Preconditioned Low-order Newton Methods

by

Y.F. Hu and C. Storey

Department of Mathematical Sciences
Loughborough University of Technology
LOUGHBOROUGH
Leicestershire
LE11 3TU

Mathematics Report Number A159

MAY 1992

Abstract

In this paper low-order Newton methods are proposed that make use of previously obtained second derivative information by suitable preconditioning.

When applied to a particular two-dimensional Newton method (the "LS method"), it is shown that a member of the Broyden family of quasi-Newton methods is obtained. Algorithms based on this preconditioned LS model are tested against some variations of the BFGS method and shown to be much superior in terms of number of iterations and function evaluations, but not so effective in terms of number of gradient evaluations.

Keywords

Unconstrained optimization, preconditioning, lower-order Newton methods, quasi-Newton methods.

1. Introduction

Consider the unconstrained minimization problem

$$\min_{x \in R^n} \; f(x),$$

where f is a twice continuously differentiable function. This problem is usually solved iteratively. Starting with an initial estimate $x^{(1)}$ of the minimum point, each subsequent point $x^{(k+1)}$ $(k \geq 1)$ will be derived by searching along a descent direction $s^{(k)}$ ($s^{(k)}$ is a descent direction if ${s^{(k)}}^T \nabla f(x^{(k)}) < 0$), so that

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}, \qquad k \geq 1.$$

Here $\lambda^{(k)}$ is the step length satisfying line search conditions such as

$$f(x^{(k)} + \lambda^{(k)} s^{(k)}) \leq f(x^{(k)}) + \rho \lambda^{(k)} \nabla f(x^{(k)})^T s^{(k)} \tag{1}$$

and

$$|\nabla f(x^{(k)} + \lambda^{(k)} s^{(k)})^T s^{(k)}| \leq -\sigma \; \nabla f(x^{(k)})^T s^{(k)}, \tag{2}$$

where $0 < \rho < \sigma < 1$. We will denote $f^{(k)} = f(x^{(k)})$, $g^{(k)} = \nabla f(x^{(k)})$, $G^{(k)} = \nabla^2 f(x^{(k)})$, $\delta^{(k)} = x^{(k+1)} - x^{(k)}$ and $\gamma^{(k)} = g^{(k+1)} - g^{(k)}$.

Most of the methods for the unconstrained minimization problem are based on minimizing quadratic approximations to the function f. They differ in the way the search directions are calculated. Some typical such methods are described below.

Newton methods set

$$s^{(k+1)} = -(G^{(k+1)})^{-1} g^{(k+1)}, \tag{3}$$

with necessary modifications to $G^{(k+1)}$ if it is not positive definite, so as to make $s^{(k+1)}$ a descent direction. These methods are very efficient in terms of the number of iterations needed for convergence. However they require a knowledge of the Hessian matrix $G^{(k+1)}$, and to solve the linear system (3) needs $O(n^3)$ multiplications per iteration.

Quasi-Newton methods avoid the use of second derivatives by maintaining an approximation $H^{(k+1)}$ to the inverse of the Hessian (or the Hessian itself) and setting $s^{(k+1)} = -H^{(k+1)} g^{(k+1)}$. Since

$$\int_0^1 \; \nabla^2 f(x^{(k)} + t \; \delta^{(k)}) \; \delta^{(k)} \; dt = \gamma^{(k)},$$

so quasi-Newton methods (which approximate $(G^{(k)})^{-1}$) require $H^{(k+1)}$ to satisfy the quasi-Newton equation

$$H^{(k+1)}\gamma^{(k)} = \delta^{(k)}. \tag{4}$$

Clearly (4) does not specify the matrix $H^{(k+1)}$ completely. One way of specifying it is to assume (see Ref.1)

$$\begin{aligned} H^{(k+1)} =& H^{(k)} + c_1\delta^{(k)}\delta^{(k)^T} + c_2(H^{(k)}\gamma^{(k)}\delta^{(k)^T} + \delta^{(k)}\gamma^{(k)^T}H^{(k)}) \\ &+ c_3 H^{(k)}\gamma^{(k)}\gamma^{(k)^T}H^{(k)} \end{aligned} \tag{5}$$

and then use (4) to reduce the three parameters $c_1,\ c_2,\ c_3$ to a single parameter $\phi^{(k)}$. This gives the Broyden family of updates

$$\begin{aligned} H^{(k+1)} =& H^{(k)} - \frac{H^{(k)}\gamma^{(k)}\gamma^{(k)^T}H^{(k)}}{\gamma^{(k)^T}H^{(k)}\gamma^{(k)}} + \frac{\delta^{(k)}\delta^{(k)^T}}{\delta^{(k)^T}\gamma^{(k)}} \\ &+ \phi^{(k)}(\gamma^{(k)^T}H^{(k)}\gamma^{(k)})^{\frac{1}{2}}v^{(k)}v^{(k)^T}, \end{aligned} \tag{6}$$

where $v^{(k)} = \delta^{(k)}/\delta^{(k)^T}\gamma^{(k)} - H^{(k)}\gamma^{(k)}/\gamma^{(k)^T}H^{(k)}\gamma^{(k)}$. Clearly $H^{(k+1)}$ is symmetric if $H^{(k)}$ is, and if we denote

$$a^{(k)} = \delta^{(k)^T}B^{(k)}\delta^{(k)}, \quad b^{(k)} = \delta^{(k)^T}\gamma^{(k)}, \quad c^{(k)} = \gamma^{(k)^T}H^{(k)}\gamma^{(k)},$$

then $H^{(k+1)}$ is positive definite if $H^{(k)}$ is, $\delta^{(k)^T}\gamma^{(k)} > 0$ and $\phi^{(k)} > \bar{\phi}^{(k)}$. Here $\bar{\phi}^{(k)} = b^{(k)^2}/(b^{(k)^2} - a^{(k)}c^{(k)})$ is the value of $\phi^{(k)}$ that makes the right hand side matrix of (6) singular. We note that under line search condition (2), we always have $\delta^{(k)^T}\gamma^{(k)} > 0$.

Quasi-Newton methods are quite efficient, although they need to store the matrix $H^{(k+1)}$, and each iteration requires $O(n^2)$ multiplications to update the matrix. Among the members of the Broyden family, the BFGS update, given by $\phi^{(k)} = 1$, is thought to give the most efficient algorithm.

Conjugate gradient methods set

$$s^{(k+1)} = -g^{(k)} + \beta^{(k)}s^{(k)}. \tag{7}$$

They need only vector storage, but generally need more iterations and function evaluations than quasi-Newton methods. They also require an accurate line search routine.

There are many formulae (see Ref.1) for $\beta^{(k)}$ in (7). However most of them do not take into account the effect of inexact line searches. In an effort to find the best $\beta^{(k)}$, Liu and Storey (Ref.2) suggested a method which we will refer to as the LS method. It was found (Ref.3) that the method is actually a 2-dimensional Newton method in span $\{-g^{(k+1)},\ s^{(k)}\}$ and the new search direction is

$$s^{(k+1)} = -(g^{(k+1)},\ s^{(k)}) \left(\bar{G}^{(k+1)}\right)^{-1} \bar{g}^{(k+1)}, \tag{8}$$

where

$$\bar{G}^{(k+1)} = \begin{pmatrix} g^{(k+1)^T} G^{(k+1)} g^{(k+1)} & g^{(k+1)^T} G^{(k+1)} s^{(k)} \\ g^{(k+1)^T} G^{(k+1)} s^{(k)} & s^{(k)^T} G^{(k+1)} s^{(k)} \end{pmatrix}$$

and $\bar{g}^{(k+1)} = (g^{(k+1)^T} g^{(k+1)},\ g^{(k+1)^T} s^{(k)})^T$ are the Hessian and the gradient of the restriction of f on the two dimensional affine subspace $x^{(k+1)} +$ span $\{g^{(k+1)}, s^{(k)}\}$. We notice that the direction $s^{(k+1)}$ given by the LS method is equivalent to that given in equation (2.4) of Nazareth (Ref.4). The method of derivation is quite different however as are the implementations.

On quadratics with exact line searches, the two directions (7) and (8) are parallel. Numerical results show that compared with conjugate gradient methods, the LS method generally needs fewer iterations to converge. The idea of 2-dimensional Newton methods can be generalized to m-dimensional Newton methods ($m \geq 3$). The m-dimensional Newton direction in span $\{v_1^{(k)}, \ldots, v_m^{(k)}\}$ is

$$s^{(k+1)} = -V^{(k)}(\bar{G}^{(k+1)})^{-1}\bar{g}^{(k+1)}, \tag{9}$$

where $V^{(k)} = (v_1^{(k)}, \ldots, v_m^{(k)}) \in R^{n\times m}$, $\bar{g}^{(k+1)} = V^{(k)^T} g^{(k+1)}$ and $\bar{G}^{(k+1)} = V^{(k)^T} G^{(k+1)} V^{(k)}$.

Study of the m-dimensional Newton methods in the subspace spanned by the current gradient and some previous directions shows (see Ref.5) that as m increases, fewer iterations are required for convergence. However the gain obtained by reducing the number of iterations becomes smaller as m increases.

We believe that one of the problems with these m-dimensional Newton methods is that once the m-dimensional Hessian $\bar{G}^{(k+1)} = V^{(k)^T} G^{(k+1)} V^{(k)}$ is formed, it is only used once to form the m-dimensional Newton direction (9) and then it is discarded. If the second derivative information is stored in some way and used in future iterations to precondition the Hessian, then it might help the algorithm to converge more quickly. Of course in doing so, we have lost the advantage of low storage requirement.

The result turns out to be very interesting. If the proposed preconditioning technique is used on the 2-dimensional Newton method in span $\{g^{(k+1)},\ s^{(k)}\}$, a method that is very related to quasi-Newton methods is derived. If the approximation $G^{(k+1)}\delta^{(k)} \approx \gamma^{(k)}$ is used to estimate the 2-dimensional Hessian, then we get exactly a member of the Broyden family. Thus we have a new interpretation of quasi-Newton methods. The proposed technique can also be applied to other m-dimensional Newton methods.

There are strong similarities between the derivation of our methods and the work of Nazareth (Ref.6, 7). The motivation here, however, is quite different from that of Nazareth who is concerned with using BFGS or Newton methods with limited storage by restricting them to some affine subspaces.

2. The Preconditioned m-dimensional Newton Method

Assume that the current point is $x^{(k+1)}$, with the symmetric positive definite matrix $H^{(k)}$ as the last approximation to the inverse of the Hessian, and the last search direction is $s^{(k)} = -H^{(k)}g^{(k)}$. Let $H^{(k)} = Z^{(k)}Z^{(k)^T}$ with $Z^{(k)} \in R^{n\times n}$ and nonsingular. Denote $Z^{(k)^{-T}} = (Z^{(k)^{-1}})^T$. Make a transformation of variables

$$y = Z^{(k)^{-1}}x. \tag{10}$$

If we use the subscript "y" to denote the appropriate quantity in y-space, then the gradient, Hessian etc. in y-space are

$$g_y = Z^{(k)^T}g,$$

$$G_y = Z^{(k)^T}G\ Z^{(k)},$$

$$\delta_y = Z^{(k)^{-1}}\delta,$$

$$\gamma_y = Z^{(k)^T}\gamma.$$

By making the transformation (10), provided that $H^{(k)}$ is a good approximation to $(G^{(k+1)})^{-1}$, the Hessian $G_y^{(k+1)} = Z^{(k)^T}G^{(k+1)}Z^{(k)}$ will be "better conditioned", so that the unconstrained minimization problem in y-space will be easier to solve.

Thus instead of minimizing f in x-space, we consider minimizing it in y-space. Choose m independent vectors $p_1^{(k)}, \ldots, p_m^{(k)}$ in y-space, form a matrix $P_y^{(k)}$ with these m vectors. Then the m-dimensional Newton direction from $y^{(k)} = Z^{(k)-1}x^{(k)}$ in the subspace span $\{P_y^{(k)}\}$ (for a matrix A, span $\{A\}$ denotes the range space of A) is

$$s_y^{(k+1)} = -P_y^{(k)}\left(\bar{G}_y^{(k+1)}\right)^{-1}\ \bar{g}_y^{(k+1)}, \tag{11}$$

where

$$\bar{G}_y^{(k+1)} = P_y^{(k)T} G_y^{(k+1)} P_y^{(k)}$$

and

$$\bar{g}_y^{(k+1)} = P_y^{(k)T} g_y^{(k+1)}$$

are the Hessian and the gradient of the restriction of f on the affine subspace $y^{(k+1)} + \text{span}\ \{P_y^{(k)}\}$. We shall also denote

$$\bar{\delta}_y^{(k)} = P_y^{(k)T} \delta_y^{(k)}$$

and

$$\bar{\gamma}_y^{(k)} = P_y^{(k)T} \gamma_y^{(k)}.$$

Clearly the m-dimensional Newton direction $s_y^{(k+1)}$ will not change if $P_y^{(k)}$ is replaced by another n by m matrix with the same range space. Thus without loss of generality we can assume that $P_y^{(k)}$ is orthogonal. Let $Q_y^{(k)}$ be its orthogonal complement and let $\Omega^{(k)} = (P_y^{(k)},\ Q_y^{(k)}) \in R^{n \times n}$.

Back in x-space, the direction (11) is

$$\begin{aligned} s^{(k+1)} &= -Z^{(k)} P_y^{(k)} (\bar{G}_y^{(k+1)})^{-1} P_y^{(k)T} Z^{(k)^T} g^{(k+1)} \\ &= -P^{(k)} (P^{(k)T} G^{(k+1)} P^{(k)})^{-1} P^{(k)^T} g^{(k+1)}, \end{aligned} \tag{12}$$

where we denote $P^{(k)} = Z^{(k)} P_y^{(k)}$.

Since we know the m-dimensional Hessian $\bar{G}_y^{(k+1)}$, we prefer not to discard this information, therefore we make an estimate of the full Hessian $G^{(k+1)}$ with this information. Consider the matrix

$$\Omega^{(k)^T} G_y^{(k+1)} \Omega^{(k)} = \begin{pmatrix} \bar{G}_y^{(k+1)} & P_y^{(k)^T} G_y^{(k+1)} Q_y^{(k)} \\ Q_y^{(k)^T} G_y^{(k+1)} P_y^{(k)} & Q_y^{(k)^T} G_y^{(k+1)} Q_y^{(k)} \end{pmatrix}. \tag{13}$$

Since no curvature information in the range space of $Q_y^{(k)}$ is known, we will extend the three submatrices of (13) involving $Q_y^{(k)}$ in some way. Noticing that if $H^{(k)}$ is a good approximation of $G^{(k+1)}$, then $G_y^{(k+1)} \approx I$, so $P_y^{(k)^T} G_y^{(k+1)} Q_y^{(k)} \approx 0$ and $Q_y^{(k)^T} G_y^{(k+1)} Q_y^{(k)} \approx I$, thus a convenient way is to assign

$$\Omega^{(k)^T} G_y^{(k+1)} \Omega^{(k)} \approx \begin{pmatrix} \bar{G}_y^{(k+1)} & 0 \\ 0 & I \end{pmatrix}. \tag{14}$$

Then

$$(G_y^{(k+1)})^{-1} \approx \Omega^{(k)} \begin{pmatrix} (\bar{G}_y^{(k+1)})^{-1} & 0 \\ 0 & I \end{pmatrix} \Omega^{(k)^T}$$
$$= P_y^{(k)} \, (\bar{G}_y^{(k+1)})^{-1} P_y^{(k)^T} + I - P_y^{(k)} P_y^{(k)^T}.$$

In x-space this becomes

$$(G^{(k+1)})^{-1} \approx Z^{(k)} \Omega^{(k)} \begin{pmatrix} (\bar{G}_y^{(k+1)})^{-1} & 0 \\ 0 & I \end{pmatrix} \Omega^{(k)^T} Z^{(k)^T}$$
$$= P^{(k)} \, (P^{(k)^T} G^{(k+1)} P^{(k)})^{-1} P^{(k)} + H^{(k)} - P^{(k)} P^{(k)^T}.$$

Therefore in x-space a good approximation to the inverse of the Hessian is

$$\begin{aligned} H^{(k+1)} &= Z^{(k)} \Omega^{(k)} \begin{pmatrix} (\bar{G}_y^{(k+1)})^{-1} & 0 \\ 0 & I \end{pmatrix} \Omega^{(k)^T} Z^{(k)^T} \\ &= P^{(k)} \, (P^{(k)^T} G^{(k+1)} P^{(k)})^{-1} P^{(k)^T} + H^{(k)} - P^{(k)} P^{(k)^T}. \end{aligned} \tag{15}$$

If we assume from now on that $g_y^{(k+1)} \in \text{span}\,\{P_y^{(k)}\}$, then (12) can also be written as

$$s^{(k+1)} = -H^{(k+1)} g^{(k+1)}, \tag{16}$$

because for any $d \in \text{span}\,\{P_y^{(k)}\}$, $(H^{(k)} - P^{(k)} P^{(k)^T}) Z^{(k)^{-T}} d = Z^{(k)} Q_y^{(k)} Q_y^{(k)^T} d = 0$. The method given by (15) and (16) will be called the preconditioned m-dimensional Newton method.

Note 2.1 If $\bar{G}_y^{(k+1)}$ is positive definite, then the matrix (15) is also positive definite. Otherwise it is possible to modify the matrix $\bar{G}_y^{(k+1)}$ to make it positive definite, for example, by adding to it a positive definite multiple of the unit matrix.

Note 2.2 In formula (15), if $P_y^{(k)}$ in $\Omega^{(k)}$ is replaced by another orthogonal matrix, R say, having the same range space, then $P_y^{(k)}$ must equal R multiplied by an $m \times m$ orthogonal matrix, and the resulting matrix $H^{(k+1)}$ will be unchanged.

Note 2.3 Denote $\bar{H}^{(k+1)} = (\bar{G}_y^{(k+1)})^{-1}$, then (15) becomes

$$\begin{aligned} H^{(k+1)} &= Z^{(k)} \Omega^{(k)} \begin{pmatrix} \bar{H}^{(k+1)} & 0 \\ 0 & I \end{pmatrix} \Omega^{(k)^T} Z^{(k)^T} \\ &= P^{(k)} \bar{H}^{(k+1)} P^{(k)^T} + H^{(k)} - P^{(k)} P^{(k)^T}, \end{aligned} \tag{17}$$

If we assume $\gamma_y^{(k)} \in \text{span}\ \{P_y^{(k)}\}$, then $\left(H^{(k)} - P^{(k)}P^{(k)^T}\right)\gamma^{(k)} = 0$, thus $H^{(k+1)}$ satisfies the quasi-Newton equation $H^{(k+1)}\gamma^{(k)} = \delta^{(k)}$ if and only if $P^{(k)}\bar{H}^{(k+1)}P^{(k)^T}\gamma^{(k)} = \delta^{(k)}$, or

$$\bar{H}^{(k+1)}\bar{\gamma}_y^{(k)} = \bar{\delta}_y^{(k)}. \tag{18}$$

Note 2.4 If $g_y^{(k+1)},\ \gamma_y^{(k)} \in \text{span}\ \{P_y^{(k)}\}$, then $\delta_y^{(k)} \in \text{span}\ \{P_y^{(k)}\}$ (because $g_y^{(k+1)} = \gamma_y^{(k)} - \lambda^{(k)}\delta_y^{(k)}$). Further assume $G^{(k+1)}\delta^{(k)} = \gamma^{(k)}$, then

$$\begin{aligned} \bar{G}_y^{(k+1)}\bar{\delta}_y^{(k+1)} &= P_y^{(k)^T}G_y^{(k+1)}P_y^{(k)}P_y^{(k)^T}\delta_y^{(k)} \\ &= P_y^{(k)^T}Z^{(k)^T}G^{(k+1)}Z^{(k)}\delta_y^{(k)} \\ &= \bar{\gamma}_y^{(k)}, \end{aligned}$$

or $\bar{H}_y^{(k+1)}\bar{\gamma}_y^{(k)} = \bar{\delta}_y^{(k)}$, thus by Note 2.3, under these two assumptions the matrix (15) satisfies the quasi-Newton equation.

Note 2.5 For maintaining the positive definiteness of $H^{(k+1)}$, instead of updating $H^{(k)}$, it would be better to update its factorization $Z^{(k)}$. If $\bar{H}^{(k+1)}$ can be factorized as $L^{(k+1)}L^{(k+1)^T}$, then

$$H^{(k+1)} = Z^{(k+1)}Z^{(k+1)^T}$$

with

$$Z^{(k+1)} = Z^{(k)}\Omega^{(k)}\begin{pmatrix} L^{(k+1)} & 0 \\ 0 & I \end{pmatrix}.$$

The matrix $\Omega^{(k)}$ can be formed as follows. If $p_1^{(k)}, p_2^{(k)}, \ldots, p_m^{(k)}$ are m independent vectors in y-space, to form an orthogonal matrix $P_y^{(k)}$ with these m vectors, the matrix $\Omega^{(k)} = (P_y^{(k)},\ Q_y^{(k)})$ can be chosen as the orthogonal matrix such that

$$\Omega^{(k)^T}(p_1^{(k)}, \ldots, p_m^{(k)}) = \left(\begin{array}{ccc} \star & \ldots & \star \\ & \ddots & \vdots \\ & & \star \\ & \mathbf{O} & \end{array}\right),$$

where "⋆" denotes those elements of a matrix that may not necessarily be zero. For example if $p_1^{(k)} = \gamma_y^{(k)} = Z^{(k)^T}\gamma^{(k)}$ and $p_2^{(k)} = \delta_y^{(k)} = Z^{(k)^{-1}}\delta^{(k)}$, then

$$\Omega^{(k)^T}(Z^{(k)^T}\gamma^{(k)},\ Z^{(k)^{-1}}\delta^{(k)}) = \begin{pmatrix} \star & \star \\ 0 & \star \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}. \tag{19}$$

3. A Preconditioned 2-dimensional Newton Method

Having set up the framework for the preconditioned m-dimensional Newton methods, we now look at a special case.

For $m = 2$, it is natural to try the preconditioned LS method, that is the method in span $\{g_y^{(k+1)},\ \delta_y^{(k)}\}$. However, in y-space, it is interesting that

$$\begin{aligned} g_y^{(k+1)} &= Z^{(k)^T} g^{(k+1)} \\ &= Z^{(k)^T} \gamma^{(k)} + Z^{(k)^T} g^{(k)} \\ &= \gamma_y^{(k)} - \lambda^{(k)} Z^{(k)^{-1}} \delta^{(k)} \\ &= \gamma_y^{(k)} - \lambda^{(k)} \delta_y^{(k)}. \end{aligned}$$

Thus span $\{g_y^{(k+1)},\ \delta_y^{(k)}\}$ = span $\{\gamma_y^{(k)},\ \delta_y^{(k)}\}$. So by Note 2.2, the orthogonal matrix $P_y^{(k)}$ can be formed using $\gamma_y^{(k)}$ and $\delta_y^{(k)}$ instead of $g_y^{(k+1)}$ and $\delta_y^{(k+1)}$ without affecting the final matrix $H^{(k+1)}$. We choose to work with $\gamma_y^{(k)}$, $\delta_y^{(k)}$ because by doing so it is easier to see the relationship between the preconditioned 2-dimensional Newton method and the Broyden family (6).

Orthonormalizing $\gamma_y^{(k)}$, $\delta_y^{(k)}$ gives

$$P_y^{(k)} = \left(\frac{\gamma_y^{(k)}}{\|\gamma_y^{(k)}\|},\quad \frac{\delta_y^{(k)} - \dfrac{\delta_y^{(k)^T} \gamma_y^{(k)}}{\|\gamma_y^{(k)}\|^2} \gamma_y^{(k)}}{\sqrt{\|\delta_y^{(k)}\|^2 - \dfrac{(\delta_y^{(k)^T} \gamma_y^{(k)})^2}{\|\gamma_y^{(k)}\|^2}}} \right)$$

thus

$$P^{(k)} = Z^{(k)} P_y^{(k)} = \left(\frac{H^{(k)} \gamma^{(k)}}{\sqrt{c^{(k)}}},\quad \frac{c^{(k)} \delta^{(k)} - b^{(k)} H^{(k)} \gamma^{(k)}}{\sqrt{c^{(k)^2} a^{(k)} - b^{(k)^2} c^{(k)}}} \right).$$

By the form of $P^{(k)}$ and (15) we know that $H^{(k+1)}$ must be of the form (5). If f is a quadratic function, then by Note 2.4, $H^{(k+1)}$ also satisfies the quasi-Newton equation, so it must be a member of Broyden's family. Therefore we have:

Theorem 3.1 The preconditioned LS method has the quadratic termination property. □

We now analyse the method more carefully. Assume $G^{(k+1)} \delta^{(k)} = \gamma^{(k)}$ (which is true for quadratics) and denote $T^{(k)} = \gamma^{(k)^T} H^{(k)} G^{(k+1)} H^{(k)} \gamma^{(k)}$. Then after some calculation we have

$$\begin{aligned}\bar{G}_y^{(k+1)} &= P^{(k)^T} G^{(k+1)} P^{(k)} \\ &= \begin{pmatrix} \dfrac{T^{(k)}}{c^{(k)}} & \dfrac{c^{(k)^2} - b^{(k)}T^{(k)}}{c^{(k)}\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}} \\ \dfrac{c^{(k)^2} - b^{(k)}T^{(k)}}{c^{(k)}\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}} & \dfrac{b^{(k)}(b^{(k)}T^{(k)} - c^{(k)^2})}{c^{(k)}(a^{(k)}c^{(k)} - b^{(k)^2})} \end{pmatrix}. \end{aligned} \tag{20}$$

So

$$\bar{H}^{(k+1)} = (\bar{G}_y^{(k+1)})^{-1} = \begin{pmatrix} \dfrac{b^{(k)}}{c^{(k)}} & \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} \\ \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} & \dfrac{(a^{(k)}c^{(k)} - b^{(k)^2})T^{(k)}}{(b^{(k)}T^{(k)} - c^{(k)^2})c^{(k)}} \end{pmatrix}. \tag{21}$$

Clearly (21) satisfies (18) because

$$\bar{\gamma}_y^{(k)} = P_y^{(k)^T}\gamma_y = \left(\sqrt{c^{(k)}}, 0\right)^T$$

and

$$\bar{\delta}_y^{(k)} = P_y^{(k)^T}\delta_y = \left(\frac{b^{(k)}}{\sqrt{c^{(k)}}}, \quad \frac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{\sqrt{c^{(k)}}}\right)^T.$$

It was shown in Ref.8 that Broyden's family of updates (6) can be written in the form (17), that is,

$$H^{(k+1)} = Z^{(k)}\Omega^{(k)} \begin{pmatrix} \bar{H}^{(k+1)} & 0 \\ 0 & I \end{pmatrix} \Omega^{(k)^T} Z^{(k)^T},$$

with $\Omega^{(k)}$ an orthogonal matrix satisfying (19) and

$$\bar{H}^{(k+1)} = \begin{pmatrix} \dfrac{b^{(k)}}{c^{(k)}} & \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} \\ \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} & \dfrac{a^{(k)}}{c^{(k)}} - \dfrac{b^{(k)}}{c^{(k)}} + 1 + \phi^{(k)}\left(\dfrac{a^{(k)}c^{(k)}}{b^{(k)^2}} - 1\right) \end{pmatrix}. \tag{22}$$

Thus equating (21) and (22) we see that under the assumption $G^{(k+1)}\gamma^{(k)} = \delta^{(k)}$, the preconditioned LS method is in the Broyden's family with

$$\begin{aligned}\phi^{(k)} &= \frac{b^{(k)^2}}{b^{(k)^2} - a^{(k)}c^{(k)}} + \frac{b^{(k)}c^{(k)}}{b^{(k)}T^{(k)} - c^{(k)^2}} \\ &= \bar{\phi}^{(k)} + \frac{b^{(k)}c^{(k)}}{b^{(k)}T^{(k)} - c^{(k)^2}}. \end{aligned} \tag{23}$$

This formula gives the "best" $\phi^{(k)}$ for the Broyden family (6), in the sense that it corresponds to the preconditioned LS method. The LS method, as mentioned previously, has fewer iteration counts compared with conjugate gradient methods.

If $G^{(k+1)}\delta^{(k)} = \gamma^{(k)}$ and $G^{(k+1)}$ is positive definite, then it is easy to prove, using the Cauchy-Schwarz inequality, that $b^{(k)}T^{(k)} - c^{(k)^2} \geq 0$, therefore $\phi^{(k)} \geq \bar{\phi}^{(k)}$, which means that the matrix $H^{(k+1)}$ for the preconditioned LS method is a positive definite member of the Broyden family.

Conversely any method given by a member of the Broyden family can also be viewed as a preconditioned LS method with the approximation $G^{(k)}\gamma^{(k)} = \delta^{(k)}$ and with proper choice of $T^{(k)}$.

When we do not wish to use second derivatives to calculate $T^{(k)}$ in (21), then we have to set the parameter $T^{(k)}$ in some way. The various choices of $T^{(k)}$, or equivalently of $\phi^{(k)}$ in (22), can be interpreted in the following way (see also Ref.9).

Consider in y-space the problem in the 2-dimensional affine subspace $y^{(k+1)}+$ span$\{P_y^{(k)}\}$, with $P_y^{(k)}$ the orthogonal matrix formed with $\gamma_y^{(k)}$, $\delta_y^{(k)}$. Then since $\delta_y^{(k)}$, $\gamma_y^{(k)}$, $g_y^{(k+1)} \in$ span$\{P_y^{(k)}\}$, so under the new coordinate system they become

$$\bar{\delta}_y^{(k)} = P_y^{(k)^T}\delta_y^{(k)} = \left(\frac{b^{(k)}}{\sqrt{c^{(k)}}}, \frac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{\sqrt{c^{(k)}}}\right)^T, \tag{24}$$

$$\bar{\gamma}_y^{(k)} = P_y^{(k)^T}\gamma_y^{(k)} = \left(\sqrt{c^{(k)}}, 0\right)^T \tag{25}$$

and

$$\bar{g}_y^{(k+1)} = P_y^{(k)^T} g_y^{(k+1)}.$$

The starting point $y^{(k+1)}$ becomes the orgin $\bar{y}^{(k+1)} = 0$.

Assume the function f in the 2-dimensional affine subspace is quadratic, then the point that would be reached if the line search were exact is (see Figure 4.1)

$$\tilde{y}^{(k+1)} = \bar{y}^{(k+1)} - \frac{\bar{g}_y^{(k+1)^T}\bar{\delta}_y^{(k)}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\bar{\delta}_y^{(k)},$$

and the gradient at this point is

$$\tilde{g}_y^{(k+1)} = \bar{g}_y^{(k+1)} - \frac{\bar{g}_y^{(k+1)^T}\bar{\delta}_y^{(k)}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\bar{\gamma}_y^{(k)}.$$

At the point $\tilde{y}^{(k+1)}$, if a search direction $\tilde{s}_y^{(k+1)}$ is generated using the gradient at this point and the previous direction, in the form (7), then

$$\begin{aligned}\tilde{s}_y^{(k+1)} &= -\tilde{g}_y^{(k+1)} + \frac{\tilde{g}_y^{(k+1)^T}\bar{\gamma}_y^{(k)}}{\bar{\delta}_y^{(k+1)^T}\bar{\gamma}_y^{(k)}}\bar{\delta}_y^{(k)} \\ &= -\left(I - \frac{\bar{\delta}_y^{(k)}\bar{\gamma}_y^{(k)^T}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\right)\left(I - \frac{\bar{\gamma}_y^{(k)}\bar{\delta}_y^{(k)^T}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\right)\bar{g}_y^{(k+1)}.\end{aligned}$$

Assume the steplength along $\tilde{s}_y^{(k+1)}$ is taken as $\alpha^{(k)}$, then the new point is $\tilde{y}^{(k+2)} = \tilde{y}^{(k+1)} + \alpha^{(k)}\tilde{s}_y^{(k+1)}$. However this point can be reached by searching along the following direction $\bar{s}_y^{(k+1)}$ from $\bar{y}^{(k+1)}$.

$$\begin{aligned}\bar{s}_y^{(k+1)} &= \tilde{y}^{(k+2)} - \bar{y}^{(k+1)} \\ &= -\left[\frac{\bar{\delta}_y^{(k)}\bar{\delta}_y^{(k)^T}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}} + \alpha^{(k)}\left(I - \frac{\bar{\delta}_y^{(k)}\bar{\gamma}_y^{(k)^T}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\right)\left(I - \frac{\bar{\gamma}_y^{(k)}\bar{\delta}_y^{(k)^T}}{\bar{\delta}_y^{(k)^T}\bar{\gamma}_y^{(k)}}\right)\right]\bar{g}_y^{(k)} \qquad (26) \\ &= -\bar{H}^{(k+1)}\bar{g}_y^{(k)}.\end{aligned}$$

Figure 4.1 The function f in the 2-dimensional affine subspace

It is easy to verify, by substituting (24) and (25) into (26), that

$$\bar{H}^{(k+1)} = \begin{pmatrix} \dfrac{b^{(k)}}{c^{(k)}} & \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} \\ \dfrac{\sqrt{a^{(k)}c^{(k)} - b^{(k)^2}}}{c^{(k)}} & \dfrac{a^{(k)}}{b^{(k)}} - \dfrac{a^{(k)}}{c^{(k)}} + \dfrac{\alpha^{(k)}a^{(k)}c^{(k)}}{b^{(k)^2}} \end{pmatrix}.$$

Equating this formula and (22) gives

$$\alpha^{(k)} = \left(1 - \frac{b^{(k)^2}}{a^{(k)}c^{(k)}}\right)\phi^{(k)} + \frac{b^{(k)^2}}{a^{(k)}c^{(k)}}.$$

Thus the BFGS update ($\phi^{(k)} = 1$) corresponds to steplength $\alpha^{(k)} = 1$, the DFP formula ($\phi^{(k)} = 0$) gives $\alpha^{(k)} = b^{(k)^2}/(a^{(k)}c^{(k)}) \leq 1$ and the singular value $\phi^{(k)} = \bar{\phi}^{(k)}$ gives $\alpha^{(k)} = 0$.

4. Implementations of the Preconditioned LS method

In this section we will implement the preconditioned LS method and compare it with the BFGS method.

As we mentioned in Note 2.5, for maintaining positive definiteness, it is better to update the factorization of $H^{(k)}$ rather than $H^{(k)}$ itself. Denote

$$\bar{Z}^{(k)} = \left(\bar{z}_1^{(k)}, \ldots, \bar{z}_n^{(k)}\right) = Z^{(k)}\Omega^{(k)},$$

then (17) becomes

$$H^{(k+1)} = \bar{Z}^{(k)} \begin{pmatrix} \bar{H}^{(k+1)} & 0 \\ 0 & I \end{pmatrix} \bar{Z}^{(k)},$$

and since $P^{(k)} = Z^{(k)}P_y^{(k)} = (\bar{z}_1^{(k)}, \ldots, \bar{z}_m^{(k)})$, so for $m = 2$, $\bar{H}^{(k+1)} = (\bar{G}_y^{(k+1)})^{-1}$ with

$$\bar{G}_y^{(k+1)} = P^{(k)^T}G^{(k+1)}P^{(k)} = \begin{pmatrix} \bar{z}_1^{(k)^T}G^{(k+1)}\bar{z}_1^{(k)} & \bar{z}_1^{(k)^T}G^{(k+1)}\bar{z}_2^{(k)} \\ \bar{z}_2^{(k)^T}G^{(k+1)}\bar{z}_1^{(k)} & \bar{z}_2^{(k)^T}G^{(k+1)}\bar{z}_2^{(k)} \end{pmatrix}. \qquad (27)$$

However on general nonquadratic functions, $\bar{G}_y^{(k+1)}$ may not be positive definite. Thus to ensure the positive definiteness of $H^{(k+1)}$, some modification to the 2×2 matrix $\bar{G}_y^{(k+1)}$ is necessary when it is not positive definite. Assume we that we wish to factorize

$$\bar{G}_y^{(k+1)} = \begin{pmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{pmatrix}$$

into

$$\begin{pmatrix} 1 & d_3 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ d_3 & 1 \end{pmatrix} = \begin{pmatrix} d_1 + d_2 d_3^2 & d_2 d_3 \\ d_2 d_3 & d_2 \end{pmatrix}, \qquad d_1, d_2 > 0.$$

Clearly $d_2 = g_{22}$ if $g_{22} > 0$. But if $g_{22} \leq 0$, then one way of modification is to let $d_2 = t$ for some positive constant t. Unfortunately it is difficult to find a suitable constant t to reflect the magnitude of the 2−dimensional Hessian. Thus we simply set $d_2 = -g_{22}$ if $g_{22} \leq 0$ (for the sake of simplicity we do not use other more sophisticated options). We treat d_1 in a similar way. Thus we have

$$d_2 = \begin{cases} g_{22}, & \text{if } g_{22} > 0, \\ -g_{22}, & \text{if } g_{22} \leq 0, \end{cases} \tag{28a}$$

$$d_3 = g_{12}/d_2 \tag{28b}$$

and

$$d_1 = \begin{cases} g_{11} - d_2 d_3^2, & \text{if } g_{11} - d_2 d_3^2 > 0, \\ -(g_{11} - d_2 d_3^2) & \text{if } g_{11} - d_2 d_3^2 \leq 0. \end{cases} \tag{28c}$$

So $\bar{H}^{(k+1)} = \left(\bar{G}_y^{(k+1)}\right)^{-1} = L^{(k+1)} L^{(k+1)^T}$ with

$$L^{(k+1)} = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{12} & \ell_{22} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{\sqrt{d_1}} & 0 \\ -\dfrac{d_3}{\sqrt{d_1}} & \dfrac{1}{\sqrt{d_2}} \end{pmatrix}. \tag{28d}$$

The preconditioned LS method is thus given by:

Algorithm 4.1

Step 1 Let $x^{(1)}$ be a starting point, set $Z^{(1)} = I$, $k = 1$. If $\|g^{(k)}\| \leq 10^{-5} \max\{1, \|x^{(k)}\|\}$ then stop.

Step 2 Form the search direction $s^{(k)} = -Z^{(k)} Z^{(k)^T} g^{(k)}$.

Step 3 Line search to get $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$, if $\|g^{(k+1)}\| \leq 10^{-5} \max\{1, \|x^{(k+1)}\|\}$ then stop.

Step 4 Let $\Omega^{(k)}$ be the orthogonal matrix such that

$$\Omega^{(k)^T} (Z^{(k)^T} \gamma^{(k)},\ Z^{(k)^{-1}} \delta^{(k)}) = \begin{pmatrix} \star & 0 & 0 & \dots & 0 \\ \star & \star & 0 & \dots & 0 \end{pmatrix}^T.$$

Let $\bar{Z}^{(k)} = Z^{(k)} \Omega^{(k)}$.

Step 5 Form the 2−dimensional Hessian (27) (with finite differences) and factorize it as in (28).

Step 6 Set $Z^{(k+1)} = (z_1^{(k+1)}, \ldots, z_n^{(k+1)})$ with

$$z_1^{(k+1)} = \ell_{11}\bar{z}_1^{(k)} + \ell_{12}\bar{z}_2^{(k)}, \qquad (29a)$$

$$z_2^{(k+1)} = \ell_{22}\bar{z}_2^{(k)} \qquad (29b)$$

and

$$z_i^{(k+1)} = \bar{z}_i^{(k+1)}, \quad i = 3, \ldots, n. \qquad (29c)$$

Set $k := k+1$, go to Step 2.

To avoid excessive details, the algorithm given here is simplified. We refer to Ref.8 for complete details including the generation of the orthogonal matrix $\Omega^{(k)}$. There it was found that Algorithm 4.1 needs about $4n^2$ multiplications per iteration.

As stated in Ref.8, any method in the Broyden family can also be implemented using Algorithm 4.1. Factorizing (22) gives $\bar{H}^{(k+1)} = L^{(k+1)}L^{(k+1)^T}$ with

$$L^{(k+1)} = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{12} & \ell_{22} \end{pmatrix} = \begin{pmatrix} \sqrt{\dfrac{b^{(k)}}{c^{(k)}}} & 0 \\ \sqrt{\dfrac{a^{(k)}c^{(k)} - b^{(k)^2}}{b^{(k)}c^{(k)}}} & \sqrt{1 + \phi^{(k)}\left(\dfrac{a^{(k)}c^{(k)}}{b^{(k)^2}} - 1\right)} \end{pmatrix}. \qquad (30)$$

We shall denote [PLS] as the preconditioned LS algorithm using Algorithm 4.1 with $L^{(k+1)}$ given by (28), and [BFGS] as the BFGS algorithm using Algorithm 4.1 with $L^{(k+1)}$ given by (30) and $\phi^{(k)} = 1$. We also tried an algorithm [PLS-BFGS], that is, if $\bar{G}^{(k+1)}$ is positive definite, then $L^{(k+1)}$ is given by (28), otherwise switch to BFGS by setting $L^{(k+1)}$ to (30).

Both [PLS] and [PLS-BFGS] need two extra gradient evaluations to calculate the 2-dimensional Hessian (27) using the following finite difference formula,

$$G^{(k+1)}\bar{z}_i^{(k)} \approx \frac{g(x^{(k+1)} + t^{(k)}\bar{z}_i^{(k)}) - g^{(k+1)}}{t^{(k)}}, \quad i = 1, 2,$$

where we take $t^{(k)} = 10^{-8}/\|\bar{z}_i^{(k)}\|$. Of course it is also possible to calculate (27) using finite differences of function values, in which case five extra function evaluations are needed (see Ref.4).

To reduce the number of gradient evaluations, we can use the approximation $G^{(k+1)}\delta^{(k)} \approx \gamma^{(k)}$, which gives (21). Factorization of (21) gives $\bar{H}^{(k+1)} = L^{(k+1)}L^{(k+1)^T}$ with

$$L^{(k+1)} = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{12} & \ell_{22} \end{pmatrix} = \begin{pmatrix} \sqrt{\dfrac{b^{(k)}}{c^{(k)}}} & 0 \\ \sqrt{\dfrac{a^{(k)}c^{(k)} - b^{(k)^2}}{b^{(k)}c^{(k)}}} & \sqrt{\dfrac{(a^{(k)}c^{(k)} - b^{(k)^2})c^{(k)}}{(b^{(k)}T^{(k)} - c^{(k)^2})b^{(k)}}} \end{pmatrix}, \tag{31}$$

where $T^{(k)} = \gamma^{(k)^T}H^{(k)}G^{(k+1)}H^{(k)}\gamma^{(k)}$. However, as $x^{(k)}$ tends to the minimum point, we have found in our numerical work that usually $a^{(k)}c^{(k)} - b^{(k)^2} \to 0$ and $b^{(k)}T^{(k)} - c^{(k)^2} \to 0$, thus it is very difficult to calculate accurately the second diagonal element of $L^{(k+1)}$ using (31) because of round off errors. Our implementation of Algorithm 4.1 using (31) therefore frequently gives erratic results. To overcome this difficulty, notice that (20) shows that the second diagonal element of $\bar{G}_y^{(k+1)}$ is

$$g_{22} = \frac{b^{(k)}(b^{(k)}T^{(k)} - c^{(k)^2})}{c^{(k)}(a^{(k)}c^{(k)} - b^{(k)^2})}. \tag{32}$$

However, $g_{22} = \bar{z}_2^T G^{(k+1)} \bar{z}_2$ can be calculated easily with finite differences. Thus substituting (32) into (31) gives

$$L^{(k+1)} = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{12} & \ell_{22} \end{pmatrix} = \begin{pmatrix} \sqrt{\dfrac{b^{(k)}}{c^{(k)}}} & 0 \\ \sqrt{\dfrac{a^{(k)}c^{(k)} - b^{(k)^2}}{b^{(k)}c^{(k)}}} & \sqrt{\dfrac{1}{g_{22}}} \end{pmatrix}. \tag{33}$$

Since (33) corresponds to a member of the Broyden family with $\phi^{(k)}$ given by (23), and the $\phi^{(k)}$ is "best" in our sense, so we denote the algorithm using (33) as [BESTϕ]. In the implementation, we actually took $\ell_{22} = 1/(|g_{22}|)^{\frac{1}{2}}$.

As Shanno and Phua (Ref.10) found, initial scaling is usually beneficial to the BFGS algorithm, thus we also tried the initial scaling versions of the four algorithms, which we denote respectively as [PLS2], [BFGS2], [PLS-BFGS2] and [BESTϕ2]. For the BFGS algorithm, initial scaling with $\xi^{(1)} = \delta^{(1)^T}\gamma^{(1)}/\gamma^{(1)^T}H^{(1)}\gamma^{(1)}$ (here $H^{(1)} = Z^{(1)}Z^{(1)^T}$) corresponds to (see Ref.8) replacing (29b) and (29c), when $k = 1$, with

$$z_2^{(2)} = \sqrt{\xi^{(1)}a^{(1)}c^{(1)}/b^{(1)^2}}\ \bar{z}_2^{(1)}$$

and

$$z_i^{(2)} = \sqrt{\xi^{(1)}}\ \bar{z}_i^{(1)}, \quad i = 3, \ldots, n.$$

For the other three algorithms, by initial scaling we mean replacing (29c) when $k = 1$ with

$$z_i^{(2)} = \sqrt{\xi^{(1)}}\ \bar{z}_i^{(1)}, \quad i = 3, \ldots, n.$$

The eight algorithms are implemented on an HP9000/870 computer with double precision in FORTRAN. The line search routine satisfies the line search conditions (1) and (2) with $\rho = 10^{-4}$ and $\sigma = 0.9$. If during the line search the length of the interval in which the final step length is predicted to lie becomes less than 10^{-15}, then the line search is assumed to fail. The initial step length is always taken as one, except when $k = 1$, in which case we take it as $\max\{2,\ (\mathrm{EST} - f^{(k)})/g^{(k)^T} s^{(k)}\}$ where we set EST equal to zero.

We used the eight algorithms on the first 31 test functions of Moré et al. (Ref.11). These functions are all sums of squares with the number of square terms either as given, if a number is recommended in their paper, or set to 100. Standard starting points are used.

Table 4.1 contains the results of the eight algorithms. The first column gives the number of the test function, the second column gives the number of variables, the third to the tenth columns contains "NI/NFE/NGE" for each algorithm on each test function, where "NI" denotes the number of iterations, "NFE" the number of function evaluations and "NGE" the number of gradient evaluations. We use "F2" to denote failure due to line search, "F3" failure due to overflow of functions (gradients). The last row of the table gives the totals of NI, NFE and NGE. In calculating these totals, we do not count functions 6, 10, 14 and 17 on which some algorithms fail.

From this table, we can see that in terms of NI and NFE, algorithms [PLS] and [PLS-BFGS] are clearly better than [BFGS] and [BFGS2]. Algorithms [PLS2] and [PLS-BFGS2] are even better. However, due to the extra gradient evaluations needed in calculating the 2−dimensional Hessian, NGE for these four preconditioned LS algorithms is higher than that for the two BFGS algorithms. The two algorithms using the "best" $\phi^{(k)}$ are also quite good in terms of NI and NFE if we compare their results on each problem separately with those of the two BFGS algorithms. However because of the erratic results of [BESTϕ] on problem 24, the total NI and NFE of [BESTϕ] is quite high.

To test the algorithms further, we tried them on functions 20 to 30 (on which the number of variables is adjustable) and set the number of variables to 100 instead of 12. The results are in Table 4.2. Clearly the conclusions of the last paragraph still apply. We also notice that now the results of [PLS] and [PLS-BFGS] are exactly the same. This is because during the iterations for solving each test problem, the 2−dimensional Hessian (27) remains positive definite, thus the two algorithms are equivalent. In fact when solving the complete set of 31 test functions in Table 4.1, the 2−dimensional Hessian also remains positive definite for most of the iterations.

We have also tried the algorithms on functions 20 to 31, with 12 variables. But instead of using the standard starting point $x^{(1)}$, we tried $10x^{(1)}$. The results in Table 4.3 again show that in terms of NI and NF, the six preconditioned LS algorithms are better than the two BFGS algorithms.

5. Discussion

In this paper we have introduced the preconditioned m-dimensional Newton method. We have seen that any method in the Broyden rank-2 Family can be regarded as a preconditioned 2−dimensional Newton method in span $\{g_y^{(k+1)},\ \delta_y^{(k+1)}\}$, with the approximation $G^{(k+1)}\delta^{(k)} = \gamma^{(k)}$. This gives us an explanation of the nature of quasi-Newton methods. Numerical results on the preconditioned LS algorithms show clearly that they take fewer iterations and function evaluations than BFGS alghorithms, but that they take more gradient evaluations than the latter.

Research is currently being carried out on preconditioned m-dimensional Newton methods for $m \geq 3$. There are many possibilities for the choice of the m-dimensional subspaces. For example, with $m = 3$, one can choose span $\{g_y^{(k+1)},\ G_y^{(k+1)}\ g_y^{(k+1)},\ (G_y^{(k+1)})^2 g_y^{(k+1)}\}$, or span $\{g_y^{(k+1)},\ \delta_y^{(k)},\ \delta_y^{(k-1)}\}$.

In formula (13), after calculating $G_y^{(k+1)}P_y^{(k)}$, in fact three of the four blocks are known, so the result of calculating these three blocks by finite differences and assigning properly the fourth block deserves investigation. Note however that the block in the bottom right-hand corner could not then be simply set to the unit matrix since this may not preserve positive definiteness.

[1] Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, Chichester, 1987.

[2] Liu, Y. and Storey, C., *Efficient Generalized Conjugate Gradient Algorithms*, part I: Theory, Journal of Optimization Theory and Applications, Vol. 69, pp. 129-137, 1991.

[3] Hu, Y.F. and Storey, C., *Efficient Generalized Conjugate Gradient Algorithms, part II: Implementation*, Journal of Optimization Theory and Applications, Vol. 69, pp. 139-152, 1991.

[4] Nazareth, J. L., *An Algorithm Based upon Successive Affine Reducation and Newton's Method*, in Computing Methods in Applied Sciences and Engineering, VII, eds. Glowinski, R. and Lions, J. L., Elsevier Science Publishers B. V. (North-Holland), 1986.

[5] Hu, Y.F., *New Conjugate Gradient Methods and Their Connections with Quasi-Newton and Low-Dimensional Newton Methods*, Ph.D. Thesis, Department of Mathematical Sciences, Loughborough University of Technology, England, 1992.

[6] Nazareth, J.L., *Conjugate Gradient Methods Less Dependent on Conjugacy*, SIAM Review, Vol. 28, pp. 501-511, 1986.

[7] Nazareth, J.L., *The Method of Successive Affine Reduction for Nonlinear Minimization*, Mathematical Programming, Vol. 35, pp. 97-109, 1986.

[8] Hu, Y.F. and C. Storey., *A Family of Optimally Conditioned Quasi-Newton Updates for Unconstrained Optimization*, Mathematics Report A 143, Department of Mathematical Sciences, Loughborough University of Technology, England, 1991.

[9] Hu, Y.F. and C. Storey., *Motivating Quasi-Newton Updates by Preconditioned Conjugate Gradient Methods*, Mathematics Report A 150, Department of Mathematical Sciences, Loughborough University of Technology, England, 1991. (presented at the 14th International Symposium on Mathematical Programming, Amsterdam, The Netherlands, August 5-9, 1991.)

[10] Shanno, D.F. and Phua, K.H., *Matrix Conditioning and Nonlinear Optimization*, Mathematical Programming, Vol. 14, pp. 149-160, 1978.

[11] Moré, J.J., Garbow, B.S. and Hillstrom, K.E., *Testing Unconstrained Optimization Software*, ACM Transactions on Mathematical Software, Vol. 7, pp. 17-41, 1978.