Mathematical Programming 53 (1992) 17–44
North-Holland

Partitioned quasi-Newton methods for nonlinear equality constrained optimization

Thomas F. Coleman*

Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA

Peter A. Fenyes

General Motors Research Laboratories, Warren, MI, USA

Received 16 August 1988
Revised manuscript received 31 October 1989

We derive new quasi-Newton updates for the (nonlinear) equality constrained minimization problem. The new updates satisfy a quasi-Newton equation, maintain positive definiteness on the null space of the active constraint matrix, and satisfy a minimum change condition. The application of the updates is not restricted to a small neighbourhood of the solution. In addition to derivation and motivational remarks, we discuss various numerical subtleties and provide results of numerical experiments.

AMS/MOS 1980 Subject Classifications: 65H10, 65K05, 65K10.

Key words: Constrained optimization, equality constraints, numerical optimization, quasi-Newton method, secant method, sequential quadratic programming, SQP-method, augmented Lagrangian method, penalty function methods.

1. The nonlinear equality problem

In this paper we derive new quasi-Newton updates for the nonlinear equality constrained problem

$$\text{(NEP)} \quad \underset{x}{\text{minimize}} \quad f(x)$$

$$\text{subject to} \quad c(x)=0,$$

where $x \in \mathbb{R}^n$, $f:\mathbb{R}^n \to \mathbb{R}$ and $c:\mathbb{R}^n \to \mathbb{R}^t$ and $t \leqslant n$. The functions f and c_i, $i=1,\ldots,t$, are twice continuously differentiable. The new updates satisfy a quasi-Newton equation, maintain positive definiteness on the null space of the matrix of active gradients, and satisfy a minimum change condition. Furthermore, the application of the new updates is not restricted to a small neighborhood near the solution. To further motivate the development of our updates we first review optimality conditions and some existing update strategies.

* Research partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the US Department of Energy under grant DE-FG02-86ER25013.A000, and by the US Army Research Office through the Mathematical Sciences Institute, Cornell University.

It will be useful to define $A \in \mathbb{R}^{n\times t}$ as the matrix whose columns are the gradients of the constraints. In other words, $A(x) = [\nabla c_1(x), \ldots, \nabla c_t(x)]$. If A is of full column rank, we can define two matrices $Y \in \mathbb{R}^{n\times t}$ and $Z \in \mathbb{R}^{n\times(n-t)}$ using the QR factorization

$$A = Q\bar{R} = [Y \quad Z]\begin{bmatrix} R \\ 0 \end{bmatrix}$$

where $Q \doteq [Y\ Z]$ is orthogonal, $\bar{R}^{\mathrm{T}} \doteq [R^{\mathrm{T}}\ 0]$, and $R \in \mathbb{R}^{t\times t}$ is upper triangular. When defined this way, Y and Z have orthonormal columns and satisfy the following relationships: $QQ^{\mathrm{T}} = YY^{\mathrm{T}} + ZZ^{\mathrm{T}} = I_{n\times n}$, $Y^{\mathrm{T}}Y = I_{t\times t}$, $Z^{\mathrm{T}}Z = I_{m\times m}$, $Y^{\mathrm{T}}Z = 0_{t\times m}$, and $Z^{\mathrm{T}}Y = 0_{m\times t}$ where we have taken $m = n - t$. Thus, Y and Z represent orthogonal bases for the range of A and the null space of A^{T}, respectively.

The solution to (NEP) must satisfy the first order necessary condition for optimality:

First order necessary condition. If x^* is a local minimizer of (NEP), Z is defined as above, and a constraint qualification (e.g. $A(x)$ has full column rank) holds at x^*, then x^* satisfies

$$Z(x^*)^{\mathrm{T}}\nabla_x f(x^*) = 0, \qquad c(x^*) = 0. \tag{1.1}$$

We may restate this in terms of the Lagrangian

$$L(x, \lambda) = f(x) - \sum_{i=1}^{t} \lambda_i c_i(x),$$

with $\lambda \in \mathbb{R}^t$. The first order necessary conditions for optimality then requires the existence of Lagrange multipliers λ^* such that x^* and λ^* satisfy

$$d_L(x^*, \lambda^*) \doteq \begin{pmatrix} \nabla_x L(x^*, \lambda^*) \\ \nabla \lambda L(x^*, \lambda^*) \end{pmatrix} = \begin{pmatrix} g(x^*) - A(x^*)\lambda(x^*) \\ c(x^*) \end{pmatrix} = (0) \tag{1.2}$$

where $d_L \in \mathbb{R}^{n+t}$ and we have defined $g \doteq \nabla_x f$ and $A\lambda = \sum \lambda_i \nabla_x c_i$.

In the remainder of this paper we will also assume that the solution (x^*, λ^*) satisfies the second order sufficiency condition. Specifically we assume:

- f and c_i, $i = 1, \ldots, t$, are twice continuously differentiable.
- $\nabla c_i(x^*)$, $i = 1, \ldots, t$, are linearly independent.
- $Z(x^*)^{\mathrm{T}}\nabla_{xx} L(x^*, \lambda^*)Z(x^*)$ is positive definite where $\nabla_{xx} L(x, \lambda)$ is the second derivative, or Hessian matrix, of the Lagrangian.

Note that although many authors make the strong assumption that the full Hessian at (x^*, λ^*) is positive definite, we assume that only the reduced Hessian matrix, $Z(x^*)^{\mathrm{T}}\nabla_{xx}L(x^*, \lambda^*)Z(x^*)$, is positive definite as required by the second order condition.

1.1. Iterative methods for NEP – the nonlinear equations view

Let us first consider the solution of (NEP), emphasizing the matrices we wish to approximate in a quasi-Newton approach. Most methods designed to solve (NEP) strive for a point x^* and a set of multipliers λ^* such that equations (1.2) hold. Consider the kth step of a Newton iteration, obtained by solving

$$d'_L(x_k, \lambda_k)\begin{pmatrix} s_k \\ \gamma_k \end{pmatrix} = \begin{bmatrix} \nabla_x d_L(x, \lambda)^{\mathrm{T}} \\ \nabla_\lambda d_L(x, \lambda)^{\mathrm{T}} \end{bmatrix}\begin{pmatrix} s_k \\ \gamma_k \end{pmatrix} = -d_L(x_k, \lambda_k),$$

where $s_k \doteq x_{k+1} - x_k$ and $\gamma_k \doteq \lambda_{k+1} - \lambda_k$. Dropping the subscript k, using the subscript $+$ for $k+1$, and the definition of $d_L(x_k, \lambda_k)$ we write the Newton iteration as

$$\begin{bmatrix} \nabla_{xx} L(x, \lambda) & -A(x) \\ A(x)^{\mathrm{T}} & 0 \end{bmatrix}\begin{pmatrix} s \\ \lambda_+ \end{pmatrix} = -\begin{pmatrix} g \\ c \end{pmatrix}. \tag{1.3}$$

If A is of full column rank and $\nabla_{xx} L(x, \lambda)$ is positive definite in the null-space of A^{T} then there exists a unique solution of system (1.3). Defining the orthogonal matrix $\bar{Q} \in \mathbb{R}^{(n+t)\times(n+t)}$,

$$\bar{Q} \doteq \begin{bmatrix} Y & Z & 0 \\ 0 & 0 & I_{t\times t} \end{bmatrix},$$

the solution to (1.3) may be rewritten by multiplying both sides by $\bar{Q}^{\mathrm{T}}$ and inserting $\bar{Q}\bar{Q}^{\mathrm{T}} = I$,

$$\bar{Q}^{\mathrm{T}}\begin{bmatrix} \nabla_{xx} L(x, \lambda) & -A(x) \\ A(x)^{\mathrm{T}} & 0 \end{bmatrix}\bar{Q}\bar{Q}^{\mathrm{T}}\begin{pmatrix} s \\ \lambda_+ \end{pmatrix} = -\bar{Q}^{\mathrm{T}}\begin{pmatrix} g \\ c \end{pmatrix}.$$

Dropping the arguments x and λ and carrying out the multiplication by $\bar{Q}$ we have

$$\begin{bmatrix} Y^{\mathrm{T}}\nabla^2 LY & Y^{\mathrm{T}}\nabla^2 LZ & -R \\ Z^{\mathrm{T}}\nabla^2 LY & Z^{\mathrm{T}}\nabla^2 LZ & 0 \\ R^{\mathrm{T}} & 0 & 0 \end{bmatrix}\begin{pmatrix} Y^{\mathrm{T}} s \\ Z^{\mathrm{T}} s \\ \lambda_+ \end{pmatrix} = -\begin{pmatrix} Y^{\mathrm{T}} g \\ Z^{\mathrm{T}} g \\ c \end{pmatrix} \tag{1.4}$$

where the shorthand $\nabla^2 L$ denotes the Hessian $\nabla_{xx} L(x, \lambda)$. This system is written using the four matrices $Y^{\mathrm{T}}\nabla^2 LY$, $Y^{\mathrm{T}}\nabla^2 LZ$, $Z^{\mathrm{T}}\nabla^2 LY$ and $Z^{\mathrm{T}}\nabla^2 LZ$ which we refer to as reduced or projected Hessian matrices. Together, the four blocks will be referred to as the partitioned Hessian matrix, and we call system (1.4) the partitioned or transformed Newton system. Since this system is in block triangular form, we can immediately write down its solution. Defining

$$s_Y \doteq Y^{\mathrm{T}} s, \qquad g_T \doteq Y^{\mathrm{T}} g, \qquad s_Z \doteq Z^{\mathrm{T}} s, \qquad g_Z \doteq Z^{\mathrm{T}} g,$$

we have the Newton step

$$s = Ys_Y + Zs_Z, \quad s_Y = -R^{-\mathrm{T}} c, \; s_Z = -(Z^{\mathrm{T}}\nabla^2 LZ)^{-1}(g_Z + Z^{\mathrm{T}}\nabla^2 LYs_Y),$$
$$\lambda_+ = R^{-1}(g_Y + Y^{\mathrm{T}}\nabla^2 LYs_Y + Y^{\mathrm{T}}\nabla^2 LZs_Z). \tag{1.5}$$

Note that we obtain the full step s as the sum of two orthogonal components $v = Ys_Y$ and $h = Zs_Z$. Since Zs_Z lies in the null space of the constraints, the horizontal step h tends to decrease $Z^{\mathrm{T}} g$ while moving tangent to the constraint contour. Similarly, since Ys_Y lies in the range space of the constraints, the vertical step v tends to reduce the constraint values.

Using equations (1.5), the Newton algorithm for (NEP) can be defined as follows:

Newton algorithm.

```
Initialize:
  $k \leftarrow 0$.
  Choose $x_0$, $\lambda_0$.
  Do while ($\|g_Z^k\| > \varepsilon_{\text{stop}}$ or $\|c_k\| > \varepsilon_{\text{stop}}$)
    Evaluate $f_k$, $g_k$, $\nabla^2 L_k$, $c_k$, $A_k$.
    Factor $A_k = [Y_k \ Z_k][R_k^{\mathrm{T}} \ 0^{\mathrm{T}}]^{\mathrm{T}}$.
    SOLVE (1.5) for $s_Y^k$, $s_Z^k$, and $\lambda_{k+1}$.
    $x_{k+1} \leftarrow x_k + v_k + h_k = x_k + Ys_Y^k + Zs_Z^k$;
    $k \leftarrow k+1$.
  End
```

Noting that system (1.3) can also be derived as a step in a Sequential Quadratic Programming (SQP) algorithm, we will refer to $(x_{k+1} - x_k, \lambda_{k+1} - \lambda_k)$ as an SQP step.

In the remainder of this paper we restrict our attention to quasi-Newton methods for (NEP). By approximating the transformed second derivative matrix $[Y \ Z]^{\mathrm{T}}\nabla^2 L[Y \ Z]$,

$$\begin{bmatrix} B_{YY} & B_{YZ} \\ B_{ZY} & B_{ZZ} \end{bmatrix} \sim \begin{bmatrix} Y^{\mathrm{T}}\nabla^2 LY & Y^{\mathrm{T}}\nabla^2 LZ \\ Z^{\mathrm{T}}\nabla^2 LY & Z^{\mathrm{T}}\nabla^2 LZ \end{bmatrix} \tag{1.6}$$

and substituting our approximation into (1.4) and (1.5) we obtain the quasi-Newton SQP step

$$s = Ys_Y + Zs_Z, \quad s_Y = -R^{-\mathrm{T}}c, \ s_Z = -B_{ZZ}^{-1}(g_Z + B_{ZY}s_Y),$$
$$\lambda_+ = \lambda^{\mathrm{QP}} = R^{-1}(g_Y + B_{YY}s_Y + B_{YZ}s_Z). \tag{1.7}$$

In the next section we discuss current quasi-Newton methods for (NEP), and show that they all lack certain desirable features. We remedy these deficiencies in Section 3 by deriving two new quasi-Newton updates with very desirable properties. In Section 4 we analyze the formulas derived for the two new updates. Finally, we present the results of our extensive computational experience with the two new updates in Section 5.

2. Current quasi-Newton methods for the nonlinear equality problem

Since quasi-Newton methods for (NEP) are quite complex, space does not permit a full discussion. Greater detail may be found in Fenyes (1987). Here, we concentrate on a major deficiency of current quasi-Newton methods for the constrained problem — simple, reliable methods for keeping the null-space Hessian approximation positive definite, particularly when far from the solution. In unconstrained methods, positivity of the Hessian approximation insures that the quasi-Newton step $s = -B^{-1}g$ will be a descent direction for the objective function at each iteration. In constrained methods, keeping at least B_{ZZ} positive definite is generally accepted to be desirable. Here, we prove that when B_{ZZ} is a positive definite approximation to $Z^{\mathrm{T}}\nabla^2 LZ$, the quasi-Newton SQP step (1.7) yields a well defined descent direction for the augmented Lagrangian merit function

$$L_{\mathrm{a}}(x, \lambda, \sigma) \doteq f(x) - c(x)^{\mathrm{T}}\lambda + \tfrac{1}{2}\sigma c(x)^{\mathrm{T}}c(x).$$

We examine two choices for the Lagrange multipliers in the definition of $L_{\mathrm{a}}: \lambda = \lambda^{\mathrm{QP}}$ where λ^{QP} is given above, and $\lambda = \lambda^{\mathrm{LS}}$ where λ^{LS} solves

$$\min_{\lambda} \|g - A\lambda\|_2^2.$$

By definition, the SQP step will be a descent direction for L_{a} if $-s^{\mathrm{T}}\nabla L_{\mathrm{a}} > 0$. Note that we use the simplified notation ∇L_{a} to mean $\nabla_x L_{\mathrm{a}}(x, \lambda, \sigma)$ with λ and σ held constant. Since $s = Ys_Y + Zs_Z$ and $QQ^{\mathrm{T}} = I$, we may write $s^{\mathrm{T}}\nabla L_{\mathrm{a}}$ as

$$s^{\mathrm{T}}QQ^{\mathrm{T}}\nabla L_{\mathrm{a}} = s^{\mathrm{T}}(YY^{\mathrm{T}} + ZZ^{\mathrm{T}})\nabla L_{\mathrm{a}} = s_Y^{\mathrm{T}}Y^{\mathrm{T}}\nabla L_{\mathrm{a}} + s_Z^{T}Z^{\mathrm{T}}\nabla L_{\mathrm{a}}.$$

Using $A = YR$ and the properties of L_{a}, Y, and Z yields

$$s^{\mathrm{T}}\nabla L_{\mathrm{a}} = s_Y^{\mathrm{T}}Y^{\mathrm{T}}(g - A\lambda) + s_Y^{\mathrm{T}}\sigma Rc + s_Z^{\mathrm{T}}g_Z. \tag{2.1}$$

Substituting c, g_Z and λ^{QP} from equations (1.7) into (2.1) and letting $L_{\mathrm{a}}^{\mathrm{QP}} = L_{\mathrm{a}}(x, \lambda^{\mathrm{QP}}, \sigma)$, we have

$$-s^{\mathrm{T}}\nabla L_{\mathrm{a}}^{\mathrm{QP}} = s_Y^{\mathrm{T}}B_{YY}s_Y + s_Y^{\mathrm{T}}B_{YZ}s_Z + s_Z^{\mathrm{T}}B_{ZY}s_Y + s_Z^{\mathrm{T}}B_{ZZ}s_Z + \sigma s_Y^{\mathrm{T}}RR^{\mathrm{T}}s_Y. \tag{2.2}$$

Or, using λ^{LS} and letting $L_{\mathrm{a}}^{\mathrm{LS}} = L_{\mathrm{a}}(x, \lambda^{\mathrm{LS}}, \sigma)$ we get

$$-s^{\mathrm{T}}\nabla L_{\mathrm{a}}^{\mathrm{LS}} = s_Z^{\mathrm{T}}B_{ZZ}s_Z + s_Z^{\mathrm{T}}B_{ZY}s_Y + \sigma s_Y^{\mathrm{T}}RR^{\mathrm{T}}s_Y. \tag{2.3}$$

To establish the sign of these quantities we use a result from Debreu (1952).

Lemma 2.1. *Define $C \in \mathbb{R}^{n\times t}$, $W \in \mathbb{R}^{n\times n}$, $z \in \mathbb{R}^n$. Then, $z^{\mathrm{T}}Wz > 0$ for every $z \neq 0$ such that $C^{\mathrm{T}}z = 0$ if and only if there exists a number $\bar{\alpha}$ such that $z^{\mathrm{T}}Wz + \alpha z^{\mathrm{T}}CC^{\mathrm{T}}z > 0$ for all $z \neq 0$ and $\alpha > \bar{\alpha}$.* □

Applying this result, we prove the following theorem.

Theorem 2.2. *Suppose s_Y and s_Z are given by (1.7) and $x \neq x^*$. If $s_Z \neq 0$, R is full rank, and B_{ZZ} is positive definite, then there exists a $\bar{\sigma}$ such that the SQP step $s = Ys_Y + Zs_Z$ is a descent direction for $L_{\mathrm{a}}(x, \lambda, \sigma)$ for either λ^{QP} or λ^{LS} and all $\sigma > \bar{\sigma}$.*

Proof. We will show $-s^{\mathrm{T}}\nabla L_{\mathrm{a}}^{\mathrm{QP}}$ and $-s^{\mathrm{T}}\nabla L_{\mathrm{a}}^{\mathrm{LS}}$ given by (2.2) and (2.3) are positive by applying Lemma 2.1. First choose $z^{\mathrm{T}} \doteq (s_Y^{\mathrm{T}}, s_Z^{\mathrm{T}})$ and set $C^{\mathrm{T}} = [R^{\mathrm{T}}\ 0^{\mathrm{T}}]$,

$$W = \begin{cases} \begin{bmatrix} B_{YY} & B_{YZ} \\ B_{ZY} & B_{ZZ} \end{bmatrix} & \text{for (2.2)}, \\ \begin{bmatrix} 0 & 0 \\ B_{ZY} & B_{ZZ} \end{bmatrix} & \text{for (2.3)}. \end{cases}$$

Since B_{ZZ} is positive definite and R has full rank we can assume s_Y and s_Z from (1.7) are well defined and of finite norm. In addition, $C^{\mathrm{T}}z = 0$ for $z \neq 0$ if and only if $s_Y = 0$ since R is of full rank. Thus, for all $s_Z \neq 0$ and $\bar{z}^{\mathrm{T}} \doteq (0^{\mathrm{T}}, s_Z^{\mathrm{T}})$, we have $C^{\mathrm{T}}\bar{z} = 0$ and $\bar{z}^{\mathrm{T}}W\bar{z} = s_Z^{\mathrm{T}}B_{ZZ}s_Z > 0$ since B_{ZZ} is positive definite. We conclude from Lemma 2.1 that there exists a $\bar{\sigma}$ such that $-s^{\mathrm{T}}\nabla L_{\mathrm{a}}(x, \lambda, \sigma) > 0$ for λ^{QP} or λ^{LS} and all $\sigma > \bar{\sigma}$. □

The modified SQP x-step

$$s = Ys_Y + Zs_Z, \quad s_Y = -R^{-\mathrm{T}}c, \ s_Z = -B_{ZZ}^{-1}g_Z, \tag{2.4}$$

used in many of the projected quasi-Newton methods discussed below, may also be shown to be a descent direction for L_{a} when B_{ZZ} is positive definite. In addition, a quasi-Newton step for the l_2-penalty function, $p \doteq f + \frac{1}{2}\sigma c^{\mathrm{T}}c$, is a descent direction for p if the null space portion of the Hessian approximation is positive definite (Fenyes, 1987).

Given this motivation, how do we maintain positivity? In unconstrained quasi-Newton methods, we can maintain a positive definite Hessian approximation when far from the solution by using a line-search procedure. For example, by requiring the line-search to satisfy the Goldstein–Armijo criteria (Goldstein, 1967; Armijo, 1966)

$$f_+ \leqslant f + \beta_1 g^{\mathrm{T}}(x_+ - x), \qquad g_+^{\mathrm{T}}(x_+ - x) \geqslant \beta_2 g^{\mathrm{T}}(x_+ - x), \tag{2.5}$$

with $0 < \beta_1 < \beta_2 < 1$, we can guarantee the existence of a positive definite update at each iteration — even far from the solution. In the constrained case, however, if we update using a step in the full space it is nearly impossible to guarantee positivity of the nullspace Hessian approximation by a simple line-search procedure.

To handle this, two basic quasi-Newton approaches for constrained problems have emerged. The earliest methods approximated the full Hessian matrix. Schittkowski (1984) and Tapia (1978) compared most of the existing algorithms, and showed that many are equivalent in the equality constrained case. Hestenes (1969), Murray (1969), Biggs (1972, 1975), Garcia et al. (1976), Han (1976), Tapia (1977), Glad (1979), Schittkowski (1981), Mayne and Polak (1982), Gill et al. (1985, 1986), and Powell and Yuan (1986) approximated the full Hessian of either the Lagrangian or the augmented Lagrangian. These methods have several drawbacks. First, they require the storage of the full $n \times n$ Hessian matrix though we need at most an $(n-m) \times n$ submatrix to solve for the step from x to x_+. Second, none of these methods can reliably assure that the Hessian approximation will remain well conditioned and positive definite at each iteration — even near the solution. In part, this may be due to incompatibilities between our chosen merit function and the function whose Hessian we are approximating (Boggs and Tolle, 1987). Powell (1978a) overcame this difficulty by introducing a modified BFGS formula which keeps the entire Hessian approximation positive definite. Unfortunately Powell's modification may generate very ill-conditioned updates (Powell, 1985).

Projected methods first suggested by Wright (1976) and Murray and Wright (1978), and more recently by Tanabe (1981), Womersley and Fletcher (1982), Gabay (1982), Coleman and Conn (1984), Nocedal and Overton (1985) and Gurwitz (1986), approximate the projected Hessian $Z^T\nabla^2LZ$, and either approximate the quantity $Z^T\nabla^2LYs_Y$ by finite differences or ignore it. Consequently, we solve smaller $(n-m)\times(n-m)$ systems and never store or project the full Hessian matrix. Since the matrix $Z^T\nabla^2LZ$ is assumed positive definite in a neighborhood of the solution these approaches attempt to maintain positivity of the null-space approximation near the solution without introducing a penalty parameter. However, some difficulties remain.

In the methods of Womersley and Fletcher (1982) and Gabay (1982), projections of the full-step quantities

$$s=Z^T(x_+-x), \qquad y=Z^T(\nabla L(x_+)-\nabla L(x)), \tag{2.6}$$

are used to update an approximation to $Z^T\nabla^2LZ$. Since this choice can not guarantee $s^Ty>0$ they use Powell's modified BFGS update to keep the reduced Hessian approximation positive definite. As noted by Nocedal and Overton (1985), this modification may lead to ill-conditioned updates if $s^Ty\gg s^TBs$ even when B approximates the reduced Hessian.

The projected methods of Coleman and Conn (1984) and Fontecilla (1988) have no penalty parameters or modified updates. They maintain $B_{ZZ}>0$ by evaluating one extra gradient, $\nabla L(x+h)$, per iteration. Using the projected quantities $s=Z^T(x_+-x)$, $y=Z^T(\nabla L(x+h)-\nabla L(x))$ and the quasi-Newton equation $B^+_{ZZ}s_Z=Z^T(\nabla L(x+h)-\nabla L(x))$, they compute either a BFGS or DFP update. This choice insures that $s^Ty>0$ for x near x^* so that B_{ZZ} will remain positive definite near the solution. Unfortunately, when far from the solution, even this approach can not guarantee positivity.

Nocedal and Overton (1985) use the projected full-step quantities (2.6). They guarantee positivity of the null space portion of the Hessian by updating at iteration k only when

$$\|v_k\|<\frac{\eta}{(k+1)^{1+\phi}}\|h_k\|.$$

Here, η and ϕ are arbitrary parameters which must be properly selected for good performance. They prove that this restriction guarantees $s^Ty>0$ in a region very near x^*. However, their result assures frequent updating only very near the solution and may have limited practical computational value.

3. Derivation of new quasi-Newton updates for constrained optimization using a variational approach

Let us now consider approximating the transformed Hessian matrix (1.6), while maintaining positivity of the null-space approximation, a least change condition, and the transformed quasi-Newton equation $Q^{\mathrm{T}}(B_{+})QQ^{\mathrm{T}}s = Q^{\mathrm{T}}y$, or in partitioned form

$$\begin{bmatrix} B^{+}_{YY} & B^{+}_{YZ} \\ B^{+}_{ZY} & B^{+}_{ZZ} \end{bmatrix}\begin{pmatrix} s_Y \\ s_Z \end{pmatrix} = \begin{pmatrix} y_Y \\ y_Z \end{pmatrix} \doteq \begin{pmatrix} Y^{\mathrm{T}}(\nabla L(x+s)-\nabla L(x)) \\ Z^{\mathrm{T}}(\nabla L(x+s)-\nabla L(x)) \end{pmatrix}. \tag{3.1}$$

Note that $y \doteq \nabla L(x+s)+\nabla L(x)$ requires gradient information only at x and $x_{+} = x+s$ — no extra midpoint gradient evaluations are required.

3.1. The lower partition BFGS update

Since the matrices $Y^{\mathrm{T}}\nabla^2 LY$ and $Y^{\mathrm{T}}\nabla^2 LZ$ are required only to solve for the new Lagrangian multipliers, if we obtain the multipliers by a least squares estimate we need approximate only the terms $Z^{\mathrm{T}}\nabla^2 LY$ and $Z^{\mathrm{T}}\nabla^2 LZ$, the lower half of the partitioned Hessian matrix (1.6). We will approximate $Z^{\mathrm{T}}\nabla^2 LZ$ by a positive definite, symmetric matrix B_{ZZ} and $Z^{\mathrm{T}}\nabla^2 LY$ by a matrix B_{ZY} and require that B^{+}_{ZY} and B^{+}_{ZZ} satisfy only the lower half of the quasi-Newton equation (3.1),

$$[B^{+}_{ZY} \quad B^{+}_{ZZ}]\begin{pmatrix} s_Y \\ s_Z \end{pmatrix} = y_Z. \tag{3.2}$$

As in Dennis and Schnabel's (1979) derivation for unconstrained updates, we assume B_{ZZ} is positive definite and symmetric with a Cholesky decomposition $B_{ZZ} = LL^{\mathrm{T}}$, and then define

$$B^{+}_{ZZ} \doteq (L+E)(L+E)^{\mathrm{T}} = J_{+}J_{+}^{\mathrm{T}}, \tag{3.3}$$

$$B^{+}_{ZY} \doteq B_{ZY} + H, \tag{3.4}$$

where $H \in \mathbb{R}^{(n-t)\times t}$, $E \in \mathbb{R}^{(n-t)\times(n-t)}$. Substituting these expressions into the quasi-Newton condition, (3.2), and rearranging, we obtain

$$[H \quad (L+E)(L+E)^{\mathrm{T}}]\begin{pmatrix} s_Y \\ s_Z \end{pmatrix} = y_Z - B_{ZY}s_Y \doteq r_Z,$$

where $r_Z \in \mathbb{R}^{(n-t)}$. Defining $w \doteq (L+E)^{\mathrm{T}}s_Z$, $w \in \mathbb{R}^{(n-t)}$, our complete variational formulation is to choose H and E in order to

$$\underset{h,\, e}{\text{minimize}} \quad \tfrac{1}{2}\|[H \quad E]\|_{\mathrm{F}}^{2}$$

$$\text{subject to} \quad [H \quad (L+E)]\begin{pmatrix} s_Y \\ w \end{pmatrix} = r_Z. \tag{3.5}$$

Thus, we have an equality constrained QP in the variables h_{ij} and e_{ij}, the entries of H and E.

Applying the first order optimality conditions to this QP, we determine

$$H = us_Y^{\mathrm{T}} \tag{3.6}$$

and $E = uw^{\mathrm{T}}$ where $u \in \mathbb{R}^{(n-t)}$ is the vector of Lagrange multipliers. Substituting E into the definition of w we get $w = L^{\mathrm{T}} s_Z/(1 - s_Z^{\mathrm{T}} u)$ and thus

$$E = \frac{us_Z^{\mathrm{T}} L}{1 - s_Z^{\mathrm{T}} u}. \tag{3.7}$$

Substituting H and E from (3.6) and (3.7) into the quasi-Newton equation (3.5), noting $B_{ZZ} = LL^{\mathrm{T}}$ and defining $\alpha \doteq 1/(1 - u^{\mathrm{T}} s_Z)$ we may write u as a function of the unknown scalar α,

$$u = \frac{r_Z - \alpha B_{ZZ} s_Z}{s_Y^{\mathrm{T}} s_Y + \alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z}. \tag{3.8}$$

Using this expression for u we form $\alpha \doteq 1/(1 - u^{\mathrm{T}} s_Z)$ and, noting that $s_Z^{\mathrm{T}} B_{ZZ} s_Z > 0$, collect terms in α to obtain the cubic equation

$$\alpha^3 + \alpha \frac{(s_Y^{\mathrm{T}} s_Y - s_Z^{\mathrm{T}} r_Z)}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} - \frac{s_Y^{\mathrm{T}} s_Y}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} = 0. \tag{3.9}$$

Thus, given a solution for this cubic in α, we can determine u, H and E from equations (3.8), (3.6) and (3.7).

In a variational derivation of the standard BFGS formula, α must solve $\alpha^2 - (s^{\mathrm{T}} y / s^{\mathrm{T}} Bs) = 0$, making necessary the familiar condition $s^{\mathrm{T}} y > 0$ in order to insure a well-defined update. In contrast, since equation (3.9) is a cubic, it always has at least one real root. Thus, it seems, we are guaranteed that α, u, H and E will be well defined under all circumstances. The next theorem shows that this generally holds for the Lower Partition BFGS Update (LPB). However, as discussed below, special circumstances require a more detailed analysis of the cubic root, α.

Theorem 3.1. *If $s_Y \neq 0$, $s_Z \neq 0$, and $B_{ZZ} = LL^{\mathrm{T}}$ is positive definite, then the updated matrices B_{ZY}^+ and B_{ZZ}^+, given by equations* (3.3), (3.4) *and* (3.6)–(3.9) *are well defined and satisfy the quasi-Newton equation* (3.2). *In addition, B_{ZZ}^+ is positive definite and symmetric.*

Proof. B_{ZY}^+ and B_{ZZ}^+ are well defined, since a real solution to the cubic equation (3.9) always exists under the stated assumptions. By definition, they clearly satisfy the quasi-Newton equation (3.2) and insure symmetry of B_{ZZ}^+.

To see that $B_{ZZ}^+ = J_+ J_+^{\mathrm{T}}$ is positive definite we must show that $J_+^{\mathrm{T}} = L^{\mathrm{T}} + E^{\mathrm{T}} = L^{\mathrm{T}}(I + \alpha s_Z u^{\mathrm{T}})$ is nonsingular. Since L is nonsingular, we consider the matrix $I + \alpha s_Z u^{\mathrm{T}}$. By the Sherman–Morrison–Woodbury formula this is nonsingular if and only if $\sigma \neq 0$, where

$$\sigma = 1 + \frac{s_Z^{\mathrm{T}} u}{1 - s_Z^{\mathrm{T}} u} = \frac{1}{1 - s_Z^{\mathrm{T}} u} \doteq \alpha.$$

But, clearly, $\alpha \neq 0$ since $\bar{\alpha} = 0$ is a real root of

$$\alpha^3 + \alpha \frac{(s_Y^{\mathrm{T}} s_Y - s_Z^{\mathrm{T}} r_Z)}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} - \frac{s_Y^{\mathrm{T}} s_Y}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} = 0.$$

if and only if $s_Y = 0$. Since $s_Y \neq 0$, $s_Z \neq 0$ and B_{ZZ} is positive definite by assumption, the cubic is well defined and $\bar{\alpha} \neq 0$. Thus, we have $\sigma \neq 0$ and J_+^{T} is nonsingular. □

Before we continue with two special cases, let us rewrite the update of B_{ZZ} in a form resembling the standard BFGS update. We can show

$$B_{ZZ}^+ = B_{ZZ} + \frac{pp^{\mathrm{T}}}{p^{\mathrm{T}} s_Z} - \frac{B_{ZZ} s_Z s_Z^{\mathrm{T}} B_{ZZ}^{\mathrm{T}}}{s_Z^{\mathrm{T}} B_{ZZ}^{\mathrm{T}} s_Z} \tag{3.10}$$

where

$$p = (1-\theta)\alpha B_{ZZ} s_Z + \theta r_Z, \quad p^{\mathrm{T}} s_Z = \alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z, \quad \theta = \frac{\alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z}{s_Y^{\mathrm{T}} s_Y + \alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z}. \tag{3.11}$$

In the special case $s_Y = 0$, we find the cubic degenerates to a "quadratic" equation and $\alpha = (s_Z^{\mathrm{T}} y_Z / s_Z^{\mathrm{T}} B_{ZZ} s_Z)^{1/2}$. In addition, $\theta = 1$, and $p = y_Z$ which leads to the following corollary showing the LPB is naturally equivalent to a projected BFGS update of the matrix B_{ZZ} which is well defined if $s_Z^{\mathrm{T}} y_Z > 0$.

Corollary 3.2. *Assume $s_Y = 0$ and $s_Z^{\mathrm{T}} r_Z = s_Z^{\mathrm{T}} y_Z > 0$ which implies $s_Z \neq 0$. As before, assume $B_{ZZ} = LL^{\mathrm{T}}$ is positive definite. Then the updated matrices B_{ZY}^+ and B_{ZZ}^+, given by equations* (3.3), (3.4) *and* (3.6)-(3.9) *are well defined and satisfy the quasi-Newton equation* (3.2). *In addition, $B_{ZY}^+ = B_{ZY}$, and B_{ZZ}^+ is given by the projected BFGS update*

$$B_{ZZ}^+ = B_{ZZ} + \frac{y_Z y_Z^{\mathrm{T}}}{s_Z^{\mathrm{T}} y_Z} - \frac{B_{ZZ} s_Z s_Z^{\mathrm{T}} B_{ZZ}^{\mathrm{T}}}{s_Z^{\mathrm{T}} B_{ZZ} s_Z}. \tag{3.12}$$

which is symmetric and positive definite. □

We state, without proof, the next corollary which shows that new information may be incorporated into the Hessian approximation even when $s_Z = 0$.

Corollary 3.3. *If $s_Z = 0$ and $s_Y \neq 0$ then $B_{ZZ}^+ = B_{ZZ}$. Furthermore, B_{ZY}^+, given by the projected Broyden update*

$$B_{ZY}^+ = B_{ZY} + \frac{(y_Z - B_{ZY} s_Y) s_Y^{\mathrm{T}}}{s_Y^{\mathrm{T}} s_Y}, \tag{3.13}$$

is well defined and satisfies the quasi-Newton equation. □

For some algorithms, (e.g. transformation methods based on the l_2-penalty function), we may wish to approximate all four projections of the Hessian matrix. It is possible to use the LPB in a two-state updating procedure to approximate $Y^T\nabla^2LY$ and $Y^T\nabla^2LZ$ in addition to $Z^T\nabla^2\mathrm{LZ}$ and $Z^T\nabla^2LY$. Details of this update may be found in Coleman and Fenyes (1988). However, in the next section we define a new update formula which simultaneously updates the full partitioned Hessian matrix.

3.2. The full partition BFGS update

Returning to the partitioned form of NEP, consider approximating of all four matrices $Y^T\nabla^2LY$, $Y^T\nabla^2LZ$, $Z^T\nabla^2LY$ and $Z^T\nabla^2LZ$. Again, approximate $Z^T\nabla^2LZ$ by a positive definite, symmetric matrix B_{ZZ}. To maintain symmetry, we require B_{YY} symmetric and $B_{YZ} = B_{ZY}^T$. We also require that B_{YY}^+, B_{YZ}^+, B_{ZY}^+ and B_{ZZ}^+ satisfy the full projected quasi-Newton equation (3.1). Again, assume the current approximation $B_{ZZ} = LL^T$ is positive definite and symmetric and then define

$$B_{ZZ}^+ = (L+E)(L+E)^T = J_+J_+^T, \tag{3.14}$$

$$B_{ZY}^+ = B_{ZY} + H = B_{YZ}^{+T}, \tag{3.15}$$

$$B_{YY}^+ = B_{YY} + (G+G^T). \tag{3.16}$$

Defining $w \doteq (L+E)^Ts_Z$, $w \in \mathbb{R}^{(n-t)}$, and assuming $L+E$ nonsingular, the quasi-Newton condition, (3.1), may be written

$$\begin{bmatrix} (G+G^T) & H^T & 0 \\ H & 0 & (L+E) \end{bmatrix} \begin{pmatrix} s_Y \\ s_Z \\ w \end{pmatrix} = \begin{pmatrix} y_Y - B_{YY}s_Y - B_{YZ}s_Z \\ y_Z - B_{ZY}s_Y \end{pmatrix} \doteq \begin{pmatrix} r_Y \\ r_Z \end{pmatrix}. \tag{3.17}$$

where $r_Y \in \mathbb{R}^t$ and $r_Z \in \mathbb{R}^{(n-t)}$. The variational formulation is completed by choosing G, H and E to

$$\underset{g,h,e}{\text{minimize}} \ \frac{1}{2} \left\| \begin{bmatrix} (G+G^T) & H^T \\ H & E \end{bmatrix} \right\|_F^2$$

subject to the quasi-Newton condition (3.17).

Forming the Lagrangian in the variables g_{ij}, h_{ij} and e_{ij} and two Lagrange multipliers $u_Y \in \mathbb{R}^t$ and $u_Z \in \mathbb{R}^{(n-t)}$, differentiating, and applying the first order conditions, we obtain

$$G + G^T = u_Ys_Y^T + s_Yu_Y^T, \tag{3.18}$$

$$\begin{aligned} H &= u_Zs_Y^T + s_Zu_Y^T, \\ E &= 2u_Zw^T. \end{aligned} \tag{3.19}$$

Using E in the definition of w we get $w = L^Ts_Z/(1-2s_Z^Tu_Z)$ and

$$E = \frac{2u_Zs_Z^TL}{1-2s_Z^Tu_Z}. \tag{3.20}$$

Substituting H, $G+G^{\mathrm{T}}$ and E from (3.18), (3.19) and (3.20) into the quasi-Newton equation (3.17) yields expressions for u_Y and u_Z in terms of the unknown scalars $u_Z^{\mathrm{T}}s_Z$ and $u_Y^{\mathrm{T}}s_Y$;

$$u_Y = \frac{r_Y - s_Y(u_Y^{\mathrm{T}}s_Y + u_Z^{\mathrm{T}}s_Z)}{s^{\mathrm{T}}s}, \tag{3.21}$$

$$u_Z = \frac{r_Z - s_Z u_Y^{\mathrm{T}}s_Y - \alpha B_{ZZ}s_Z}{s_Y^{\mathrm{T}}s_Y + 2\alpha^2 s_Z^{\mathrm{T}}B_{ZZ}s_Z}, \tag{3.22}$$

where $s^{\mathrm{T}}s \doteq s_Y^{\mathrm{T}}s_Y + s_Z^{\mathrm{T}}s_Z$ and $\alpha \doteq 1/(1-2s_Z^{\mathrm{T}}u_Z)$. Multiplying (3.21) by s_Y^{T} we solve for $u_Y^{\mathrm{T}}s_Y \doteq \gamma$ as a function of $u_Z^{\mathrm{T}}s_Z$,

$$\gamma \doteq u_Y^{\mathrm{T}}s_Y = \frac{s_Y^{\mathrm{T}}r_Y - s_Y^{\mathrm{T}}s_Y u_Z^{\mathrm{T}}s_Z}{s_Y^{\mathrm{T}}s_Y + s^{\mathrm{T}}s}. \tag{3.23}$$

Substituting γ into (3.22), multiplying both sides by $-2s_Z^{\mathrm{T}}$, adding 1 to both sides and simplifying, gives the cubic

$$\alpha^3 + \alpha\eta - \nu = 0 \tag{3.24}$$

where $\alpha = 1/(1-2s_Z^{\mathrm{T}}u_Z)$ and

$$\eta \doteq \frac{s_Y^{\mathrm{T}}s_Y(s_Y^{\mathrm{T}}s_Y - 2s_Z^{\mathrm{T}}r_Z) + s_Z^{\mathrm{T}}s_Z(s_Y^{\mathrm{T}}r_Y - s_Z^{\mathrm{T}}r_Z)}{s_Z^{\mathrm{T}}B_{ZZ}s_Z(s_Y^{\mathrm{T}}s_Y + s^{\mathrm{T}}s)},$$

$$\nu \doteq \frac{(s_Y^{\mathrm{T}}s_Y)^2}{s_Z^{\mathrm{T}}B_{ZZ}s_Z(s_Y^{\mathrm{T}}s_Y + s^{\mathrm{T}}s)}.$$

Given a solution for this cubic, we can compute $u_Z^{\mathrm{T}}s_Z$ from α and then γ, u_Y, u_Z, $G+G^{\mathrm{T}}$, H and E from equations (3.23), (3.21), (3.22), (3.18)-(3.20).

As we found for the LPB, this cubic will have at least one real root when $s_Z \neq 0$. Carrying out an analysis similar to that for the LPB yields an analog of Theorem 3.1 showing the Full Partition BFGS Update (FPB) is well defined when $s_Y \neq 0$ and $s_Z \neq 0$. The proof follows that of Theorem 3.1 and is omitted.

Theorem 3.4. *If $s_Y \neq 0$, $s_Z \neq 0$ and $B_{ZZ} = LL^{\mathrm{T}}$ is positive definite, then the updated matrices B_{YY}^+, B_{YZ}^+, B_{ZY}^+, and B_{ZZ}^+, given by equations* (3.14)-(3.16), *and* (3.18)-(3.24) *are well defined and satisfy the quasi-Newton equation* (3.17). *In addition, B_{ZZ}^+ is positive definite and symmetric.* □

Analogs of Corollaries 3.2 and 3.3 show the FPB is naturally equivalent to projected versions of standard updates when either $s_Y = 0$ or $s_Z = 0$.

Corollary 3.5. *Assume now $s_Y = 0$ and $s_Z^{\mathrm{T}}r_Z = s_Z^{\mathrm{T}}y_Z > 0$ which implies $s_Z \neq 0$. As before, also assume $B_{ZZ} = LL^{\mathrm{T}}$ is positive definite. Then the updated matrices B_{YY}^+, B_{YZ}^+, B_{ZY}^+, and B_{ZZ}^+, given by equations* (3.14)-(3.16) *and* (3.18)-(3.24) *are well defined* *and satisfy the full quasi-Newton equation* (3.1). *In addition,* $B^+_{YY} = B_{YY}$, *and* $B^+_{YZ} = B^{+\mathrm{T}}_{ZY}$ *is given by the projected Broyden formula*

$$B^+_{YZ} = B_{YZ} + \frac{(y_Y - B_{YZ}s_Z)s_Z^{\mathrm{T}}}{s_Z^{\mathrm{T}}s_Z}.$$

Furthermore, B^+_{ZZ}, *given by the projected BFGS formula* (3.12), *is symmetric and positive definite.* □

Corollary 3.6. *If* $s_Z = 0$ *and* $s_Y \neq 0$ *then the updated matrices are well defined and satisfy the full quasi-Newton equation. In addition,* $B^+_{ZZ} = B_{ZZ}$, *and* $B^+_{ZY} = B^{+\mathrm{T}}_{YZ}$ *is given by the projected Broyden formula,* (3.13) *while* B^+_{YY} *is given by the projected PSB,*

$$B^+_{YY} = B_{YY} + \frac{(y_Y - B_{YY}s_Y)s_Y^{\mathrm{T}} + s_Y(y_Y - B_{YY}s_Y)^{\mathrm{T}}}{s_Y^{\mathrm{T}}s_Y} - \frac{s_Y^{\mathrm{T}}(y_Y - B_{YY}s_Y)s_Ys_Y^{\mathrm{T}}}{(s_Y^{\mathrm{T}}s_Y)^2}. \quad \square$$

Again, B^+_{ZZ} may be written in the standard form (3.10) where α is given by (3.24) and

$$p = (1-\theta)\alpha B_{ZZ}s_Z + \theta(r_Z - \gamma s_Z), \quad p^{\mathrm{T}}s_Z = \alpha^2 s_Z^{\mathrm{T}}B_{ZZ}s_Z,$$

$$\theta = \frac{2\alpha^2 s_Z^{\mathrm{T}}B_{ZZ}s_Z}{s_Y^{\mathrm{T}}s_Y + 2\alpha^2 s_Z^{\mathrm{T}}B_{ZZ}s_Z}.$$

4. Analysis of the cubic equations in the LPB and FPB

The cubic equations (3.9) and (3.24) must be solved to compute either the LPB or FPB update. In the standard form,

$$c(\alpha) \doteq \alpha^3 + \pi\alpha^2 + \eta\alpha + \nu,$$

we have for the FPB:

(c-FPB) $\pi = 0$,

$$\eta = \frac{s_Y^{\mathrm{T}}s_Y(s_Y^{\mathrm{T}}s_Y - 2s_Z^{\mathrm{T}}r_Z) + s_Z^{\mathrm{T}}s_Z(s_Y^{\mathrm{T}}r_Y - s_Z^{\mathrm{T}}r_Z)}{s_Z^{\mathrm{T}}B_{ZZ}s_Z(s_Y^{\mathrm{T}}s_Y + s^{\mathrm{T}}s)},$$

$$\nu = -\frac{(s_Y^{\mathrm{T}}s_Y)^2}{s_Z^{\mathrm{T}}B_{ZZ}s_Z(s_Y^{\mathrm{T}}s_Y + s^{\mathrm{T}}s)},$$

and for the LPB:

(c-LPB) $\pi = 0$,

$$\eta = \frac{(s_Y^{\mathrm{T}}s_Y - s_Z^{\mathrm{T}}r_Z)}{s_Z^{\mathrm{T}}B_{ZZ}s_Z},$$

$$\nu = -\frac{s_Y^{\mathrm{T}}s_Y}{s_Z^{\mathrm{T}}B_{ZZ}s_Z},$$

with $s^{\mathrm{T}}s \doteq s_Y^{\mathrm{T}}s_Y + s_Z^{\mathrm{T}}s_Z$.

The analytical solution (Abramowitz and Stegun, 1964) yields the three roots

$$\begin{aligned}&\alpha_1 = A+B, \quad \alpha_2 = -\tfrac{1}{2}(A+B)+\tfrac{1}{2}\sqrt{-3}\,(A-B),\\ &\alpha_3 = -\tfrac{1}{2}(A+B)-\tfrac{1}{2}\sqrt{-3}\,(A-B),\end{aligned} \tag{4.1}$$

where

$$A, B = [-\tfrac{1}{2}\nu \pm (\omega)^{1/2}]^{1/3} \tag{4.2}$$

and

$$\omega = \tfrac{1}{27}\eta^3 + \tfrac{1}{4}\nu^2. \tag{4.3}$$

The number of real roots can be determined from the sign of ω as follows:

$\omega > 0 \Rightarrow$ 1 real root, 2 complex conjugate roots,

$\omega = 0 \Rightarrow$ 3 real roots, at least 2 equal,

$\omega < 0 \Rightarrow$ 3 real distinct roots.

Note that the cubic always has at least one real root, guaranteeing existence of the new updates in all cases. Although the formulas above may be used directly, we used safeguarded formulas and iterative improvement (Fenyes, 1987) to accurately evaluate the cubic roots.

4.1. Root selection and rejection

When three real roots are available, one must be selected to define the update. As shown in Coleman and Fenyes (1988), the three real cubic roots obtained when $\omega \leqslant 0$ may be written

$$\alpha_1 = 2\,\mathrm{Re}(A), \quad \alpha_2 = -\mathrm{Re}(A) - \sqrt{3}\,\mathrm{Im}(A), \quad \alpha_3 = -\mathrm{Re}(A) + \sqrt{3}\,\mathrm{Im}(A),$$

In general, these roots are well behaved. As discussed, when $s_Y = 0$ the cubic updates reduce to the standard projected BFGS update for B_{ZZ}. Thus, as $s_Y \to 0$ it is important that the selected root smoothly reduces the LPB and FPB to the standard projected formula. Specifically, as shown in Coleman and Fenyes (1988), this can be achieved by selecting either α_2 or α_3 which smoothly approach the roots $\alpha_{\mathrm{BFGS}} = \pm(s_Z^{\mathrm{T}} y_Z / s_Z^{\mathrm{T}} B_{ZZ} s_Z)^{1/2}$ for the standard projected BFGS update. We reject α_1 which approaches the unusable root, zero. Unlike the standard BFGS update, the roots α_2 and α_3 do not give the same update. Also, there is no apparent theoretical justification for selecting one root over the other. After extensive experimentation the following selection rule was adopted. First, since we wish the roots to be near the standard BFGS roots, when the quantity $s_Z^{\mathrm{T}} r_Z$ is positive and s_Y is small relative to s_Z we select the root whose absolute value is nearest to $+(s_Z^{\mathrm{T}} r_Z / s_Z^{\mathrm{T}} B_{ZZ} s_Z)^{1/2}$. We consider s_Y small relative to s_Z when $|\nu/\eta| < 0.10$ since the Taylor expansions given in Coleman and Fenyes (1988) indicate that the cubic roots will be near the roots α_{BFGS} in this case. In all other cases, we use the same rule used in the standard

BFGS update: choose the root which makes $\|E\|_2 = \|u\|_2 \|w\|_2$ as small as possible. Here, w_{LPB}, $w_{\mathrm{FPB}} = \alpha L^{\mathrm{T}} s_Z$ and

$$u_{\mathrm{LPB}} = \frac{r_Z - \alpha B_{ZZ} s_Z}{s_Y^{\mathrm{T}} s_Y + \alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z},$$

$$u_{\mathrm{FPB}} = \frac{2(r_Z - \alpha B_{ZZ} s_Z - \gamma s_Z)}{s_Y^{\mathrm{T}} s_Y + 2\alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z}, \quad \gamma = \frac{s_Y^{\mathrm{T}} r_Y - s_Y^{\mathrm{T}} s_Y (\alpha - 1)/2\alpha}{s_Y^{\mathrm{T}} s_Y + s^{\mathrm{T}} s}.$$

When $\omega \cong 0$, all three roots approach zero. This corresponds directly to failure of the standard BFGS update when $s^{\mathrm{T}} y \cong 0$. A similar situation exists when $s_Y \cong 0$ and $\omega > 0$. In this case, the cubic generates only one real root which approaches zero, and two complex roots corresponding to the roots obtained in the standard BFGS when $s^{\mathrm{T}} y < 0$. Updating with a root near zero generally results in an ill-conditioned update to B_{ZZ} and must be avoided. To see this, we note that (3.10) yields $s_Z^{\mathrm{T}} B_{ZZ}^{+} = \alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z$ which implies that as $\alpha \to 0$, B_{ZZ}^{+} approaches singularity. To avoid ill-conditioning, it may be necessary to reject the cubic roots near zero and skip the update. Fortunately we can predict this situation by estimating the condition number of the new matrix B_{ZZ}^{+} *before* updating. Defining the condition number of a symmetric matrix A as $K(A) \doteq \lambda_{\max}/\lambda_{\min}$, we estimate $\lambda_{\max}$ and $\lambda_{\min}$, the largest and smallest eigenvalues of A which satisfy the relations $\max(\lambda \mid Ax = \lambda x,\ \|x\| = 1)$ and $\min(\lambda \mid Ax = \lambda x,\ \|x\| = 1)$ respectively. Thus, for all x we have $x^{\mathrm{T}} Ax / x^{\mathrm{T}} x \leqslant \lambda_{\max}$ and $x^{\mathrm{T}} Ax / x^{\mathrm{T}} x \geqslant \lambda_{\min}$.

Consider the eigenvalues of B_{ZZ}^{*}. For small α, $\lambda_{\min}$ may be closely approximated by

$$\tilde{\lambda}_{\min} \doteq \frac{s_Z^{\mathrm{T}} B_{ZZ}^{+} s_Z}{\|s_Z\|^2} = \frac{\alpha^2 s_Z^{\mathrm{T}} B_{ZZ} s_Z}{\|s_Z\|^2} \geqslant \lambda_{\min}.$$

Also, using $B_{ZZ} = LDL^{\mathrm{T}}$ and the definition of $\lambda_{\max}$ we have for all x,

$$\lambda_{\max} = \|B_{ZZ}^{+}\| \geqslant \frac{x^{\mathrm{T}} B_{ZZ}^{+} x}{x^{\mathrm{T}} x} = \frac{1}{\|x\|^2} x^{\mathrm{T}} \left(LDL^{\mathrm{T}} + \frac{pp^{\mathrm{T}}}{p^{\mathrm{T}} s_Z} - \frac{B_{ZZ} s_Z s_Z^{\mathrm{T}} B_{ZZ}}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} \right) x,$$

and thus, for the best estimate, we seek x to maximize the right hand side. If d_i is the largest entry in the diagonal matrix D, we can estimate the desired x by l_{*i}: column i of the matrix L. Thus, an estimate of the largest eigenvalue of B_{ZZ}^{+} is given by

$$\lambda_{\max} \doteq \frac{1}{\|l_{*i}\|^2} l_{*i}^{\mathrm{T}} \left(LDL^{\mathrm{T}} + \frac{pp^{\mathrm{T}}}{p^{\mathrm{T}} s_Z} - \frac{B_{ZZ} s_Z s_Z^{\mathrm{T}} B_{ZZ}}{s_Z^{\mathrm{T}} B_{ZZ} s_Z} \right) l_{*i} \leqslant \lambda_{\max}.$$

Now, define $\tilde{K}$ as an estimate of the condition number,

$$\tilde{K} \doteq \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} \leqslant \frac{\lambda_{\max}}{\lambda_{\min}} \doteq K(B_{ZZ}^{+}).$$

Since we are most concerned about the case $\alpha \to 0$ and B_{ZZ}^{+} nearly singular, the estimate of $\lambda_{\min}$ is the most important component of the condition number estimate.

Fortunately the estimate of $\lambda_{\min}$ will be fairly accurate when B^{+}_{ZZ} is almost singular, since the singularity occurs in a direction nearly parallel to s_Z. This can easily be seen by considering (3.10) which yields $B^{+}_{ZZ}s_Z = p$. Since $p \to 0$ as $\alpha \to 0$, B^{+}_{ZZ} approaches singularity in the direction s_Z.

A more accurate condition number estimate may be obtained after updating by using the LINPACK (Dongarra et al., 1979) algorithm DCHCO or equivalent. Of course, this requires extra storage and computation. However, to confirm that $\tilde{K}$ is a reasonable estimate, it was compared to the condition number estimate computed by DCHCO. At all iterations, our estimate $\tilde{K}$ was found to be within an order of magnitude of the estimate obtained from the Cholesky decomposition. We concluded that $\tilde{K}$ was a simple and effective predictor of ill-conditioning for our numerical experiments. Other predictors have been used to avoid ill-conditioned approximations. For example, Gurwitz (1986) presented a simple post-update strategy which uses the diagonal terms of the updated Cholesky decomposition to estimate the condition number. Condition number estimates could also be used with Powell's modified BFGS to avoid ill-conditioning, however, this was not considered in our current study.

5. Computational testing

The LPB and FPB formulas were tested in two algorithms based on the projected form of the SQP step. They were compared with a standard projected-BFGS update formula used in nonlinearly constrained optimization (Coleman and Conn, 1984). These updates approximate only the null-space projection of the full Hessian matrix using an extra midpoint gradient evaluation to compute y_Z, the change in projected gradients due to the step in the null space. Letting B_{ZZ} represent the current approximation to the null-space projection of the Hessian, we define the BFGS-PM (Projected Midpoint) update by

$$\text{(BFGS-PM)}\quad s_Z = Z^{\mathrm{T}}(h+\nu) = Z^{\mathrm{T}}h,$$

$$y_Z = Z^{\mathrm{T}}(\nabla L(x+h+\nu) - \nabla L(x+\nu)),$$

$$B^{+}_{ZZ} = B_{ZZ} + \frac{y_Z y_Z^{\mathrm{T}}}{s_Z^{\mathrm{T}} y_Z} - \frac{B_{ZZ} s_Z s_Z^{\mathrm{T}} B_{ZZ}}{s_Z^{\mathrm{T}} B_{ZZ} s_Z},$$

where, again, h is a step in the null space and ν is a step in the range space. Differencing ∇L along h to compute y_Z guarantees $s_Z^{\mathrm{T}} y_Z > 0$ in a neighbourhood of the solution and also yields more accurate Hessian approximations. Since this update performs well near x^*, we use it as a standard to measure the new updates. To maintain positivity, we skipped the BFGS-PM update when $s_Z^{\mathrm{T}} y_Z < \varepsilon_{\mathrm{mach}}^{1/2} \|s_Z\|\,\|y_Z\|$.

5.1. Two algorithms for testing the LPB and FPB updates

The two algorithms shown below differ mainly in two features: the definition of the null-space step $h = Zs_Z$ and the use of a globalization procedure. Although both algorithms use the least squares multiplier estimate λ^{LS}, the local algorithm uses the modified SQP x-step (2.4) without line search or trust region. The semi-global algorithm uses the full SQP x-step (1.7) and a line search-based on the augmented Lagrangian merit function

$$L_{\mathrm{a}}(x, \lambda^{\mathrm{LS}}, \sigma) \doteq f(x) - c(x)^{\mathrm{T}}\lambda^{\mathrm{LS}} + \tfrac{1}{2}\sigma c(x)^{\mathrm{T}}c(x)$$

to insure that the Goldstein-Armijo (2.5) conditions are satisfied at each iteration. Thus,

$$L_{\mathrm{a}}(x_{k+1}, \lambda_k^{\mathrm{LS}}, \sigma) < L_{\mathrm{a}}(x_k, \lambda_k^{\mathrm{LS}}, \sigma)$$

for each iteration when σ is sufficiently large. However, this approach can not guarantee global convergence. Since λ^{LS} is recomputed at the end of each iteration, we can not guarantee

$$L_{\mathrm{a}}(x_{k+1}, \lambda_{k+1}^{\mathrm{LS}}, \sigma) < L_{\mathrm{a}}(x_{k+1}, \lambda_k^{\mathrm{LS}}, \sigma)$$

and thus L_{a} may not decrease. Near the solution, however, the multipliers will change very little and we can anticipate a monotonic decrease of the augmented Lagrangian. In the semi-global algorithm, M_σ is a factor greater than unity, e.g., $M_\sigma = 10$.

Semi-global algorithm.

```
*Initialize*:
    $k \leftarrow 0$.
    Choose $\sigma, M_\sigma, x_0, B_{YY}^0, B_{YZ}^0, B_{ZY}^0, B_{ZZ}^0$.
    Evaluate $f_0, g_0, c_0, A_0$.
    Factor $A_0 = [Y_0 \;\; Z_0][R_0^{\mathrm{T}} \;\; 0^{\mathrm{T}}]^{\mathrm{T}}$.
    Evaluate $\lambda_0^{\mathrm{LS}} = R_0^{-1} g_Y^0$, $L_{\mathrm{a}}(x_0, \lambda_0^{\mathrm{LS}}, \sigma)$.
    **Do while** $(\|g_Z^k\| > \varepsilon_{\mathrm{stop}}$ or $\|c_k\| > \varepsilon_{\mathrm{stop}})$
        $v_k \leftarrow Y_k s_Y^k = -Y_k R_k^{-\mathrm{T}} c_k$;
        $h_k \leftarrow Z_k s_Z^k = -Z_k (B_{ZZ}^k)^{-1}(g_Z^k + B_{ZY}^k s_Y^k)$;
        $s_k \leftarrow h_k + v_k$.
        **Do while** $((\nabla L_{\mathrm{a}}^k)^{\mathrm{T}} s_k \geq 0)$
            $\sigma \leftarrow \sigma M_\sigma$.
        **End**
            **Call** LINESEARCH$(\sigma, \tau_k, x_k, s_k, L_{\mathrm{a}}^k)$.
        $x_{k+1} \leftarrow x_k + \tau_k s_k$.
        Evaluate $f_{k+1}, g_{k+1}, c_{k+1}, A_{k+1}$.
        Factor $A_{k+1} = [Y_{k+1} \;\; Z_{k+1}][R_{k+1}^{\mathrm{T}} \;\; 0^{\mathrm{T}}]^{\mathrm{T}}$.
        Evaluate $\lambda_{k+1}^{\mathrm{LS}}$, $L_{\mathrm{a}}(x_{k+1}, \lambda_{k+1}^{\mathrm{LS}}, \sigma)$.
        $B^{k+1} \leftarrow U(B^k, x_k, v_k, h_k)$;
        $k \leftarrow k+1$.
    End
```

Note that the local approach has no linesearch and uses a modified h-step which ignores the change in gradients due to the step $v = Ys_Y$.

Local algorithm.

```
  Initialize:
    $k \leftarrow 0$.
    Choose $x_0, B^0_{YY}, B^0_{YZ}, B^0_{ZY}, B^0_{ZZ}$.
    Evaluate $f_0, g_0, c_0, A_0$.
    Factor $A_0 = [Y_0 \;\; Z_0][R_0^{\mathrm{T}} \;\; 0^{\mathrm{T}}]^{\mathrm{T}}$.
    Evaluate $\lambda_0^{\mathrm{LS}} = R_0^{-1} g_Y^0$.
    Do while ($\|g_Z^k\| > \varepsilon_{\text{stop}}$ or $\|c_k\| > \varepsilon_{\text{stop}}$)
        $v_k \leftarrow Y_k s_Y^k = -Y_k R_k^{-\mathrm{T}} c_k$;
        $h_k \leftarrow Z_k s_Z^k = -Z_k (B_{ZZ}^k)^{-1} g_Z^k$;
        $s_k \leftarrow h_k + v_k$;
        $x_{k+1} \leftarrow x_k + s_k$.
        Evaluate $f_{k+1}, g_{k+1}, c_{k+1}, A_{k+1}$.
        Factor $A_{k+1} = [Y_{k+1} \;\; Z_{k+1}][R_{k+1}^{\mathrm{T}} \;\; 0^{\mathrm{T}}]^{\mathrm{T}}$.
        Evaluate $\lambda_{k+1}^{\mathrm{LS}}$.
        $B^{k+1} \leftarrow U(B^k, x_k, v_k, h_k)$;
        $k \leftarrow k+1$.
    End
```

The formulas presented in Section 3 are used to define either the LPB or FPB update after solving the appropriate cubic equation and selecting a root. The Hessian approximations are stored in projected form. In all cases, we update the Cholesky factorization of $B_{ZZ} = LDL^{\mathrm{T}}$ using Gill et al.'s (1974) method. The LPB and FPB updates were skipped if the condition number estimate for B^+_{ZZ}, $\tilde{K}$, exceeded 10^5. When it was necessary to skip the FPB, we updated using the Modified FPB (MFPB),

$$\text{(MFPB)} \quad B^+_{YY} = B_{YY} + (G + G^{\mathrm{T}}) = B_{YY} + u_Y s_Y^{\mathrm{T}} + s_Y u_Y^{\mathrm{T}},$$
$$B^+_{YZ} = B_{YZ} + H^{\mathrm{T}} = B_{YZ} + u_Y s_Z^{\mathrm{T}} = B^{+\mathrm{T}}_{ZY},$$

where

$$u_Y = \frac{r_Y - s_Y \gamma}{s_Y^{\mathrm{T}} s_Y + s_Z^{\mathrm{T}} s_Z}, \qquad \gamma = \frac{s_Y^{\mathrm{T}} r_Y}{s_Y^{\mathrm{T}} s_Y + s^{\mathrm{T}} s}.$$

This update leaves B_{ZZ} unchanged, updating only the blocks B_{YY}, B_{YZ} and B_{ZY} while maintaining the quasi-Newton condition for only the upper half of the partitioned matrix. Specifically, it solves the variational problem

$$\underset{g,h}{\text{minimize}}\, \tfrac{1}{4} \left\| \begin{bmatrix} (G + G^{\mathrm{T}}) & H^{\mathrm{T}} \\ H & 0 \end{bmatrix} \right\|_{\mathrm{F}}^2$$

subject to the quasi-Newton condition

$$[(G+G^{\mathrm{T}}) \quad H^{\mathrm{T}}]\begin{pmatrix} s_Y \\ s_Z \end{pmatrix} = y_Y - B_{YY}s_Y - B_{YZ}s_Z \doteq r_Y.$$

When using the LPB and FPB, the matrix B_{ZY} is readily available to compute the horizontal step $h = -Z(B_{ZZ})^{-1}(g_Z + B_{ZY}s_Y)$. However, when using the BFGS-PM update we must approximate the term $B_{ZY}s_Y$. Since the extra gradient $\nabla L(x+Ys_Y)$ is required to compute the BFGS-PM update we also use it to approximate the vector $B_{ZY}s_Y$ with $Z^{\mathrm{T}}(\nabla L(x+Ys_Y) - \nabla L(x))$. For the LPB and FPB we initialize $B^0_{ZY} \leftarrow 0$ and $B^0_{YZ} \leftarrow 0$. The initial approximation $B^0_{YY} \leftarrow I_{t\times t}$ is used for the FPB. For most of the testing the first step is computed using $B^0_{ZZ} \leftarrow I_{(n-t)\times(n-t)}$. When using $B^0_{ZZ} = I$ and $B^0_{YY} = I$ we obtain better estimates of B_{ZZ} and B_{YY} by scaling the identity matrices at the end of the first iteration, before the first update. Consider scaling $\bar{B}^0_{ZZ} = \rho_{11}I$ and $\bar{B}^0_{ZZ} = \rho_{22}I$. Since $Y^{\mathrm{T}}\nabla^2 LZ$ and $Z^{\mathrm{T}}\nabla^2 LY$ are non-zero they satisfy

$$\begin{bmatrix} Y^{\mathrm{T}}\nabla^2 LY & Y^{\mathrm{T}}\nabla^2 LZ \\ Z^{\mathrm{T}}\nabla^2 LY & Z^{\mathrm{T}}\nabla^2 LZ \end{bmatrix}\begin{pmatrix} s_Y \\ s_Z \end{pmatrix} = \begin{pmatrix} y_Y \\ y_Z \end{pmatrix} + \mathrm{O}(\|s\|^2)$$

where y_Y and y_Z are projections of the gradient change $\nabla L(x+Ys_Y+Zs_Z) - \nabla L(x)$. Using an extra gradient evaluation at the point $x_0 + Ys_Y$ we define

$$\bar{y}_Y \doteq Y^{\mathrm{T}}(\nabla L(x_0+Ys_Y) - \nabla L(x_0)),$$

$$\bar{y}_Z \doteq Z^{\mathrm{T}}(\nabla L(x_0+Ys_Y+Zs_Z) - \nabla L(x_0+Ys_Y)),$$

and approximate

$$Y^{\mathrm{T}}\nabla^2 LYs_Y \cong y_Y - Y^{\mathrm{T}}\nabla^2 LZs_Z \cong \bar{y}_Y, \qquad Z^{\mathrm{T}}\nabla^2 LZs_Z \cong y_Z - Z^{\mathrm{T}}\nabla^2 LYs_Y \cong \bar{y}_Z.$$

Approximating $Y^{\mathrm{T}}\nabla^2 LY$ and $Z^{\mathrm{T}}\nabla^2 LZ$ by $\bar{B}^0_{YY}$ and $\bar{B}^0_{ZZ}$ in the directions s_Y and s_Z yields

$$s_Y^{\mathrm{T}}\bar{B}^0_{YY}s_Y = \rho_{11}s_Y^{\mathrm{T}}s_Y = s_Y^{\mathrm{T}}\bar{y}_Y, \qquad s_Z^{\mathrm{T}}\bar{B}^0_{ZZ}s_Z = \rho_{22}s_Z^{\mathrm{T}}s_Z = |s_Z^{\mathrm{T}}\bar{y}_Z|,$$

to give $\rho_{11} = s_Y^{\mathrm{T}}\bar{y}_Y/s_Y^{\mathrm{T}}s_Y$ and $\rho_{22} = |s_Z^{\mathrm{T}}\bar{y}_Z|/s_Z^{\mathrm{T}}s_Z$. When using the BFGS-PM update, the extra gradient at $x+Ys_Y$ is already available since it is required to compute the update. For the LPB and FPB algorithms, the extra gradient for the initial scaling increases the total number of gradient evaluations by one. Because the additional work is small, this improved scaling was implemented in both the semi-global and local algorithms.

Finally, the line-search procedure used in the semi-global algorithm was a test version of the routine CVSRCH supplied by J.J. Moré at Argonne National Laboratory. CVSRCH uses a combination of a cubic and a quadratic fit to find the safeguarded step which satisfies the sufficient decrease and curvature conditions of the Goldstein–Armijo (2.5) criteria.

5.2. Test problems

The problems briefly described in Table 5.1 were used to test the new updates in both the local and semi-global algorithms. Detailed descriptions of these problems may be found in Fenyes (1987) or in earlier sources. The first four problems were taken from Nocedal and Overton (1985). The next three are from Wright (1976) and Rosen and Suzuki (1965), and the problems labeled HSxx are from Hock and Schittkowski (1981). Here, N is the number of variables and T the number of constraints. "Objective" and "Constraints" briefly describe the functions using POLY O(n) to represent a polynomial of order n and EXP and TRIG to represent expontial and trigonometric functions respectively.

5.3. Comparison with Nocedal and Overton's method

Nocedal and Overton's (1985) local algorithm C7 uses the same step s and gradient change y as our local algorithm given above. They use a projected BFGS update formula, updating only if the step satisfies

$$\|v_k\| < \frac{\eta}{(k+1)^{1+\phi}} \|h_k\|$$

where they take $\phi = 0.01$ and $\eta = 1.0$. Although Nocedal and Overton's results are quite local, we felt it would be informative to compare our results with theirs. We selected five of their test problems P1, P2, P3, P4 and problem 100 from Hock and Schittkowski. In the listing above, these are referred to as problems NO1, NO2,

Table 5.1
Test problems

Problem	N	T	Objective	Constraints
NO1	2	1	POLY O(3)	POLY O(2)
NO2	3	2	POLY O(2)	POLY O(2)
NO3	5	3	EXP+POLY O(6)	POLY O(3)
NO4	5	3	POLY O(4)	POLY O(3)
Wright 8	9	6	POLY O(2)	POLY O(2)
Wright 9	5	2	TRIG+POLY O(9)	POLY O(3)
RS	4	2	POLY O(2)	POLY O(2)
HS27	3	1	POLY O(4)	POLY O(2)
HS53	5	3	POLY O(2)	POLY O(1)
HS60	3	1	POLY O(4)	POLY O(4)
HS61	3	2	POLY O(2)	POLY O(2)
HS77	5	2	POLY O(6)	TRIG+POLY O(6)
HS78	5	3	POLY O(5)	POLY O(3)
HS79	5	3	POLY O(4)	POLY O(3)
HS100	7	2	POLY O(6)	POLY O(4)

NO3, NO4 and HS100 respectively. Using Nocedal and Overton's starting points, we ran the local algorithm above, updating with the LPB, the FPB, and the BFGS-PM update. Like Nocedal and Overton, we initialize the LPB, FPB and BFGS-PM methods by setting B^0_{ZZ} to a finite difference estimate of $Z^T\nabla^2 LZ$. We insure positivity of this initial approximation by factoring it with the modified Cholesky decomposition of Gill and Murray (1974). For the LPB and FPB we set $B^0_{YZ} = B^0_{ZY} = 0$ and $B^0_{YY} = I$. The matrices B_{YY}, B_{YZ} and B_{ZY} are not used in the BFGS-PM method. For this test we set $\varepsilon_{\text{stop}} = 10^{-8}$.

The results are shown in Table 5.2 where the BFGS-PM and Nocedal and Overton methods are listed as BFGS and NO/C7 respectively. The columns labels are:

NIT: Number of iterations.

NGD: Number of times the gradients were evaluated.

NUS: Number of times the update was skipped.

NSY: Number of times $s^T y < 0$.

NT1: Number of times the Nocedal and Overton test failed with $\eta = 1.0$.

NT2: Number of times the Nocedal and Overton test failed with $\eta = 2.0$.

NT3: Number of times the Nocedal and Overton test failed with $\eta = 0.1$.

The count NGD does not include the gradients required to obtain the initial finite difference approximation of B_{ZZ}. Also, since an extra gradient is required to check

Table 5.2
Local algorithm: B_{22} by finite difference

Problem	Update	NIT	NGD	NUS	NSY	NT1	NT2	NT3
NO1	LPB	5	6	0	0	1	0	4
	FPB	6	7	0	0	1	0	4
	BFGS	5	11	0	0	1	0	4
	NO/C7	5	6	1	–	1	–	–
NO2	LPB	6	7	0	1	2	2	4
	FPB	6	7	0	1	3	2	4
	BFGS	5	11	0	0	3	2	4
	NO/C7	6	7	3	–	3	–	–
NO3	LPB	5	6	0	0	3	2	3
	FPB	6	7	0	0	3	1	4
	BFGS	4	9	0	0	2	1	2
	NO/C7	5	6	1	–	1	–	–
NO4	LPB	9	10	0	0	0	0	4
	FPB	10	11	0	0	1	0	5
	BFGS	8	17	0	0	1	0	5
	NO/C7	8	9	0	–	0	–	–
HS100	LPB	11	12	0	0	0	0	4
	FPB	12	13	0	0	1	0	6
	BFGS	11	23	0	0	1	0	5
	NO/C7	12	13	0	–	0	–	–

convergence at the last iteration, NGD=2(NIT)+1 for the BFGS-PM, and NGD= NIT+1 for all other methods.

All four updates achieve convergence in approximately the same number of iterations. The BFGS-PM method requires slightly fewer iterations but, as we would expect, this is achieved at the expense of nearly twice as many gradient evaluations. It is also interesting to note that even though the starting points used were very near the solutions, NO/C7 skips updates in three of the problems while the other methods are able to update at every iteration. In problem NO2, the algorithm NO/C7 seems to perform well, although only two of the five possible updates are performed. Skipping an update apparently has little effect in these problems because the initial finite difference Hessian approximation is taken near the solution.

5.4. Further testing of the local and global algorithms

The results in Table 5.2 are based on single starting points quite near the solution. Since the LPB and FPB were formulated to maintain a positive definite Hessian approximation even far from the solution, additional testing was performed to study the updates with a variety of starting points. A total of 540 starting points were randomly selected: 36 starting points for each of the 15 problems listed above. Twelve points were chosen from each of the full, range, or null spaces determined at the solution: three starting points at each of the distances 0.01, 0.10, 1.00 and 5.00 from the solution. By considering many problems with many starting points we obtain an unbiased view of the performance of the LPB, FPB, and the BFGS-PM updates when used in either the local or semi-global algorithm. In all cases we initialize B_{YY} and B_{ZZ} to scaled identity matrices and set $B_{YZ}=B_{ZY}=0$. We note that for the BFGS-PM update we must have $s_Z^{\mathrm{T}}y_Z>0$ in order to update. For the LPB and FPB it may still be possible to update even when $s_Z^{\mathrm{T}}y_Z<0$.

The detailed computational results presented in Fenyes (1987) are analyzed and summarized here. The starting points were divided into two groups: NEAR for the points at radius 0.01 and 0.10, and FAR for the points at radius 1.00 and 5.00. First, we determined the number of iterations and gradient evaluations required for convergence of each algorithm when using the BFGS-PM, LPB and FPB formulas from each starting point. Using SAS (SAS Institute Inc., 1982) we analyzed our results as discussed below.

The averages, grouped by starting distance from the solution, are shown in Figures 5.1 and 5.2 for the local and semi-global algorithms. They give a measure of the relative efficiencies for each update formula, but do not indicate if the differences are significant. To study these differences, we paired the tests by starting point and averaged, within groups, the quantities

$$\mathrm{NIT}_k(\mathrm{UPDT}_i)-\mathrm{NIT}_k(\mathrm{UPDT}_j), \qquad \mathrm{NGD}_k(\mathrm{UPDT}_i)-\mathrm{NGD}_k(\mathrm{UPDT}_j). \quad (5.1)$$

Here, $\mathrm{NIT}_k(\mathrm{UPDT}_i)$ indicates the number of iterations required for convergence

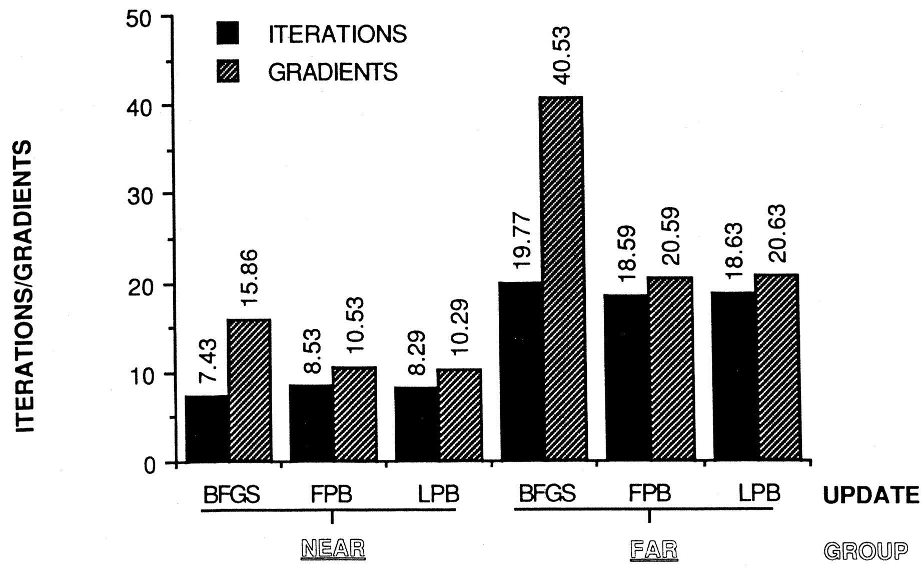

Fig. 5.1. Local algorithm.

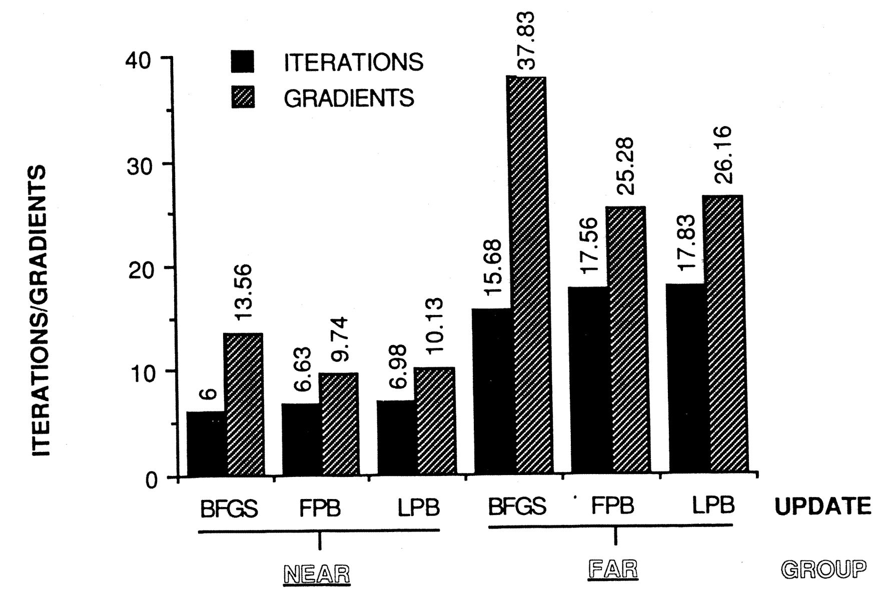

Fig. 5.2. Semi-global algorithm.

of update i from starting point k while i and j represent any of the BFGS-PM, LPB or FPB formulas. The paired t-test was used to condense the voluminous information and determine if the differences between updates were statistically significant. The paired t-statistics are presented in Tables 5.3, 5.4 and 5.5. Averages of the differences (5.1) are shown in the AVG columns. Under the null hypothesis that the means for the three updates were equal, we computed the significance levels shown in the column labelled SIG LVL for the alternate hypotheses $\overline{\mathrm{NIT}}(\mathrm{UPDT}_i) < \overline{\mathrm{NIT}}(\mathrm{UPDT}_j)$

Table 5.3

Comparison with BFGS-PM: Iterations

Algorithm	UPDT	Points	AVG(BFGS-UPDT)	SIG LVL
Local	LPB	NEAR	−0.86	0.00005
		FAR	+0.30	0.36970
	FPB	NEAR	−1.09	0.00005
		FAR	+0.16	0.39890
S-Glo	LPB	NEAR	−1.03	0.00005
		FAR	−1.65	0.00690
	FPB	NEAR	−0.68	0.00005
		FAR	−1.65	0.00450

Table 5.4

Comparison with BFGS-PM: Gradients

Algorithm	UPDT	Points	AVG(BFGS-UPDT)	SIG LVL
Local	LPB	NEAR	+5.57	0.00005
		FAR	+18.0	0.00005
	FPB	NEAR	+5.33	0.00005
		FAR	+17.71	0.00005
S-Glo	LPB	NEAR	+3.43	0.00005
		FAR	+13.11	0.00005
	FPB	NEAR	+3.82	0.00005
		FAR	+12.95	0.00005

Table 5.5

Comparison between LPB and FPB

Algorithm	Count	Points	AVG(LPB-FPB)	SIG LVL
Local	NIT	NEAR	−0.23	0.20385
		FAR	−0.25	0.32200
	NGD	NEAR	−0.23	0.20385
		FAR	−0.25	0.32200
S-Glo	NIT	NEAR	+0.35	0.00010
		FAR	−0.06	0.46125
	NGD	NEAR	+0.39	0.16040
		FAR	−0.26	0.41480

or, depending on the sign of the average difference, $\overline{\text{NIT}}(\text{UPDT}_i) > \overline{\text{NIT}}(\text{UPDT}_j)$. Here, we use the overbar to indicate an average. For values of SIG LVL less than 0.01 we concluded that the null hypothesis was false and that the two methods were statistically different. In other words, for SIG LVL less than 0.01, when $\text{AVG} > 0$ ($\text{AVG} < 0$) we could then conclude that $\overline{\text{NIT}}(\text{UPDT}_i) > \overline{\text{NIT}}(\text{UPDT}_j)$ ($\overline{\text{NIT}}(\text{UPDT}_i) < \overline{\text{NIT}}(\text{UPDT}_j)$). For SIG LVL greater than 0.01 we concluded that the update methods were, on average, the same.

Let us first consider Table 5.3, which compares the number of iterations required for convergence of the BFGS-PM with the LPB and FPB formulas.

We would expect the BFGS-PM to require fewer iterations for convergence, since it uses the additional gradient information at the midpoints. However, Table 5.3 shows this advantage is small. For starting points near the solution, the BFGS-PM formula has about a single iteration advantage over both the LPB and FPB formulas. For starting points far from the solution, the slight advantage of the BFGS-PM formula is maintained in the semi-global algorithm, but not for the local algorithm. No statistically significant difference can be demonstrated for the BFGS-PM compared to the LPB and FPB when using the local algorithm from starting points far from the solution.

Table 5.4 compares the number of gradients required for convergence in the BFGS-PM against the LPB and FPB. As we would expect, the BFGS-PM requires many more gradient evaluations than either the LPB or FPB formulas and the differences are statistically significant.

In fact, we could show, at a significance level of 0.01, that the BFGS-PM formula requires more than 13 (8) extra gradients than the LPB or FPB when using the local (semi-global) algorithm far from the solution.

In Table 5.5 we look for differences between the two new updates. The only statistically significant difference is the lower iteration count for the FPB formula when using the semi-global algorithm near the solution.

6. Discussion

It is clear that the LPB and the FPB formulas have several desirable algorithmic features. Since no extra midpoint gradient evaluations are needed to insure positivity of B_{ZZ}, these updates are very efficient. In addition, there is no need to satisfy a condition of the type $s^{\mathrm{T}}y > 0$ when $s_Y \neq 0$ and $s_Z \neq 0$, allowing us to update our Hessian approximation at nearly every iteration, even when far from the solution. Finally, Corollaries 3.2, 3.3, 3.5 and 3.6 show that the FPB and LPB formulas reduce to projected forms of the Broyden, PSB and BFGS updates in the special cases $s_Y = 0$ or $s_Z = 0$. This is a particularly nice property of the LPB and FPB formulas.

Theorems 3.1 and 3.4 show that these updates are well defined when both $s_Y \neq 0$ and $s_Z \neq 0$, and that the formulas will keep B_{ZZ} positive definite even when far from the solution. In some sense this is counter-intuitive. Suppose we have taken a step which lies mostly in the null space, in other words $s_Y \cong 0$, and we have $s_Z^T y_Z < 0$. In this situation the standard projected BFGS formula fails to keep B_{ZZ} positive definite and it must therefore be skipped. In contrast, although we have seen no positive curvature in the null space, we can still update with either the FPB or LPB and maintain positivity of B_{ZZ}. How can this be? Comparing the projected BFGS formula with the FPB we can resolve this apparent paradox. First consider the quasi-Newton equations for both formulas. For the standard BFGS we satisfy an equation of the form $B_{ZZ}^+ s_Z = y_Z$. If $s_Z^T y_Z = s_Z^T B_{ZZ}^+ s_Z < 0$ then clearly B_{ZZ}^+ is not positive definite and the update can not be performed. However, for the FPB we have the quasi-Newton equation $B_{ZZ}^+ s_Z = y_Z - B_{ZY}^+ s_Y$. Here, again, $s_Z^T B_{ZZ}^+ s_Z$ must be positive if B_{ZZ}^+ is to be positive definite. But this implies $s_Z^T y_Z - s_Z^T B_{ZY}^+ s_Y > 0$. In this case, when $s_Z^T y_Z < 0$ we can force $s_Z^T B_{ZZ}^+ s_Z > 0$ by choosing B_{ZY}^+ to satisfy $s_Z^T B_{ZY}^+ s_Z < s_Z^T y_Z$. In some sense, B_{ZY}^+ serves as an additional degree of freedom to maintain the positivity of B_{ZZ}^+. Of course, only certain choices of B_{ZY}^+ will minimize the matrix norm and satisfy the quasi-Newton equation — these choices are determined by the roots, α, of the cubic equation. Even when $s_Y \cong 0$ and $s_Z^T y_Z < 0$ this additional freedom allows us to keep B_{ZZ}^+ positive definite.

One desirable property of the BFGS and DFP updates is their invariance to variable scalings of the type $\bar{x} = Dx$ where D is a diagonal scaling matrix. This insures that a change of units or an arbitrary scaling of the variables will not affect the updates or the algorithmic performance. The Broyden and PSB updates do not have this scale invariance. Unfortunately, the LPB and FPB updates are not scale invariant. However, the LPB update will be nearly scale invariant if $s_Y^T s_Y$ is small relative to $s_Z^T B_{ZZ} s_Z$.

The numerical results of Section 5 are quite encouraging. The comparisons with the method of Nocedal and Overton have shown that our update formula leads to more frequent updates, even near the solution, and thus may be more practical. Finally, the statistical analysis using a large number of starting points showed that the FPB and LPB are quite competitive with the standard projected BFGS update. The LPB and FPB converge in about the same number of iterations as the projected BFGS, however they require significantly fewer gradient evaluations. In addition, we have proven (Fenyes, 1987) R-superlinear convergence of a local algorithm which uses the LPB formula with the SQP x-step (1.7) and the least-squares multiplier estimates. This proof follows that used by Powell (1978b) to show R-superlinear convergence of his modified BFGS update. Q-superlinear convergence is currently under study.

Acknowledgment

We wish to thank J. J. Moré for the use of the CVSRCH linesearch procedure which he wrote with D. J. Thuente at Argonne National Laboratories.

References

M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions, Applied Mathematics Series Vol. 55* (National Bureau of Standards, Washington, DC; Dover, New York, 1968).

L. Armijo, "Minimization of functions having Lipschitz-continuous first partial derivatives," *Pacific Journal of Mathematics* 16 (1966) 1-3.

M.C. Biggs, "Constrained minimization using recursive equality quadratic programming," in: F.A. Lootsma, ed., *Numerical Methods for Non-linear Optimization* (Academic Press, London and New York, 1972) pp. 411-428.

M.C. Biggs, "Constrained minimization using recursive quadratic programming: some alternative subproblem formulations," in: L.C.W. Dixon and G.P. Szegö, eds., *Towards Global Optimization, Proceedings Workshop University of Cagliari, 1974* (North-Holland, Amsterdam, 1975) pp. 341-349.

P.T. Boggs and J.W. Tolle, "Merit functions for constrained optimization," presented at: *SIAM Conference on Optimization, Houston, TX, May 18-20, 1987.*

T.F. Coleman and A.R. Conn, "On the local convergence of a quasi-Newton method for the nonlinear programming problem," *SIAM Journal on Numerical Analysis* 21 (1984) 755-769.

T.F. Coleman and P.A. Fenyes, "Partitioned quasi-Newton methods for nonlinear equality constrained optimization," Report TR 88-931, Department of Computer Science, Cornell University (Ithaca, NY, 1988).

G. Debreu, "Definite and semidefinite quadratic forms," *Econometrica* 20 (1952) 295-300.

J.E. Dennis and R.B. Schnabel, "Least change secant updates for quasi-Newton methods," *SIAM Review* 21 (1979) 443-459.

J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, *LINPACK Users' Guide* (SIAM Publications, Philadelphia, PA, 1979).

P.A. Fenyes, "Partitioned quasi-Newton methods for nonlinear equality constrained optimization," Ph.D. Thesis, Cornell University (Ithaca, NY, 1987).

R. Fontecilla, "Local convergence of secant methods for nonlinear constrained optimization (revised manuscript)," *SIAM Journal on Numerical Analysis* 25 (1988) 692-712.

D. Gabay, "Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization," *Mathematical Programming Study* 16 (1982) 18-44.

U.M. Garcia Palomares and O.L. Mangasarian, "Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems," *Mathematical Programming* 11 (1976) 1-13.

P.E. Gill, G.H. Golub, W. Murray and M.A. Saunders, "Methods for modifying matrix factorizations," *Mathematics of Computation* 28 (1974) 505-535.

P.E. Gill and W. Murray, "Newton-type methods for unconstrained and linearly constrained optimization," *Mathematical Programming* 7 (1974) 311-350.

P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, "Model building and practical aspects of nonlinear programming," in: K. Schittkowski, ed., *Computational Mathematical Programming, NATO ASI Series* (Springer, Berlin, 1985) pp. 209-247.

P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, "User's guide for NPSOL (Version 4.0): A Fortran package for nonlinear programming," Technical Report SOL 86-2, Systems Optimization Laboratory, Stanford University (Stanford, CA, 1986).

S.T. Glad, "Properties of updating methods for the multipliers in augmented Lagrangians," *Journal of Optimization Theory and Applications* 28 (1979) 135-156.

A.A. Goldstein, *Constructive Real Analysis* (Harper & Row, New York, 1967).

C.B. Gurwitz, "Sequential quadratic programming methods based on approximating a projected Hessian matrix," Technical Report 219, Department of Computer Science, Courant Institute of Mathematical Sciences (New York, 1986).

S.-P. Han, "Superlinearly convergent variable metric algorithms for general nonlinear programming problems," *Mathematical Programming* 11 (1976) 263-282.

M.R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications* 4 (1969) 303-320.

W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes, Lecture Notes in Economics and Mathematical Systems No. 187* (Springer, New York, 1981).

D.Q. Mayne and E. Polak, "A superlinearly convergent algorithm for constrained optimization problems," *Mathematical Programming Study* 16 (1982) 45-61.

W. Murray, "An algorithm for constrained minimization," in: R. Fletcher, ed., *Optimization* (Academic Press, London, 1969) pp. 247-258.

W. Murray and M.H. Wright, "Projected Lagrangian methods based on the trajectories of penalty and barrier functions," Report SOL 78-23, Department of Operations Research, Stanford University (Stanford, CA, 1978).

J. Nocedal and M. Overton, "Projected Hessian updating algorithms for nonlinearly constrained optimization," *SIAM Journal on Numerical Analysis* 22 (1985) 821-850.

M.J.D. Powell, "A fast algorithm for nonlinearly constrained optimization calculations," in: G.A. Watson, ed., *Numerical analysis, Proceedings of the Biennial Conference, Dundee, June 1977, Lecture Notes in Mathematics No. 630* (Springer, New York, 1978a) pp. 144-157.

M.J.D. Powell, "The convergence of variable metric methods for non-linearly constrained optimization calculations," in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., *Nonlinear Programming 3* (Academic Press, New York, 1978b) pp. 27-63.

M.J.D. Powell, "The performance of two subroutines for constrained optimization on some difficult test problems," in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, eds., *Numerical Optimization 1984* (SIAM, Philadelphia, PA, 1985) pp. 160-177.

M.J.D. Powell and Y. Yuan, "A recursive quadratic programming algorithm that uses differentiable exact penalty functions," *Mathematical Programming* 35 (1986) 265-278.

J.B. Rosen and S. Suzuki, "Construction of nonlinear programming test problems," *Communications of the Association on Computing Machinery* 8 (1965) 113.

SAS Institute, Inc., *SAS User's Guide: Basics 1982 Edition* (SAS Institute Inc., Cary, NY, 1982).

K. Schittkowski, "The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function," *Numerische Mathematik* 38 (1981) 83-114.

K. Schittkowski, "A unified nonlinear programming theory for penalty, multiplier, SQP and GRG methods," in: P. Thoft-Christensen, ed., *System Modelling and Optimization, Proceedings of the 11th IFIP Conference, Copenhagen, Denmark, July 25-29, 1983* (Springer, Berlin, 1984) pp. 299-310.

K. Tanabe, "Feasibility-improving gradient-acute-projection methods: A unified approach to nonlinear programming," *Lecture Notes in Numerical Applied Analysis No. 3* (Kinokuniya Book Store, Tokyo, 1981) pp. 57-76.

R.A. Tapia, "Diagonalized multiplied methods and quasi-Newton methods for constrained optimization," *Journal of Optimization Theory and Applications* 22 (1977) 135-194.

R.A. Tapia, "Quasi-Newton methods for equality constrained optimization: equivalence of existing methods and a new implementation," in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., *Nonlinear Programming 3* (Academic Press, New York, 1978).

R.S. Womersley and R. Fletcher, "An algorithm for composite nonsmooth optimization problems," Report NA/60, Department of Mathematics, University of Dundee (Dundee, Scotland, 1982).

M.H. Wright, "Numerical methods for nonlinearly constrained optimization," Ph.D. Thesis, Stanford University (Stanford, CA, 1976).