^2 > p^T p > g^T g/M^2; \tag{5.1}$$

and if q is defined by (3.1)

$$g^T g/m^2 > q^T q > g^T g/M^2 \tag{5.2}$$

Proof
The first result (5.1) follows from the definition of p since $p^T p = g^T G^{-2} g$ and all the eigenvalues of G^2 must lie in the range $[m^2,\ M^2]$.

If $|g^T Gg| \geq mg^T g$ then q is given by (2.2) and so

$$q^T q = (g^T g)^3/(g^T Gg)^2$$

from which (5.2) follows because $M^2(g^T g)^2 > (g^T Gg)^2 > m^2(g^T g)^2$.

If $|g^T Gg| < mg^T g$ so that q is defined by (2.3) then

$$q^T q = \frac{p^T p}{g^T g} g^T g = p^T p$$

and then (5.2) follows directly from (5.1).

Lemma 2 Assumption 1 and scaling (3.1) imply that

$$g^T g/m^2 > p^T q > -g^T g/m^2 \tag{5.3}$$

$$g^T g/m > p^T g > -g^T g/m \tag{5.4}$$

and

$$-g^T g/M > q^T g > -g^T g/m \tag{5.5}$$

Proof
Since

$$p^T q = \sqrt{(}p^T p q^T q)\cos\xi$$

where ξ is the angle between p and q then

$$\sqrt{(}p^T p q^T q) \geq p^T q \geq -\sqrt{(}p^T p q^T q)$$

and (5.3) follows from the bounds in Lemma 1.

Similarly we can write

$$\sqrt{(}p^T p g^T g) \geq p^T g \geq -\sqrt{(}p^T p g^T g)$$

and (5.4) follows from the upper bound on $p^T p$ in Lemma 1.

If $|g^T G g| \geq m$ then q is defined by (2.2) and

$$q^T g = -(g^T g)^2/|g^T G g|.$$

But Assumption 1 means that $Mg^t g > |g^T G g|$ and so we get (5.5).
If, on the other hand, q is given by (2.3) then

$$q^T g = -\sqrt{(}p^T p g^T g)$$

and now the bounds on $p^T p$ from Lemma 1 give (5.5).

Lemma 3 Assumption 1 and the scaling (3.1) imply that the two-norm of search direction s is bounded by

$$||s|| < \sqrt{2}\rho\sqrt{(}g^T g)/m \tag{5.6}$$

Proof
Since $s = \rho\sin\theta q + \rho\cos\theta p$ we have

$$s^T s = \rho^2(\sin^2\theta q^T q + \cos^2\theta p^T p + \sin 2\theta p^T q).$$

Hence, using Lemmas 1 and 2,

$$s^T s < \rho^2(\sin^2\theta + \cos^2\theta + \sin 2\theta) g^T g/m^2$$

whose right hand side takes a maximum value of 2 when $\theta = \pi/4$, leading to (5.6).

We now consider the value the local quadratic model function ψ which gives the predicted change $f(x+s) - f(x)$ in the objective function. We shall use ψ^* to denote $\psi(\theta^*)$ where θ^* is the value that minimizes (2.17).

Lemma 4 Under Assumption 1 the optimal value of ψ satisfies the bound

$$\psi^* < -Cg^Tg/M \tag{5.7}$$

where

$$C = (\rho - \frac{\rho^2}{2}) \text{ when } g^Tg > 0; \tag{5.8}$$

$$C = (\rho + \frac{\rho^2}{2}) \text{ when } g^TGg \leq -mg^Tg; \tag{5.9}$$

$$C = \rho \text{ when } 0 \geq g^TGg > -mg^Tg. \tag{5.10}$$

Proof
We note first that, since ψ^* is the minimum value of (2.17),

$$\psi^* \leq \psi(\frac{\pi}{2}) = \rho c_1 + \frac{\rho^2}{2}c_4 \tag{5.11}$$

We also observe that the coefficient c_1 in ψ is bounded, because $c_1 = q^Tg$, and hence, by Lemma 2,

$$-g^Tg/m < c_1 < -g^Tg/M. \tag{5.12}$$

We now need to consider several distinct cases.

Case 1: $g^TGg \geq mg^Tg$

In this case q is defined by (2.2). Thus, by (2.6), $c_4 = -c_1$ and (5.11) becomes

$$\psi^* \leq \psi(\frac{\pi}{2}) = (\rho - \frac{\rho^2}{2})c_1.$$

Hence, using (5.12),

$$\psi^* < -(\rho - \frac{\rho^2}{2})g^Tg/M \tag{5.13}$$

Case 2: $g^TGg \leq -mg^Tg$

As in case 1, q is given by (2.2), and so (2.6) gives $c_4 = c_1$. Hence

$$\psi(\frac{\pi}{2}) = (\rho + \frac{\rho^2}{2})c_1;$$

and, using (5.11) and (5.12),

$$\psi^* < -(\rho + \frac{\rho^2}{2})g^Tg/M \tag{5.14}$$

Case 3: $mg^Tg > g^TGg > -mg^Tg$

In this case we obtain q from (2.3) which gives

$$c_1 = q^Tg = -\sqrt{(}q^Tqg^Tg) \text{ and } c_4 = q^TGq = \frac{q^Tq}{g^Tg}g^TGg.$$

Hence

$$c_4/c_1 = -\frac{\sqrt{(q^T q)}}{g^T g\sqrt{(g^T g)}} g^T G g \tag{5.15}$$

Now if $mg^T g > g^T G g > 0$, (5.15) implies

$$0 > c_4/c_1 > -1$$

because of Lemma 1. But $c_1 < 0$ and hence $c_4 < -c_1$. Thus

$$\psi(\frac{\pi}{2}) = \rho c_1 + \frac{\rho^2}{2} c_4 < \rho c_1 - \frac{\rho^2}{2} c_1.$$

Using (5.11) and (5.12), this means ψ^* is bounded by

$$\psi^* < -(\rho - \frac{\rho^2}{2})\frac{g^T g}{M} \tag{5.16}$$

On the other hand, if $0 > g^T G g > -mg^T g$ then (5.15) gives

$$0 < c_4/c_1 < 1;$$

and because $c_1 < 0$ it follows that $c_4 < 0$. Therefore

$$\psi(\frac{\pi}{2}) = \rho c_1 + \frac{\rho^2}{2} c_4 < \rho c_1;$$

and, using (5.11), (5.12),

$$\psi^* < -\rho\frac{g^T g}{M} \tag{5.17}$$

Hence the Lemma is proved, with (5.8) obtained by combining (5.13) and (5.16) and (5.9) and (5.10) coming from (5.14) and (5.17), respectively.

It is clear that the bound on ψ^* given by (5.7) and (5.8) is negative only if $\rho < 2$. Hence, from now on, we shall follow Algorithm 2 and restrict allowable values of ρ to the range $0 < \rho \leq 1$.

It is also clear that, for any particular value of ρ, (5.7) and (5.8) give the most pessimistic bound on ψ^* and hence on the predicted reduction in the objective function. For the purpose of the subsequent analysis it will be sufficient for us to use this bound only.

Theorem 1 Under Assumption 1 and for a given η_1 $(0 < \eta_1 < 1)$ there exists a constant $\bar{\rho}$, depending only on M, m and η_1, such that the actual reduction in f caused by the step s satisfies

$$f(x+s) - f(x) < \eta_1 \psi^* \tag{5.18}$$

for all ρ satisfying $0 < \rho < \bar{\rho}$.

Proof

By the mean value theorem

$$f(x+s) - f(x) = f(x) + s^T g(x) + \frac{s^T G(x+\omega s)s}{2}$$

for some ω between 0 and 1. This can be rewritten as

$$f(x+s) - f(x) = f(x) + s^T g(x) + \frac{s^T G(x)s}{2} + \frac{s^T G(x+\omega s)s - s^T G(x)s}{2}.$$

By the definition of ψ^* it follows that

$$f(x+s) - f(x) = \psi^* + \frac{s^T G(x+\omega s)s - s^T G(x)s}{2}$$

Hence (5.18) follows if

$$\frac{s^T G(x+\omega s)s - s^T G(x)s}{2} < -(1-\eta_1)\psi^*$$

Recalling from (5.7), (5.8) that

$$-\psi^* > (\rho - \frac{\rho^2}{2})g^T g/M$$

it follows that (5.18) will hold if

$$\frac{s^T G(x+\omega s)s - s^T G(x)s}{2} < (1-\eta_1)(\rho - \frac{\rho^2}{2})g^T g/M.$$

Now, applying Assumption 1 to the left hand side of the above inequality, we get

$$\frac{s^T G(x+\omega s)s - s^T G(x)s}{2} < \frac{2Ms^T s}{2};$$

and we also have, by Lemma 3,

$$\frac{2Ms^T s}{2} < \frac{2M\rho^2 g^T g}{m^2}.$$

Hence (5.18) will be satisfied if

$$\frac{2M\rho^2 g^T g}{m^2} < (1-\eta_1)(\rho - \frac{\rho^2}{2})g^T g/M.$$

Simplifying both sides we get

$$\frac{(2M^2\rho)}{m^2} < (1-\eta_1)(1 - \frac{\rho}{2})$$

from which we see that (5.18) holds provided

$$\rho < \bar{\rho} = \frac{2(1-\eta_1)}{(4M^2/m^2+1-\eta_1)}. \tag{5.19}$$

Theorem 2 There exists a constant C such that the actual reduction produced by an iteration of algorithm 2 is bounded by

$$f(x+s) - f(x) < -Cg^T g/(2M)$$

Proof
The inner iteration of Algorithm 2 proceeds by halving ρ until condition (5.18) is satisfied. It follows that *at worst* some iterations may use the value value $\rho = \bar{\rho}/2$, where $\bar{\rho}$ is given by (5.19). In this extreme case Lemma 4 implies the bound

$$\psi^* < -(\bar{\rho} - \bar{\rho}^2/4)g^T g/(2M)$$

Combining this with (5.18) and recalling that $\bar{\rho}$ is a constant depending only on M, m and η_1, the theorem follows with

$$C = \eta_1(\bar{\rho} - \bar{\rho}^2/4)$$

We can now establish a global convergence result for Algorithm 2.
Theorem 3 Suppose that $f(x)$ is a function which is bounded below and which has a strong local minimum at x^*. Suppose also that its Hessian G satisfies Assumption 1 for all x in a neighbourhood, N, which includes x^*. Then a sequence of steps s produced by Algorithm 2 must terminate at a stationary point of $f(x)$ for any starting point in N.
Proof
If the Theorem is false then there exists a positive ε such that each iteration ends with $g^T g > \varepsilon$. But the previous Theorem then implies that f is reduced on each iteration by at least $C\varepsilon/(2M)$. For this to occur on an infinite number of iterations contradicts the condition that $f(x)$ is bounded below and hence the Theorem must be true.

We note that Theorem 3 only relates to convergence to a local stationary point of $f(x)$. We shall mention this point again in the discussion in the next section.

6 Discussion and Conclusions

We have described a method of obtaining search directions in a Newton algorithm which can be used when the Hessian matrix is not positive definite. It can also be used to provide an alternative correction when the Hessian is positive definite but the usual Newton step does not provide a suitable decrease in the objective function.

The derivation of the modified step is based on finding the minimum of a local quadratic model of the objective function in the plane of the Newton and steepest descent vectors. It extends the idea suggested for nonlinear least-squares problems by Bartholomew-Biggs & Forbes (2000) in using a two-dimensional trust region constraint to deal with general (non-convex) unconstrained minimization. The approach described in this paper differs from that suggested by Powell (1970) because it finds an accurate minimum of the local quadratic model (subject to a step size constraint). Our suggested Algorithm 1 differs from previous proposals by Byrd et al (1987) and by Xhang & Xu (1999) in being a line-search, rather than a trust-region, technique.

Some limited numerical testing has been done using prototype implementations of the two algorithms given in section 3to solve a set of small non-convex problems. Results are mildly encouraging but more development work is still needed, e.g. to clarify the relative merits of the line search and trust region approaches (although initial evidence seems to favour the latter.) A more fundamental point about the algorithm and its implementation is that, in its present form, it does not deal very effectively with singularity of the Hessian since then the vector p does not exist and s becomes a steepest descent direction. Moreover, Algorithms 1 and 2 would both terminate at the saddle point of the function

$$f(x) = x_1^2 + x_2^2 - x_3^2 + 10[\max(0, (x_3 - 1)]^2$$

when started from any point with $x_3 = 0$. This remark emphasises the limitations of the convergence discussion in section 4 and indicates that we may need sometimes to consider building s from a direction of negative curvature as suggested, for instance, by Zhang & Xu (1999).

Alongside such practical investigation, there is scope for a closer look at the convergence of the algorithms, since the analysis in section 4 has not made use of all the optimal properties of the new search direction as expressed, for instance, in (2.18) and (2.19).

Notwithstanding the reservations expressed in the preceding paragraphs, we feel that the ideas discussed in this paper do have promise and potential for development. It is worth recalling, for instance, that (2.1) could also be used in a quasi-Newton context, possibly as a way of obtaining descent directions when an update is used which does not guarantee to produce positive definite Hessian estimates. The Symmetric Rank One update is one such update whose attractive properties are often felt to be outweighed by this deficiency. One can also envisage applications in the context of constrained optimization. Many algorithms are based on solving systems of KKT equations which involve the Hessian of the Lagrangian function (or some estimate of it). From these equations a search direction p is obtained which is supposed to be a Newton-like direction with respect to some penalty function $P(x)$. In order for the KKT equations to yield a suitable search direction we need the (estimated) Lagrangian to have positive curvature in the subspace tangential to the active constraints. In the event that this is not so (e.g. because the correct set of constraints has not yet been identified) then an approach like the one described in this paper could be used to combine p with the steepest descent vector $-\nabla P$ and hence obtain an effective descent direction. Such a strategy might prove to be computationally less expensive than alternatives proposed by e.g., Forsgren & Murray (1993) or Hernandez (1995) which involve special rules for pivot selection and modification during the factorization of the KKT matrix.

7 References

Bartholomew-Biggs, M.C. and Forbes, A.B., A two-dimensional search used with a non-linear least squares solver, Journ. Opt. Theory, Applics, Vol 104, 1, January 2000

Byrd, R.H., Schnabel, R.B. and Scultz, G.A., A trust region algorithm for nonlinearly constrained optimization, SIAM Journal on Numerical Analysis, Vol 24, pp 1152 1170, 1987.

Christianson, Bruce, Automatic Hessians by Reverse Accumulation, IMA Jnl of Numerical Analysis, Vol 12, pp 135 - 150, 1992

Conn, A.R., Gould, N.I.M and Toint, P.L., “Trust Region Methods”, MPS-SIAM Series in Optimization, 2000.

Dembo, R.S., and Steihaug, T., Truncated Newton algorithms for large scale unconstrained optimization, Math. Prog. Vol 26, pp 190-212, 1983

Dennis, J.E. and Mei, H.H.W., An unconstrained optimization algorithm which uses function and gradient values, Journal of Optimization Theory and Aplications Vol 28, No 4 pp453 - 482, 1979.

Dennis, J.E., Echebest, N., Guardarucci, M.T., Martinez, J.M., Scolnik H.D. and Vacchino, C., A Curvilinear Search using Tridiagonal Secant Updates for Unconstrained Optimization, SIAM J. Optimization, Vol 1, No 3, pp 333 - 357, 1991

Dixon, L.C.W. and Price, R.C., The Truncated Newton method for Sparse Unconstrained Optimization using Automatic Differentiation, Technical Report 170, Numerical Optimisation Centre, Hatfield Polytechnic, 1986.

Fletcher, R. “Practical methods of Optimization”, Wiley, Chichester, 1980.

Forsgren, A.L. & Murray, W., Newton methods for large-scale linear equality constrained minimization, SIAM Journal on Matrix Analysis & Application, 14, 560-587, 1993

Gill, P.E., Murray, W. and Wright, M.H. “Optimization in Practice”, Academic Press, London and New York, 1981

Hernandez, M. de F. G., Algorithms for large sparse constrained optimization, PhD Thesis, University of Hertfordshire, 1995

More, J.J., Recent developments in algorithms and software for trust-region methods, *in* “Mathematical Programming, The State of the Art”, (edited by A. Bachem, M. Grotschel and G. Korte), Springer-Verlag, Berlin, 1983

Pantoja, J.F.A. De O., Differential Dynamic Programming and Newton’s method, Int. J. Control, Vol 47 No 5, pp 1539 - 1553, 1988.

Powell, M.J.D., A new algorithm for unconstrained optimization, *in* ”Nonlinear Programming” (edited by J. Rosen, O. Mangasarian and K. Ritter), Academic Press, 1970.

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

Scolnik, H.D., Combining search directions in nonlinear optimization, Presented at the 19th IFIP conference on Optimization, Cambridge, England, July 1999

Steihaug, T. The conjugate gradient method and trust regions in large-scale optimization, SIAM Journal on Numerical Analysis Vol 20, No 3, pp 626 - 637, 1983.

Zhang, J., and Xu, C., A class of indefinite dogleg path methods for unconstrained optimization, SIAM Journal on Optimization Vol 9, No 3, pp 646-676, 1999.