Bulletin of the Iranian Mathematical Society Vol. 38 *No.* 3 (2012), *pp 767-786.*

SUPERLINEARLY CONVERGENT EXACT PENALTY PROJECTED STRUCTURED HESSIAN UPDATING SCHEMES FOR CONSTRAINED NONLINEAR LEAST SQUARES: ASYMPTOTIC ANALYSIS

N. MAHDAVI-AMIRI AND M. R. ANSARI*

Communicated by Mohammad Asadzadeh

Abstract. We present a structured algorithm for solving constrained nonlinear least squares problems, and establish its local two-step Q-superlinear convergence. The approach is based on an adaptive structured scheme due to Mahdavi-Amiri and Bartels of the exact penalty method of Coleman and Conn for nonlinearly constrained optimization problems. The structured adaptation also makes use of the ideas of Nocedal and Overton for handling quasi-Newton updates of projected Hessians. We discuss the comparative results of the testing of our programs and three nonlinear programming codes from KNITRO on some randomly generated test problems due to Bartels and Mahdavi-Amiri. The results indeed confirm the practical significance of our special considerations for the inherent structure of the least squares.

1. Introduction

An exact penalty method is a sequential unconstrained minimization approach for solving general nonlinear programming (NLP) problems. Coleman and Conn [6, 7, 8] presented an effective exact penalty algorithm for solving generally constrained nonlinear minimization problems. Mahdavi-Amiri and Bartels [14] adapted the approach to the structured constrained least squares problems. Although computational results given in [14] showed a promising two step superlinear asymptotic convergence, but the authors did not provide a proof. Here, we present a variant of the approach in [14]. Our algorithm appropriates the

MSC(2010): Primary: 49M37; Secondary: 90C53, 49J52.

Keywords: Constrained nonlinear programming, exact penalty method, nonlinear least squares, projected structured Hessian update.

Received: 3 November 2010, Accepted: 26 April 2011.

*Corresponding author

ideas of Dennis, Martinez and Tapia [11] for the unconstrained case to the projected structured Hessian updating and employs a new structurally adapted line search strategy. We establish a local two-step superlinear convergence and give supporting computational results.

The remainder of our work is organized as follows. Section 2 gives the notation and relevant results on constrained nonlinear least squares problems and the exact penalty approach. Section 3 discusses a structured projected Hessian updating scheme for constrained nonlinear least squares problems. In Section 4, we establish the local two-step superlinear convergence properties. Computational results in Section 5 substantiate the theoretical results and show that the new algorithm is competitive as compared to the three algorithms recently developed in KNITRO [5] integrated package. There, we show the effectiveness and efficiency of the algorithm on randomly generated test problems of Bartels and Mahdavi-Amiri [1]. Finally, we conclude in Section 6.

Throughout, $\|.\|$ denotes the l_2 norm for vectors or matrices, $\|.\|_F$ and $\|.\|_M$ denote the Frobenius norm and weighted Frobenius norm for some nonsingular matrix M, as defined by $\|Q\|_F = \sqrt{Trace(QQ^T)}$ and $\|Q\|_M = \|MQM\|_F$, respectively.

2. Constrained Nonlinear Least Squares

The problem to be solved is:

$$\begin{array}{lll} \min\limits_{x} & \phi(x) = \dfrac{1}{2}F(x)^T F(x) & \\ s.t. & c_i(x) = 0, & i \in M_1 \\ & c_j(x) \geq 0, & j \in M_2. \end{array} \tag{2.1}$$

where $F(x) = [f_1(x), \ldots, f_l(x)]^T$, x is an n-vector, and each f_δ, the c_i and c_j are functions from $\mathbf{R}^n$ to $\mathbf{R}^1$, all assumed to be twice continuously differentiable. We refer to each f_δ as a *residual*. Although the problem (2.1) can be solved by a general constrained nonlinear optimization method, in most circumstances the inherent structure of the objective function in (2.1) makes it worthwhile to appropriate specially designed techniques. Notice that if $G(x)$ is the matrix with its columns being the gradients $\nabla f_\delta(x)$, then

$$\nabla \phi(x) = G(x)F(x),$$

$$\nabla^2 \phi(x) = G(x)G(x)^T + S(x), \tag{2.2}$$

where,

$$S(x) = \sum_{\delta=1}^{l} f_\delta(x) \nabla^2 f_\delta(x). \tag{2.3}$$

In many applications, it is possible to calculate the first partial derivatives that make up the matrix $G(x)$ explicitly. These could be used to calculate the gradient $\nabla\phi(x)$. However, the distinctive feature of a least squares problem is that by knowing $G(x)$, we can compute the first part of the Hessian $\nabla^2\phi(x)$ for free. Moreover, the term $G(x)G(x)^T$ is often more important than the second summation term in (2.2), either because of near-linearity of the model near the solution (that is, the $\nabla^2 f_\delta$ being small) or because of small residues (that is, the f_δ being small). Most algorithms for nonlinear least squares exploit these structural properties of the Hessian. For problems where $\|S(x)\|$ is small compared to $\left\|G(x)G(x)^T\right\|$ as x approaches a minimizer x^*, the term $S(x)$ is dropped in (2.2) to yield the Gauss-Newton method. This method works well when $G(x)$ has full row rank and $\|S(x)\|$ is "sufficiently small"; for discussions, see [9].

Approximation for $S(x)$ in cases where it cannot be neglected is a recurring theme in the literature. In unconstrained case, quasi-Newton approximations to only the second term, $S(x)$, of the Hessian matrix (2.2) have been developed [11, 12]. Pertinent to our proposal is the work by Dennis, Gay, and Welsch [10], who made investigations of quasi-Newton updates to a matrix $B \approx S(x)$. This additive structure was analyzed by Dennis, Martinez and Tapia [11]. These strategies are called "*structured quasi-Newton*" methods.

2.1. **An Exact Penalty Function.** Coleman and Conn [7] used the penalty function,

$$\psi(x,\mu) = \mu\phi(x) + \sum_{i\in M_1} |c_i(x)| - \sum_{j\in M_2} min(0, c_j(x)),$$

where the penalty parameter μ is a positive number. It is well known that for appropriate values of the penalty parameter μ, stationary points of $\psi(x,\mu)$ are either KKT points of the nonlinear program (2.1) or infeasible stationary points; see Byrd et al. [4]. Pietrzykowski [18] showed this penalty function to be *exact* in the sense that, if x^* is an isolated minimizer of (2.1) and the gradients of the *active (binding) constraints* at x^* are linearly independent, then there exists a real number $\mu^* > 0$ such that x^* is also an isolated local minimizer of $\psi(x,\mu)$, for each μ, $0 < \mu \leq \mu^*$.

2.2. **Activity, Stationarity, and Violation.** With respect to the embedded algorithm for minimizing ψ, two tolerance parameters, ϵ and τ, and five index sets, $AE(x,\epsilon)$, $AI(x,\epsilon)$, $AC(x,\epsilon)$, $VE(x,\epsilon)$ and $VI(x,\epsilon)$, are important for management. The parameter $\epsilon \geq 0$ is used to determine the closeness of any constraint to zero, and hence, to judge its activity. This is a standard antizigzagging device. The parameter $\tau > 0$ is used to judge the distance to the nearest stationary point. This judgement is used to separate the "global" phase of the algorithm from its "local" phase. The index sets $AE(x,\epsilon)$ of *active equality constraints*, $AI(x,\epsilon)$ of *active inequality constraints*, $AC(x,\epsilon)$ of *active constraints*, $VE(x,\epsilon)$ of *violated equality constraints*, and $VI(x,\epsilon)$ of *violated inequality constraints* are defined as follows:

$$AC(x,\epsilon) = \{i \in M_1 \cup M_2 \mid |c_i(x)| \leq \epsilon\},$$

$$\begin{aligned} AE(x,\epsilon) &= AC(x,\epsilon) \cap M_1, & AI(x,\epsilon) &= AC(x,\epsilon) \cap M_2, \\ VE(x,\epsilon) &= M_1 - AE(x,\epsilon), \quad \text{and} & VI(x,\epsilon) &= M_2 - AI(x,\epsilon). \end{aligned}$$

The minimization of ψ is carried out with the aid of an $\epsilon-$active merit function:

$$\psi_\epsilon(x,\mu) = \mu\phi(x) + \sum_{i \in VE(x,\epsilon)} sgn(c_i(x))c_i(x) - \sum_{j \in VI(x,\epsilon)} c_j(x).$$

This provides a differentiable approximation to the true merit function ψ. It is trivial that ψ_ϵ is equal in value to ψ when $\epsilon = 0$. Loosely speaking, step directions are determined using ψ_ϵ, but line searches and optimality tests use ψ.

The following vectors and matrices, defined in terms of the index sets above, are conveniently used:

– the active constraint gradient matrix,

$$A(x) = [\cdots \nabla c_r(x) \cdots]_{r \in AC(x,\epsilon)},$$

– the violated equality constraint gradient matrix,

$$E(x) = [\cdots \nabla c_i(x) \cdots]_{i \in VE(x,\epsilon)},$$

– the violated inequality constraint gradient matrix,

$$I(x) = [\cdots \nabla c_j(x) \cdots]_{j \in VI(x,\epsilon)},$$

– and the violated equality vector of signs,

$$\pi(x) = [\cdots sgn(c_i(x)) \cdots]^T_{i \in VE(x,\epsilon)}.$$

Using

$$e = [1 \cdots 1 \cdots 1]^T,$$

the gradient of ψ_ϵ is:

$$\nabla\psi_\epsilon(x,\mu) = \mu G(x)F(x) + E(x)\pi(x) - I(x)e,$$

and the Hessian of ψ_ϵ is:

$$\begin{aligned} \nabla^2\psi_\epsilon(x,\mu) = \mu(G(x)G(x)^T + S(x)) \quad &+ \sum_{i \in VE(x,\epsilon)} sgn(c_i(x))\nabla^2 c_i(x) \\ &- \sum_{j \in VI(x,\epsilon)} \nabla^2 c_j(x), \end{aligned}$$

where, $S(x)$ is defined by (2.3).

A necessary condition for x^* being an isolated local minimizer of ψ under the assumptions made above on ϕ, the c_i, and the c_j is that there exist *multipliers* λ^*_r, for $r \in AC(x^*,0)$, such that

$$\nabla\psi_0(x^*,\mu) = \sum_{r \in AC(x^*,0)} \lambda^*_r \nabla C_r(x^*) = A(x^*)\lambda^*, \tag{2.4}$$

and

$$-1 < \lambda_r^* < 1, \qquad r \in AE(x^*, 0), \tag{2.5}$$

$$0 < \lambda_r^* < 1, \qquad r \in AI(x^*, 0). \tag{2.6}$$

A point x for which only (2.4) above is satisfied is said to be a *stationary point* of ψ. If x^* is a minimizer, then it is necessary for x^* to be a stationary point and satisfy (2.5) and (2.6). Note that stationarity and optimality are determined using ψ_0, which is ψ_ϵ with $\epsilon = 0$.

2.3. **The Multiplier Estimates.** Estimates λ_r of the numbers λ_r^* are calculated only in the neighborhoods of stationary points. In such neighborhoods, the numbers λ_r are taken to be the least squares solution to:

$$\min_{\lambda} \| A(x)\lambda - \nabla \psi_\epsilon(x, \mu) \| , \tag{2.7}$$

and in practice, the QR decomposition of $A(x)$ is used to solve the least squares problem:

$$A(x) = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Y & Z \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{2.8}$$

If t is the number of columns in $A(x)$ and the columns are linearly independent, then the columns of Y and Z serve as bases for the range space of $A(x)$ and null space of $A(x)^T$; that is, $A(x) = YR$, with R nonsingular, $A(x)^T Z = 0$, $Y^T Y$ and $Z^T Z$ are the identities. Nearness to a stationarity point is determined by the *stationary tolerance* $\tau > 0$. The λ_r are computed only if $\left\| Z Z^T \nabla \psi_\epsilon(x, \mu) \right\| \le \tau$.

2.4. **The Quadratic Subproblem.** Fundamental to the Coleman and Conn approach is a particular unconstrained quadratic problem:

$$\min_{w} \quad (Z^T \nabla \psi_\epsilon(x, \mu))^T w + \frac{1}{2} w^T H_z w. \tag{2.9}$$

The matrix H_z is a positive definite approximation of one of two projected Hessian matrices, the choice of which depends upon whether the algorithm is in its "global" state, that is, far from a stationary point, or in its "local" state, that is, in the proximity of a stationary point. The two matrices to which H_z is to provide an approximation are "global projected Hessian",

$$Z^T \nabla^2 \psi_\epsilon(x, \mu) Z,$$

and "local projected Hessian",

$$Z^T \big(\nabla^2 \psi_\epsilon(x, \mu) - \sum_{r \in AC(x, \epsilon)} \lambda_r \nabla^2 c_r(x) \big) Z.$$

The idea to use an approximation of a "two-sided projected Hessian" was suggested by Murray and Wright [15] and was later discussed extensively by Nocedal and Overton [16]; also see [2, 8]. For the general constrained nonlinear least squares (CNLLS) problems, in Section 4, we make use of the ideas in [14] for structuring and [16] for projecting.

2.5. **The Step Directions.** The minimization of ψ is carried out using several alternative step directions derived from ψ_ϵ:

(1) the *global horizontal step direction*;
(2) the *dropping step direction*;
(3) the *Newton step direction*;
  (3a) the *asymptotic horizontal step direction*;
  (3b) the *vertical step direction.*

The global horizontal step direction is $h_G := Zw$, where w is the solution to the quadratic problem (2.9) that could be found by solving,

$$H_z w = -Z^T \nabla \psi_\epsilon(x, \mu). \tag{2.10}$$

The matrix H_z, of course, approximates the global projected Hessian.

The asymptotic horizontal step direction h_A is a component of the Newton step direction, $h_A + v$, that lies in the null space of $A(x)^T$. Newton steps are only attempted in the neighborhood of stationary points that are expected to be minimizers. The step direction h_A is computed as in the global case above. In this case, the matrix H_z is to be an approximation of the local projected Hessian.

At $x + h_A$, the constraints of $AC(x, \epsilon)$ may no longer be within ϵ of zero. The vertical step is derived from taking a single Newton step toward the value of v that could be found by solving,

$$A(x)^T v = -C_{AC(x,\epsilon)}(x + h_A),$$

where, $C_{AC(x,\epsilon)}$ is the vector of the active constraint functions, ordered in accordance with the columns of $A(x)$.

The name of dropping step direction derives from the fact that it *drops* a function c_i or c_j from the collection of active constraints and provides a direction that gives a local first-order decrease in the penalty function value. It is the step direction d that satisfies the system of equations,

$$A(x)^T d = -sgn(\lambda_r) e_r,$$

where, $r \in AC(x, \epsilon)$ is chosen for which one of (2.5) or (2.6) is violated, and e_r is the rth unit vector.

2.6. **Strategy for Choosing Step Directions and the Algorithm.** The steps described above are used, broadly speaking, as follows:

(1) When $\left\| ZZ^T \nabla \psi_\epsilon(x, \mu) \right\| > \tau$, then we set $\bar{x} \leftarrow x + \alpha h_G$, where a line search is used to determine $\alpha > 0$.
(2) When $\left\| ZZ^T \nabla \psi_\epsilon(x, \mu) \right\| \leq$, then the multipliers λ_r, $r \in AC(x, \epsilon)$, are approximated by the least squares solution for (2.7).

(a) If (2.5) or (2.6) is not satisfied, then an index $r \in AC(x,\epsilon)$ is chosen for which one of (2.5) or (2.6) is violated, and we set $\bar{x} \leftarrow x + \alpha d$, where a line search is used to determine $\alpha > 0$.

(b) If (2.5) and (2.6) are satisfied, then we set $\bar{x} \leftarrow x + h_A + v$. No line search is used for h_A or v.

Under the standing assumptions on the f_δ, c_i, and c_j, the steps incurred by the $\epsilon-$approximation to the penalty function produce descent for the penalty function if ϵ and τ are "correctly set". Here, we assume there exist a line search strategy to determine the step length satisfying a sufficient decrease in ψ that is characterized by the line search assumption (see [7], P. 152 part (v)). We have a new line search strategy that makes use of the structure of least squares and global convergence results to be discussed in a separate work. Algorithm 1 below gives an outline of the classical ℓ_1-penalty method.

Algorithm 1. *Classical ℓ_1-penalty method.*

```
Give $\mu_0 > 0$ and starting point $x_s^0$;
for $k = 0, 1, 2, \dots$
  find minimizer $x^k$ of $\psi(x,\mu)$, starting at $x_s^k$;
  if $x^k$ is feasible then STOP
  else
    choose new $\mu_{k+1} < \mu_k$ (say, $\mu_{k+1} = \mu_k/8$);
    set a new starting point $x_s^{k+1}$ (say, $x^k$);
  endif;
endfor.
```

Note that when the value of μ is too large, in which case the optimal point for ψ may be infeasible, Algorithm 1 reduces μ and the minimization of ψ is repeated (for more details, see the first two algorithms presented by Mahdavi-Amiri and Bartels [14]).

3. The Projected Least Squares Structure

In the rest of our work, we denote $Z(x)^T G(x) G(x)^T Z(x)$ by $Q_z(x)$. For computing the steps using (2.10), we wish to provide a quasi-Newton approximation H_z for

$$Z^T \left[\mu G(x) G(x)^T + S(x,\lambda)\right] Z,$$

where,

$$\begin{aligned} S(x,\lambda) &= \mu \sum_{\delta=1}^{l} f_\delta(x) \nabla^2 f_\delta(x) + \sum_{i \in VE(x,\epsilon)} sgn(c_i(x)) \nabla^2 c_i(x) \\ &\quad - \sum_{j \in VI(x,\epsilon)} \nabla^2 c_j(x) - \sum_{r \in AC(x,\epsilon)} \lambda_r \nabla^2 c_r(x). \end{aligned}$$

The matrix in square brackets is precisely the global Hessian when the λ_r are taken to be zero, and it is the local Hessian when the λ_r are the least squares estimates obtained from (2.7). We set

$$H_z = \mu Q_z(x) + B_z, \tag{3.1}$$

and provide a quasi-Newton approximation,

$$B_z \approx Z^T S(x, \lambda) Z.$$

Here, we discuss a way to derive a secant relation for B_z and explain how to update it. We interpret the multipliers λ^k to be zero or estimated according to (2.7), whichever is appropriate at the kth iteration of the algorithm. Consider the asymptotic case. We assume that the final active set has been identified so that for all further k, with x^* designating the optimal point, we have

$$AC(x^k, \epsilon) = AC(x^*, 0), \quad VE(x^k, \epsilon) = VE(x^*, 0), \quad VI(x^k, \epsilon) = VI(x^*, 0).$$

Suppose

$$B_{z,k} \approx Z_k^T S(x^k, \lambda^k) Z_k,$$

and we wish to update $B_{z,k}$ to $B_{z,k+1}$ approximating

$$B_{z,k+1} \approx Z_{k+1}^T S(x^{k+1}, \lambda^{k+1}) Z_{k+1}.$$

Nocedal and Overton [16] discussed an approach for the quasi-Newton updating of projected Hessian approximations of general nonlinear programming problems, and later Mahdavi-Amiri and Bartels [14] adapted it to approximating $S(x^k, \lambda^k)$. The authors in [14] resolved the difference in x along the subspaces defined by Z_{k+1} and Y_{k+1}. Based on their approach, we have

$$\bar{x} - x = \bar{Y} q + \bar{Z} s, \tag{3.2}$$

where, in order to simplify the notation, the presence of a bar above a quantity indicates that it is taken at iteration $k+1$, and the absence of a bar indicates iteration k. If the constraints are linear, then $q = 0$, for $k > 0$. In the nonlinear case, asymptotically, we expect $\bar{Y} q$ to become negligible. This follows the empirical observation that the final iterations are usually taken with an unchanging set of active constraints. The algorithm converges to the optimal point along the optimal manifold where the active functions may be regarded as suitably approximated by linear functions. Observe that, using (3.2), we have

$$s = \bar{Z}^T (\bar{x} - x), \quad q = \bar{Y}^T (\bar{x} - x). \tag{3.3}$$

Similar to the approach in Mahdavi-Amiri and Bartels [14], and assuming negligibility of $\bar{Y} q$, the matrix B_z is updated such that the new matrix $\bar{B}_z$ satisfies the secant condition

$$\bar{B}_z s = y,$$

where, s is defined by (3.3) and

$$y = \bar{Z}^T [\mu(\bar{G} - G)\bar{F} + (\bar{E} - E)\bar{\pi} - (\bar{I} - I)e + A\lambda]. \tag{3.4}$$

Therefore, the full projected Hessian required in the minimization process of Algorithm 1 is taken to be:

$$\bar{H}_z = \mu \bar{Q}_z + \bar{B}_z.$$

But our assumption about the asymptotic negligibility of $\bar{Y}q$ could, of course, be wrong. To protect ourselves, consistent with the practice of Nocedal and Overton [16], the update used to obtain $\bar{B}_z$ is carried out only if $\bar{Y}q$ has actually become negligible (that is, small relative to $\|s\|$). This means that we update the approximation matrix using a structured update formula such as DFP or BFGS if the following condition holds:

$$\|q\| < \frac{\eta \|s\|}{(k+1)^{1+\nu}}, \tag{3.5}$$

for $\nu = 0.01$, where η is a positive constant to be chosen and k is the iteration number as suggested in [16]. We have observed that B_z is updated on most iterations, suggesting that our underlying assumption about the negligibility of $\bar{Y}q$ is, indeed, reasonable.

In order to simplify the notation, for matrix A and $y \approx As$, we denote the BFGS update correction by

$$BFGS(s, y, A) = \frac{yy^T}{y^T s} - \frac{As(As)^T}{s^T As},$$

and the DFP update correction by

$$DFP(s, y, A) = \frac{(y - As)y^T + y(u - As)^T}{s^T y} - \frac{s^T(y - As)}{(s^T y)^2} yy^T.$$

Note that the update rule used by Mahdavi-Amiri and Bartels [14] was the standard BFGS formula, that is,

$$\bar{B}_z = B_z + BFGS(s, y, B_z),$$

where, s and y are defined by (3.3) and (3.4), respectively. Although computational results on a variety of test problems showed a local two-step superlinear convergence of the scheme, but a formal proof is still awaiting. Here, we propose a different scheme, for which we will be able to prove the superlinear convergence.

In a structured quasi-Newton method, u is set to be

$$u = \mu \bar{Q}_z s + y.$$

Based on the structure principle given by Dennis et al. [11], Engels and Martinez [12] derived the structured Broyden family. According to this, the BFGS update is given by

$$\bar{H}_z = H'_z + BFGS(s, U, H'_z),$$

and the DFP update is defined by

$$\bar{H}_z = H'_z + DFP(s, U, H'_z),$$

where,

$$H_z = \mu Q_z + B_z, \quad \text{and} \quad H'_z = \mu \bar{Q}_z + B_z.$$

From this, a BFGS B_z-update is determined to be

$$\bar{B}_z = B_z + BFGS(s, U, H'_z),$$

and a DFP update is given by

$$\bar{B}_z = B_z + DFP(s, U, H'_z). \tag{3.6}$$

Either update rule can be used, but we consider the BFGS update formula. BFGS has been chosen because of its generally satisfactory numerical behavior in other contexts. Later, we will show that, under standard assumptions, 'near' the solution, $u^T s$ is positive and it imposes positive definiteness on the matrix $\bar{H'}_z$. Note that, in general, it is possible that $\bar{H}_z$ is not be positive definite, even if H'_z is positive definite. In practice, we update B_z if and only if (3.5) holds and $u^T s > 0$. The positive definiteness of $\bar{H}_z$ is imposed by use of the modified Cholesky factorization; see [13] and references therein.

4. **Local Superlinear Convergence**

Here, we prove a local two-step superlinear convergence of our method. The asymptotic analysis is developed following the approach in Nocedal and Overton [16]. Our main new results here are the ones concerned with the analysis of our proposed structures in the projected Hessian and are given by Lemma 4.11 and Theorem 4.13; our other results have some similarities to the ones in [16], but correspond to our specific nonlinear least squares structure. To achieve a local two-step Q-superlinear convergence, we need to show that the relation (4.3), to be seen in Corollary 4.9, is satisfied with some bounds on errors of x-values. We will denote x^* as a local optimal solution of problem (2.1) and suppose the active set at x^* to be $\{1, \ldots, t\}$. Furthermore, we assume that we are sufficiently close to x^* so that the proper active set has been identified. Throughout the rest of this section, we need the following assumptions and notations.

Assumptions 4.1.

(**A**) *ϕ and $c_i, i \in M_1 \cup M_2$, are twice continuously differentiable and their first and second derivatives are uniformly bounded in norm on a compact set D;*

(**B**) *$\{x^k\}$ is generated by Algorithm 1 starting from an arbitrary initial point, and $x^k \in D$, for all k;*

(**C**) *the gradients of the active constraints for all $x \in D$ are linearly independent.*

Assumptions 4.2. *Assume that Assumptions 4.1 hold, and that $Y(x)$ and $Z(x)$, given by a particular implementation of (2.8), are Lipschitz continuous in D. We* *assume that $\nabla^2\phi$, $\nabla^2 c_i$, $i \in AC$, and $Q = Z^TGG^TZ$ are Lipschitz continuous matrix functions such that*

$$\|Q(x) - Q(y)\| \leq L\,\|x - y\|\,.$$

Notation 4.3. *Let $e_k = x^k - x^*$, $g = \nabla\psi$, $W = \nabla^2\psi - \sum_{i=1}^{t}\lambda_i\nabla^2 c_i$, $S_z = Z^TS(x,\lambda)Z$, $S_z^* = Z^{*T}S(x^*,\lambda^*)Z^*$, $H_z^* = Z^{*T}W^*Z^*$, and $G_z^* = Z^{*T}Y^*$, where $*$ denotes evaluation at (x^*,λ^*). Let $\lambda(x)$ be the least squares estimate at x for (2.7). We also write λ^k for $\lambda(x^k)$ and W_k for $W(x^k,\lambda^k)$.*

Remark 4.4. *Consider the function $g(x) - A(x)\lambda(x)$, the gradient (with respect to x) of the Lagrangian function,*

$$L(x,\lambda) = \psi(x,\mu) - \sum_{i\in M_1\cup M_2}\lambda_i c_i(x).$$

The Jacobian of this function is $W(x) - A(x)\dot{\lambda}(x)$, where $W(x)$ is the Hessian (with respect to x) of the Lagrangian function and $\dot{\lambda}$ represents the derivative of $\lambda(x)$. Since $\lambda(x)$ may be written as $(A(x)^TA(x))^{-1})A(x)^Tg(x)$, it follows from Assumptions 4.2 that $\dot{\lambda}(x)$, as well as $W(x)$, is Lipschitz continuous in a neighborhood of x^*. Thus, the mean value theorem (see Ortega and Rheinboldt [17]*), for $\|e_k\|$ and $\|e_{k+1}\|$ small enough, gives*

$$\left\|Z_{k+1}^T\big(g_{k+1} - g_k - A_k\lambda^k - W_{k+1}(x^{k+1} - x^k)\big)\right\| \leq C_1\left\|x^{k+1} - x^k\right\|^2, \tag{4.1}$$

Remark 4.5. *Note that by the definition of Algorithm 1, we always have*

$$Y_k^T(x^{k+1} - x^k) = -R_k^TC_{AC}(x^k + h^k),$$
$$Z_k^T(x^{k+1} - x^k) = -H_{z,k}^{-1}Z_k^Tg_k,$$

$$C_{AC}(x^k + h^k) = C_{AC}(x^k) + O(\left\|h^k\right\|^2). \tag{4.2}$$

Lemma 4.6. *If Assumptions 4.2 hold and $H_{z,k}^{-1}$ is bounded above, then*

(i) $\left\|h^k\right\| = O(\|e_k\|)$,
(ii) $\left\|v^k\right\| = O(\|e_k\|)$.

Proof. By Algorithm 1,

$$h^k = -Z_kH_{z,k}^{-1}Z_k^T\nabla\phi(x^k) = -Z_kH_{z,k}^{-1}Z_k^T\nabla L(x^k,\lambda^*).$$

Since $H_{z,k}^{-1}$ is bounded above, then we have

$$\left\|h^k\right\| = O(\left\|\nabla L(x^k,\lambda^*)\right\|).$$

Thus, using the fact that $\nabla L(x^*,\lambda^*) = 0$, and Lipschitz continuity, we obtain (i). From Algorithm 1 and (4.2), we have

$$v^k = -A_k(A_k^TA_k)^{-1}C_{AC}(x^k + h^k) = -A_k(A_k^TA_k)^{-1}\big(C_{AC}(x^k) + O(\left\|h^k\right\|^2)\big).$$

Thus,

$$\left\|v^k\right\| \leq \|A_k\| \cdot \|(A_k^T A_k)^{-1}\| \left(\left\|C_{AC}(x^k)\right\| + O(\left\|h^k\right\|^2)\right).$$

But, $C_{AC}(x^*) = 0$. Thus, using the Lipschitz continuity of the c_i, and the boundedness of $A_k(A_k^T A_k)^{-1}$, we have (ii). □

The next lemma shows the quadratic contraction of certain quantities.

Lemma 4.7. *Assume that for a given k, $\|e_k\|$ and $\|e_{k-1}\|$ are sufficiently small. Then,*

(i) $\|Y_k^T e_{k+1}\| \leq C_0 \|e_k\|^2$,
(ii) $\|Y_k^T e_k\| \leq C_0(\|e_k\|^2 + \|e_{k-1}\|^2)$,
(iii) $\|Y_k^T (x^{k+1} - x^k)\| \leq C_0(\|e_k\|^2 + \|e_{k-1}\|^2)$.

Proof. Applying the Lipschitz continuity of $C_{AC}(x)$ and Lemma 4.6, the proofs are quite similar to the ones given for Lemma 4.1 in [16]. □

The next theorem shows the important tow-step contraction in the error.

Theorem 4.8. *Suppose that Algorithm 1 is applied with any update rule. Let τ_2 be a given constant. Then, $\exists \epsilon_1 > 0$ such that for any iteration k, if $\left\|H_{z,k-1}^{-1}\right\| \leq \tau_2$, $\left\|H_{z,k}^{-1}\right\| \leq \tau_2$ and $\|e_{k-1}\| \leq \epsilon_1$, then*

(i) $\|e_k\| \leq C_1 \|e_{k-1}\|$,
(ii) $\|Y_{k+1}^T (x^{k+1} - x^k)\| \leq C_1(\|e_k\|^2 + \|e_{k-1}\|^2)$.
(iii) $\|e_{k+1}\| \leq C_1(\|e_{k-1}\|^2 + \|(H_{z,k} - H_z^*) Z_{k+1}^T e_k\|)$,
(iv) $\|e_{k+1}\| \leq C_1(\|e_{k-1}\|^2 + \|(H_{z,k} - H_z^*) Z_{k+1}^T (x^{k+1} - x^k)\|)$.

Here, C_1 is a constant, depending on τ and ϵ_1 but not on k. Assume for convenience that $C_1 \geq 1$ (the assumption that $\left\|H_{z,k}^{-1}\right\| \leq \tau_2$ is needed only for (iii)).

Proof. Applying the Lipschitz continuity of $Z(x)$, the proofs of (i), (iii), and (iv) are quite similar to the ones given for Theorem 4.1 in [16].
For (ii), we have from (i) and Lemma 4.7(i),

$$\|Y_k^T e_{k+1}\| \leq \bar{C} \|e_{k-1}\|^2,$$

where $\bar{C}$ is a constant. Thus,

$$\begin{aligned} \left\|Y_{k+1}^T (x^{k+1} - x^k)\right\| &\leq \left\|Y_k^T (x^{k+1} - x^k)\right\| + \|Y_{k+1}^T - Y_k^T\| \left\|x^{k+1} - x^k\right\| \\ &\leq \left\|Y_k^T (x^{k+1} - x^k)\right\| + \|Y_{k+1}^T - Y_k^T\| \left(\|e_{k+1}\| + \|e_k\|\right). \end{aligned}$$

Now, by (i), Lipschitz continuity of Y and Lemma 4.7(iii), we obtain (ii). □

The next corollary gives sufficient conditions for a two-step Q-superlinear convergence. Nocedal and Overton [16] established these conditions for a two-step superlinear convergence of their approach for nonlinear programs, in general, and here we establish them for our approach for the least squares problem.

Corollary 4.9. *Suppose that Algorithm 1 is applied with any update rule. If $x^k \to x^*$, $\left\| H_{z,k}^{-1} \right\| \le \tau_2$, for all k, and*

$$\omega_k \equiv \frac{\|(B_{z,k} - S_z^*)s_k\|}{\|x^{k+1} - x^k\|} \to 0, \tag{4.3}$$

then $x^k \to x^*$, at a two-step Q-superlinear rate.

Proof. Apply (3.1), Lipschitz continuity of $Q(x)$ and (4.3) to obtain

$$\frac{\|(H_{z,k} - H_z^*)s_k\|}{\|x^{k+1} - x^k\|} \to 0.$$

Now, the desired result is obtained in the light of Theorem 4.8 by Corollary 4.1 in [16]. □

Note that as pointed out in [16], Corollary 4.9 is a variation of Powell's sufficient condition for a local two-step Q-superlinear convergence [19].

Lemma 4.10. *Suppose that in Algorithm 1, (3.5) holds at iteration k . Then*

$$\left\| x^{k+1} - x^k \right\| \le (1 + \eta) \|s_k\| .$$

Proof. By (3.3), we have

$$\left\| x^{k+1} - x^k \right\| \le \left\| Y_{k+1}Y_{k+1}^T(x^{k+1} - x^k) \right\| + \left\| Z_{k+1}Z_{k+1}^T(x^{k+1} - x^k) \right\| = \|q_k\| + \|s_k\| ,$$

and the result follows from (3.5). □

In what follows, define:

$$M = (H_z^*)^{-\frac{1}{2}} \quad \text{and} \quad \sigma_k = \max(\left\| x^{k+1} - x^* \right\|, \left\| x^k - x^* \right\|),$$

and set:

$$\hat{H}'_{z,k} = MH'_{z,k}M, \quad \hat{u}_k = Mu_k, \quad \hat{s}_k = M^{-1}s_k, \quad \hat{H}_{z,k+1} = MH_{z,k+1}M.$$

Lemma 4.11. *Suppose that Assumptions 4.2 hold. Assume that x^k, $x^{k+1} \in D$. If y_k in (3.4) satisfies*

$$\|y_k - S_z^* s_k\| \le \gamma_k \|s_k\| , \tag{4.4}$$

for some positive sequence $\{\gamma_k\}$ with $\sigma_k \le \gamma_k$ for all k, then

$$\|u_k - H_z^* s_k\| \le (L\mu + 1)\gamma_k \|s_k\| , \tag{4.5}$$

and

$$\|\hat{u}_k - \hat{s}_k\| \le \|M\|^2 (L\mu + 1)\gamma_k \|\hat{s}_k\| . \tag{4.6}$$

Furthermore, assume that $\gamma_k \le \gamma$ with γ sufficiently small. Then, there exist positive constants β_1 and β_2 such that

$$\beta_1 \|\hat{s}_k\|^2 \le s_k^T u_k \le \beta_2 \|\hat{s}_k\|^2 , \quad \textit{and} \quad \beta_1 \|\hat{s}_k\| \le \|\hat{u}_k\| \le \beta_2 \|\hat{s}_k\| . \tag{4.7}$$

Proof. Assumptions 4.2 and (4.4) yield:

$$\begin{aligned}\|u_k - H_z^* s_k\| &\le \mu \left\|Q(x^{k+1}) - Q(x^*)\right\| \|s_k\| + \|y_k - S_z^* s_k\| \\ &\le (L\mu\sigma_k + \gamma_k) \|s_k\|,\end{aligned}$$

and since by assumption we know $\sigma_k \le \gamma_k$, we then have (4.5).
By (4.5), we have

$$\|\hat{u}_k - \hat{s}_k\| \le \|M\| \|u_k - H_z^* s_k\| \le \|M\| (L\mu + 1)\gamma_k \|s_k\| \le \|M\|^2 (L\mu + 1)\gamma_k \|\hat{s}_k\|,$$

and this yields (4.6).
If $\gamma' = \|M\|^2 (L\mu + 1)\gamma$, then (4.6) gives:

$$\begin{aligned}\left|s_k^T u_k - \|\hat{s}_k\|^2\right| &= \left|\hat{s}_k^T(\hat{u}_k - \hat{s}_k)\right| \le \|\hat{s}_k\| \|\hat{u}_k - \hat{s}_k\| \le \|M\|^2 (L\mu + 1)\gamma_k \|\hat{s}_k\|^2 \\ &\le \|M\|^2 (L\mu + 1)\gamma \|\hat{s}_k\|^2 = \gamma' \|\hat{s}_k\|^2,\end{aligned}$$

and thus we have

$$(1 - \gamma') \|\hat{s}_k\|^2 \le s_k^T u_k \le (1 + \gamma') \|\hat{s}_k\|^2 .$$

The inequality (4.6) yields:

$$|\|\hat{u}_k\| - \|\hat{s}_k\|| \le \|\hat{u}_k - \hat{s}_k\| \le \gamma' \|\hat{s}_k\|,$$

and

$$(1 - \gamma') \|\hat{s}_k\| \le \|\hat{u}_k\| \le (1 + \gamma') \|\hat{s}_k\| .$$

Thus, for γ sufficiently small, $\gamma' < 1$ and so (4.7) is established. □

Note that a result of Lemma 4.11 is used as conditions for updating, that is, we update B_z if and only if $s_k^T u_k > 0$ and (3.5) holds.

Lemma 4.12. *For some positive constants K_0 and K_1 and any $x \in D$, $x^k \in D$,*

(i) $\left\|Z(x)^T S(x^k, \lambda^k) Z(x) - Z(x)^T S(x^*, \lambda^*) Z(x)\right\| \le K_0 \left\|x^k - x^*\right\|$,

(ii) $\left\|Z_k^T S(x, \lambda(x)) Z_k - Z^{*T} S(x, \lambda(x)) Z^*\right\| \le K_1 \left\|x^k - x^*\right\|$.

Proof. These results follow almost directly by using $\|Z\| = 1$, the Lipschitz continuities of $S(x, \lambda(x))$ and $Z(x)$, and the boundedness of $\|S(x, \lambda(x))\|$ on D. □

Theorem 4.13. *Let s_k and y_k be defined by (3.3) and (3.4), respectively. Then, there exists $\epsilon_2 > 0$ such that if $\|e_k\| \le \epsilon_2$ and $\|e_{k+1}\| \le \epsilon_2$, then*

$$\|y_k - S_z^* s_k\| \le C_2 \max(\|e_k\|, \|e_{k+1}\|) \|s_k\| + \|Z^* S^* Y^* q_k\|, \tag{4.8}$$

where, $C_2 \ge 1$ is a constant independent of k, and if (3.5) holds, then

$$\|y_k - S_z^* s_k\| \le \gamma_k \|s_k\|,$$

where,

$$\sigma_k \le \gamma_k \le C_2(1 + \eta)\sigma_k + \frac{\eta \|Z^* S^* Y^*\|}{(1 + k)^{1+\nu}}.$$

Proof. In order to simplify the notation, the presence of a bar above a quantity indicates that it is evaluated at x^{k+1}, and the absence of a bar indicates that it is evaluated at x^k. From (4.1), Taylor's theorem, and Lipschitz continuities of $\nabla^2\phi$, the $\nabla^2 c_i$ and $\lambda(x)$, for ϵ_2 sufficiently small, we have

$$\begin{aligned}
\mu(\bar{G}-G)\bar{F} &= \mu\big(\sum_{k=1}^{l} f_k(\bar{x})\nabla^2 f_k(\bar{x})\big)(\bar{x}-x)+U_1(\bar{x}-x),\\
(\bar{E}-E)\bar{\pi} &= \big(\sum_{i\in VE(\bar{x},0)} sgn(c_i(\bar{x}))\nabla^2 c_i(\bar{x})\big)(\bar{x}-x)+U_2(\bar{x}-x),\\
(\bar{I}-I)e &= \big(\sum_{j\in VI(\bar{x},0)} \nabla^2 c_j(\bar{x})\big)(\bar{x}-x)+U_3(\bar{x}-x),\\
(\bar{A}-A)\bar{\lambda} &= \big(\sum_{r\in AC(\bar{x},0)} \nabla^2\bar{\lambda}_r c_r(\bar{x})\big)(\bar{x}-x)+U_4(\bar{x}-x),\\
(\bar{A}-A)(\bar{\lambda}-\lambda) &= U_5(\bar{x}-x),
\end{aligned}$$

where, $\|U_i\| = O(\|\bar{x}-x\|)$, for $i=1,\ldots,5$. Since $\bar{Z}^T\bar{A}=0$, then by (3.4) we get

$$\begin{aligned}
y &= \bar{Z}^T(\mu(\bar{G}-G)\bar{F}+(\bar{E}-E)\bar{\pi}-(\bar{I}-I)e+A\lambda)\\
&= \bar{Z}^T(\mu(\bar{G}-G)\bar{F}+(\bar{E}-E)\bar{\pi}-(\bar{I}-I)e+A\bar{\lambda}+(\bar{A}-A)(\bar{\lambda}-\lambda))\\
&= \bar{Z}^T S(\bar{x},\bar{\lambda})(\bar{x}-x)+\bar{Z}^T U(\bar{x}-x),
\end{aligned}$$

where, $\|U\| = O(\|\bar{x}-x\|)$.
Recall that $\bar{x}-x=\bar{Z}s+\bar{Y}q$, and thus

$$y = \bar{Z}^T S(\bar{x},\bar{\lambda})(\bar{Z}s+\bar{Y}q)+\bar{Z}^T U(\bar{Z}s+\bar{Y}q).$$

Then, we have

$$y - S_z^* s = \big(\bar{Z}^T S(\bar{x},\bar{\lambda})\bar{Z} - S_z^* + \bar{Z}^T U\bar{Z}\big)s + \big(\bar{Z}^T S(\bar{x},\bar{\lambda})\bar{Y} + \bar{Z}^T U\bar{Y}\big)q. \tag{4.9}$$

But

$$O(\|\bar{x}-x\|) = O(\max(\|\bar{x}-x^*\|, \|x-x^*\|)). \tag{4.10}$$

Now, in the light of Lemma 4.12 and provided that ϵ_2 is sufficiently small, it is straightforward to show that

$$\big\|\bar{Z}^T S(\bar{x},\bar{\lambda})\bar{Z} - S_z^* + \bar{Z}^T U\bar{Z}\big\| \le K_0' \|\bar{x}-x\|, \tag{4.11}$$

and

$$\big\|\big(\bar{Z}^T S(\bar{x},\bar{\lambda})\bar{Y} + \bar{Z}^T U\bar{Y}\big)q\big\| \le \|Z^* S^* Y^* q\| + K_1' \|\bar{x}-x\|, \tag{4.12}$$

where, K_0' and K_1' are some positive constants. Hence, (4.9), (4.11) and (4.12) give:

$$\|y - S_z^* s\| \le \|Z^* S^* Y^* q\| + (K_0' + K_1') \|\bar{x}-x\|. \tag{4.13}$$

Now, by (4.10) and (4.13) we have (4.8).
By (4.8) and Lemma 4.10,

$$\|y - S_z^* s\| \leq \gamma_k \|s\|,$$

where,

$$\sigma_k \leq \gamma_k \leq C_2(1+\eta)\sigma_k + \|Z^* S^* Y^*\| \frac{\|q_k\|}{\|s_k\|}.$$

Note that we can assume $C_2 \geq 1$. Now, in Algorithm 1, if the update formula is used at iteration k, that is, (3.5) holds, then we have

$$\sigma_k \leq \gamma_k \leq C_2(1+\eta)\sigma_k + \frac{\eta \|Z^* S^* Y^*\|}{(1+k)^{1+\nu}}.$$

□

Next, we state a well-known lemma.

Lemma 4.14. *Let*

$$\frac{\|\hat{u}_k - \hat{s}_k\|}{\|\hat{s}_k\|} \leq \frac{1}{3}, \quad \text{with} \quad \hat{s} \neq 0.$$

Assume that the update formula (3.6) is used at iteration k. Then, $u_k^T s_k > 0$ *and*

$$\text{(4.14)} \qquad \left\|H'_{z,k+1} - H_z^*\right\|_M \leq ((1 - \bar{\alpha}_0 \theta_k^2)^{1/2} + \frac{\bar{\alpha}_1 \|\hat{u}_k - \hat{s}_k\|}{\|\hat{s}_k\|}) \left\|H'_{z,k} - H_z^*\right\|_M + \frac{\bar{\alpha}_2 \|u_k - H_z^* s_k\|}{\|s_k\|},$$

where, $\bar{\alpha}_0 \in (0,1]$, $\bar{\alpha}_1$, $\bar{\alpha}_2$ *are constants independent of k, and*

$$1 \geq \theta_k = \begin{cases} \frac{\left\|(\hat{H}'_{z,k} - \hat{H}_z^*)\hat{s}_k\right\|_M}{\left\|\hat{H}'_{z,k} - \hat{H}_z^*\right\|_M \|\hat{s}_k\|}, & \hat{H}'_{z,k} \neq \hat{H}_z^*, \\ 0, & \hat{H}'_{z,k} = \hat{H}_z^*. \end{cases}$$

Proof. See Broyden, Dennis, and Moré ([3], P. 223 along with P. 242). □

To achieve a superlinear rate of convergence, we need a more refined version of (4.14) by extending the result in [3].

Lemma 4.15. *Let the assumptions of Lemma 4.14 hold. Then, there are* α_0, α_1 *and* α_2, *constants independent of k, such that*

$$\|H_{z,k+1} - H_z^*\|_M \leq ((1 - \alpha_0 \theta_k^2)^{1/2} + \alpha_1 \left\|x^k - x^*\right\|) \|H_{z,k} - H_z^*\|_M + \alpha_2 \left\|x^k - x^*\right\|,$$

where,

$$1 \geq \theta_k = \begin{cases} \frac{\left\|(\hat{H}_{z,k} - \hat{H}_z^*)\hat{s}_k\right\|_M}{\left\|\hat{H}_{z,k} - \hat{H}_z^*\right\|_M \|\hat{s}_k\|}, & \hat{H}_{z,k} \neq \hat{H}_z^*, \\ 0, & \hat{H}_{z,k} = \hat{H}_z^*. \end{cases}$$

Proof. The result is obtained by Lemma 4.14, Lipschitz continuity of $Q(x)$ and the fact that $H'_{z,k} - H_{z,k} = \mu(Q_{k+1} - Q_k)$. □

Now, by (3.1), Lipschitz continuity of $Q(x)$ and with Lemma 4.15 at hand, the local linear convergence theorem (Theorem 4.3 in [16]) holds as stated next.

Theorem 4.16. *Suppose that Assumptions 4.2 hold. Let the matrix $B_{z,k}$ be updated by (3.6). Let the sequence $\left\{x^k\right\}$ be generated by Algorithm 1. Then, for any $r \in (0,1)$, there exist positive constants ϵ and δ such that if $\|e_0\| \leq \epsilon$ and $\|B_{z,0} - S_z^*\|_M \leq \delta$, then*

$$\|e_{k+1}\| \leq r \, \|e_{k-1}\| \,, \quad k \geq 1,$$

that is, $x^k \to x^*$ at least at a two-step Q-linear rate.

Now, we are ready to state our main result.

Theorem 4.17. *Suppose that all conditions of Theorem 4.16 hold. Then, the sequence $\left\{x^k\right\}$ generated by Algorithm 1 with the structured DFP update rule (3.6), converges two-step Q-superlinearly to x^*.*

Proof. By Theorem 4.16 and Lipschitz continuity of $Q(x)$, we have

$$\|(B_{z,k} - S_z^*)s_k\| = O(\|(H_{z,k} - H_z^*)s_k\| \, . \tag{4.15}$$

Considering (4.15) and Lipschitz continuity of $Z(x)$, the rest of proof follows quite similar to the proof of Theorem 4.4 in [16]. □

5. **Computational Results**

Here, we report the comparative results obtained on some test problems. For brevity, the global convergence results, implementation details and more extensive computational results will come in a separate paper. Bartels and Mahdavi-Aimiri [1] presented a program for random generation of general nonlinearly constrained nonlinear programming problems. They also gave an example of a least squares problem where the dimensions and the curvature of the problem could be varied. We considered seven problems generated from this class of problems. All of these random test problems have both nonlinear constraints and nonlinear residual functions in the objective function. Our program has successfully solved all of the considered problems quite efficiently. The local convergence on all problems clearly showed a two-step superlinear rate. We compared our results with the ones obtained by our testing of three algorithms (Interior/Direct, Interior/CG and Active-Set) recently developed in the KNITRO 6.0 package. Our call to programs in KNITRO are managed by a C++ program having the same testing environment as our program. For the three programs, we used the default parameter values without defining any spacial case for the objective function or constraints, and used the dense quasi-Newton BFGS Hessian approximation of the Hessians.

Table 1. Results on randomly generated test problems

PN	SP	σ	MA&A	D	CG	AS
1	1	1	19	36	45	45
1	10	1	40	58	62	327
1	100	1	40	83	77	97
2	1	-1	25	42	45	45
2	10	-1	45	64	76	256
2	100	-1	42	91	72	132
3	1	-10	37	52	69	54
3	10	-10	45	67	77	312
3	100	-10	37	93	87	74
4	1	1	36	266	163	151
4	10	1	56	169	130	417
5	1	-1	55	386	139	193
5	10	-1	117	182	99	766
6	1	1	105	533	3096	406
6	10	1	116	8140	17787	406
7	1	-1	125	13174	423	127
7	10	-1	140	7835	255	3813

The measures taken to implement Algorithm 1 such as test for nearness to stationarity, test for feasibility and other computational decisions are the same as the corresponding ones in [14]. The numerical positive definiteness of the matrix H_z is enforced by using the modified Cholesky factorization described in [13] during the process of solving (2.10).

Our algorithm was coded in a portable subset of C++ and ran in double precision arithmetic on the microsoft visual C++ 6.0 compiler on an x86-based PC with an AMD 1667Mhz processor. The parameters of Sections 4 and the corresponding ones that appeared in [14] were set as follows:

$$\eta = 1,\ \tau = 0.1,\ \epsilon = 0.01,\ \gamma = 10^{-7},\ \theta = 10^{-4},\ \text{and}\ \beta = 10^{-8}.$$

These values were determined by experience as ones that have worked well on most of the problems we had tried.

Our method of counting function evaluations is consistent with KNITRO, so that it serves as a basis for comparison. Obviously, the function evaluations are compared on cases where the algorithms have converged to the same solution point. In Table 1, we report the results obtained on three sets of random problems (problems 1, 2 and 3 as set 1, problems 4 and 5 as set 2, and finally problems 6 and 7 as set 3) generated and tested by Bartels and Mahdavi-Amiri [1]. Each set of random problems are generated with all quantities (x^*, Lagrange multipliers, U_1, etc.) being exactly the same, and only having different $\nabla^2 L$ obtained by setting different values for σ ($\nabla^2 L$ is definite if $\sigma > 0$ and is indefinite if $\sigma \leq 0$;

for details, see [1]). The numbers of the equality constraints, the inequality constraints and the active constraints at solution for the three sets respectively are (2, 3, 4), (5, 5, 7) and (8, 12, 10). In Table 1, the "SP" column shows the value to which all component of the starting point are set. The "σ" column shows the value considered for σ. The columns headed by "MA&A", "D", "CG" and "AS" respectively give the number of function evaluations required by our algorithm, Interior/Direct (barrier), Interior/CG (barrier) and Active-Set algorithms. The results show our method to be competitive in comparison with those solved by KNITRO. Thus, our new code here with the projected structured Hessian updating along with the new line search strategy significantly outperforms the three general nonlinear algorithms of KNITRO [5]. This attests to the practical appropriateness of our structural approach to nonlinear least squares.

6. Conclusions

We proposed a structurally projected exact penalty method for solving constrained nonlinear least squares problems. The projected structure makes appropriate use of the ideas of Nocedal and Overton for general nonlinear programs. The analytical and numerical results showed our proposed algorithm to be efficient and reliable. We proved a local two-step superlinear convergence. We implemented the proposed algorithm and tested the resulting program and three nonlinear programming codes from KNITRO on some randomly generated problems of Bartels and Mahdavi-Amiri. The results indeed confirmed the practical significance of our special structured considerations for the projected least squares Hessians.

Acknowledgments

The first author thanks the Research Council of Sharif University of Technology for supporting this work.

References

[1] R. H. Bartels and N. Mahdavi-Amiri, On generating test problems for nonlinear programming algorithms, *SIAM J. Sci. Stat. Comput.* **7** (1986), no. 3, 769–798.

[2] L. T. Biegler, J. Nocedal and C. Schmid, A reduced Hessian method for large-scale constrained optimization, *SIAM J. Optim.* **5** (1995), no. 2, 314–347.

[3] C. G. Broyden, J. E. Dennis, Jr. and J. J. Moré, On the local and superlinear convergence of quasi-Newton methods, *J. Inst. Math. Appl.* **12** (1973), 223–245.

[4] R. H. Byrd, N. I. M. Gould, J. Nocedal and R. A. Waltz, On the convergence of successive linear-quadratic programming algorithms, *SIAM J. Optim.* **16** (2005), no. 2, 471–489.

[5] R. H. Byrd, J. Nocedal and R. A. Waltz, KNITRO: An integrated package for nonlinear optimization, in: G. Di Pillo and M. Roma (Ed.) *Larg-Scale Nonlinear Optimization*, Springer, 2006, pp. 35–59.

[6] T. F. Coleman and A. R. Conn, Nonlinear programming via an exact penalty function: Asymptotic analysis, *Math. Program.* **24** (1982), no. 2, 123–136.

[7] T. F. Coleman and A. R. Conn, Nonlinear programming via an exact penalty function: Global analysis, *Math. Program.* **24** (1982), no. 2, 137–161.

[8] T. F. Coleman and A. R. Conn, On the local convergence of a quasi-Newton method for the nonlinear programming problem, *SIAM J. Numer. Anal.* **21** (1984), no. 4, 755–769.

[9] J. E. Dennis, Jr., Nonlinear Least Squares and Equations, in: D. Jacobs (Ed.) *The State of the Art of Numerical Analysis*, Academic Press, Orlando, Fla., 1977.

[10] J. E. Dennis, Jr., D. M. Gay and R. E. Welsch, An adaptive nonlinear least-squares algorithm, *ACM Trans. Math. Soft.* **7** (1981), no. 3, 348–368.

[11] J. E. Dennis, Jr., H. J. Martinez and R. A. Tapia, A convergence theory for the structured BFGS secant method with an application to nonlinear least squares, *J. Optim. Theory Appl.* **61** (1989), 161–178.

[12] J. R. Engels and H. J. Martinez, Local and superlinear convergence for partially known quasi-Newton methods, *SIAM J. Optim.* **1** (1991), no. 1, 42–56.

[13] H. -R. Fang and D. P. O'leary, Modified Cholesky algorithm: A catalog with new approaches, *Math. Program.* **115** (2008), no. 2, 319–349.

[14] N. Mahdavi-Amiri and R. H. Bartels, Constrained nonlinear least squares: An exact penalty approach with projected structured quasi-Newton updates, *ACM Trans. Math. Software* **15** (1989), no. 3, 220–242.

[15] W. Murray and M. H. Wright, Projected Lagrangian methods based on the trajectories of penalty and barrier functions, *Systems Optimization Laboratory Report*, Stanford University, Stanford, Calif., 1978, pp. 23–78.

[16] J. Nocedal and M. L. Overton, Projected Hessian updating algorithms for nonlinearly constrained optimization, *SIAM J. Numer. Anal.* **22** (1985), no. 5, 821–850.

[17] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York and London, 1970.

[18] T. Pietrzykowski, An exact potential method for constrained maxima, *SIAM J. Anal.* **6** (1969) 299–304.

[19] M. J. D. Powell, The convergence of variable metric methods for nonlinearly constrained optimization calculations, in: O. Mangasarian, R. Meyer and S. Robinson (Ed.) *Nonlinear Programming 3*, Academic Press, New York and London, 1978, pp. 224–248.

Nezamm Mahdavi-Amiri

Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran

Email: nezamm@sina.sharif.edu

Mohammad Reza Ansari

Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran

Email: ansari.mr@gmail.com