On a bound constrained optimization technique using second order information

Rolf Felkel*
Numerical Analysis Group
Department of Mathematics
Technical University Darmstadt
Schloßgartenstraße 7
D-64289 Darmstadt

September 3, 1998

Abstract

In this paper an algorithm for the unconstrained and bound constrained minimization problem using second order information is presented. It determines a point satisfying the second order necessary optimality condition. A truncated Lanczos decomposition of the reduced Hessian of the objective function is computed in order to determine three directions of descent, and, if a negative curvature direction is found, a special linesearch method is used. In the case of a bound constrained problem this approach is combined with projected gradient steps to obtain an efficient activation/inactivation strategie. As only matrix-vector-products are needed within the Lanczos algorithm the method is of special interest for large scale problems. The convergence theory is outlined and numerical results are presented.

Key words

Large scale bound constrained minimization, gradient projection step, negative curvature directions, Lanczos algorithm, partial reorthogonalization, linesearch methods

1 Introduction

In this paper we consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & l_i \le x_i \le u_i \quad i = 1, \ldots, n. \end{array} \tag{BCP}$$

*email: felkel@mathematik.tu-darmstadt.de

We assume $f \in C^3(\mathcal{D})$ with $\mathcal{D}$ an open, convex superset of the feasible box $\Omega_{\text{B}} = \prod_{i=1}^n [l_i, u_i]$. Note that some components of l and u may be infinite. Throughout the paper we assume that for the initial point $x^0 \in \mathcal{D}$ the intersection of the level set

$$\mathcal{L}_f(f(x^0)) \stackrel{\text{def}}{=} \{x \in \mathcal{D} : f(x) \leq f(x^0)\}$$

and the feasible region Ω_{B} is compact.

In section 2 a method for unconstrained programming problems is outlined. The development of the algorithm and the related convergence theory was done by Heinrich in [Hei85]. He used a spectral or a Bunch-Parlett decomposition of the exact Hessian in order to compute up to three directions of descent. Even for medium scale problems a spectral or a Bunch-Parlett decomposition of the Hessian is "expensive". A much "cheaper" way to get some information about the spectrum of the Hessian is to use a truncated Lanczos decomposition to obtain some ritzvalues approximating the outer eigenvalues (see [GL89], [Par80b] or [ST85]). This information can also be used to compute directions of descent.

In section 3 the unconstrained minimization technique is combined with a gradient projection method as proposed by Moré and Toraldo in [MT91]. The projected gradient steps are used as a multiple activation/inactivation stategy to identify the set of active variables. Whenever the gradient projection stops on a face of the box the unconstrained minimizer is used to minimize the objective function over the set of free variables. The convergence properties derived in section 2.2 can be used to prove convergence to a point satisfying the second order necessary optimality condition.

Section 4 deals with the implementation of the presented method. Some features to increase stability and effectiveness of the Lanczos algorithm are presented and the termination criteria are given.

Finally the Constrained and Unconstrained Testing Environment CUTE (see [BCGT97]) is used to solve several small, medium and large scale bound constrained and unconstrained programming problems. The numerical results are presented in section 5. To show the advantage of our method in solving (very) large scale problems the comparison with the limited memory quasi-Newton method LBFGS-B (see [BLNZ95]) is given.

Notation

We denote a diagonal matrix, whose i-th diagonal element is d_i, by $\text{diag}(d_i)$ and x_i denotes the i-th component of a vector x. The index of an element from a sequence of vectors is denoted as a superscript, as in x^k. In the case of a sequence of scalars the index is subscripted, as in σ_l. Whenever the power of a scalar is needed we write the base in brackets. Hence $\left(x_i^k\right)^2$ denotes the square of the i-th component of the k-th element in the vector-sequence $\{x^k\}_{k\to\infty} \subset \mathbb{R}^n$. For a matrix B $B_{\mathcal{A}}$ denotes the matrix made up from columns with indices in $\mathcal{A}$. The gradient of the objective function f is denoted by the column vector $\nabla f(x)$ and the Hessian of f is $\nabla^2 f(x)$. The eigenvalues of the Hessian (or any other matrix) are given by $\lambda_i(\nabla^2 f(x))$. $\lambda_{\min}(\cdot)$

denotes the minimum eigenvalue. Throughout the paper $\|\cdot\|$ is the Euclidean norm $(x^T x)^{\frac{1}{2}}$ of a vector x.

2 Unconstrained problems

Finding a local solution of the unconstrained minimization problem

$$\text{minimize } f(x) \tag{UCP}$$

can be done by searching a point x^* such that $\nabla f(x^*) = 0$ and $\lambda_{\min}(\nabla^2 f(x^*)) > 0$. This is a sufficient condition. Necessary for a local minimizer is $\nabla f(x^*) = 0$ and $\lambda_{\min}(\nabla^2 f(x^*)) \geq 0$. The algorithm outlined below tries to find a point x^* satisfying the necessary condition.

2.1 Outline of the algorithm

Algorithm 2.1 (UCA)
Given the starting point x^0, a regularity threshold $\rho > 0$ and a termination parameter $0 < \epsilon \ll 1$. Set $k = 0$ and proceed

Step 1 *Use $-\nabla f(x^k)$ as initial vector to compute a truncated Lanczos decomposition of the Hessian $\nabla^2 f(x^k)$ using maximum of $\bar{\nu}$ Lanczos steps*

$$\begin{array}{rcl} \nabla^2 f(x) Q_\nu^k - Q_\nu^k T_\nu^k &=& \beta_{\nu+1} q_{\nu+1} e_\nu^T \\ (Q_\nu^k)^T Q_\nu^k &=& I_\nu, \end{array} \tag{1}$$

where $\nu \leq \bar{\nu}$ is the actual number of computed Lanczos vectors. Compute the eigenvalues and eigenvectors of the tridiagonal matrix T_ν^k and get the ritzvalues and -vectors

$$Q_\nu^k T_\nu^k (Q_\nu^k)^T = Q_\nu^k V_\nu^k D_\nu^k (Q_\nu^k V_\nu^k)^T = S_\nu^k D_\nu^k (S_\nu^k)^T. \tag{2}$$

$D_\nu^k = diag\big((\lambda_i^k)_{i=1,\ldots,\nu}\big)$ is the diagonal matrix of the ritzvalues and $S_\nu^k = (s_1^k, \ldots, s_\nu^k) \in \mathbb{R}^{n\times\nu}$ is the matrix of the ritzvectors. Determine the index sets $\mathcal{N}_- = \{i : \lambda_i^k < 0\}$ and $\mathcal{N}_{+0} = \{i : \lambda_i^k \geq 0\}$ of negative and non-negative ritzvalues. W.l.o.g. we assume that the ritzvalues are sorted in ascending order $\lambda_1^k \leq \cdots \leq \lambda_\nu^k$.

Step 2 *Compute three different directions of descent*

a) If $|\mathcal{N}_{+0}| \neq 0$ set

$$d^k \stackrel{\text{def}}{=} -\left((S_\nu^k)_{\mathcal{N}_{+0}} (\bar{D}_\nu^k)^{-1} (S_\nu^k)^T_{\mathcal{N}_{+0}}\right) \nabla f(x^k), \tag{3}$$

where $\bar{D}_\nu^k \stackrel{\text{def}}{=} diag\big((\max(\lambda_i^k, \rho))_{i\in\mathcal{N}_{+0}}\big)$. If $|\mathcal{N}_{+0}| = 0$ set $d^k = 0$.

b) If $|\mathcal{N}_-| \geq 2$ set

$$r^k \stackrel{\text{def}}{=} -(S_\nu^k)_{\mathcal{N}_-} (S_\nu^k)^T_{\mathcal{N}_-} \nabla f(x^k) \tag{4}$$

otherwise $r^k = 0$.

c) If $|\mathcal{N}_-| \neq 0$ set

$$z^k \stackrel{\text{def}}{=} s_1^k. \tag{5}$$

otherwise $z^k = 0$.

Step 3 *Take a step along these directions using the linesearch algorithm* (LSA1) *or* (LSA2).

a) Let $\Phi_d(t) \stackrel{\text{def}}{=} f(x^k + td^k)$ *and compute* $\Phi'_d(0), \Phi''_d(0)$*. The steplength* σ_d *is given by*

(i) If $\Phi''_d(0) > 0$ *then use* (LSA1) *with*

$$(\sigma_d)_0 = \min\left\{1, -\frac{\Phi'_d(0)}{\Phi''_d(0)}\right\}.$$

(ii) If $\Phi''_d(0) = 0$ *then use* (LSA1) *with*

$$(\sigma_d)_0 = 1.$$

(iii) If $\Phi''_d(0) < 0$ *then use* (LSA2) *with*

$$(\sigma_d)_0 = \begin{cases} \|x^k - x^{k-1}\| & \text{if } k > 0 \\ \|d^k\| & \text{if } k = 0 \end{cases}$$

Set

$$x_d^k \stackrel{\text{def}}{=} x^k + \sigma_d d^k. \tag{6}$$

b) Let $\Phi_r(t) \stackrel{\text{def}}{=} f(x_d^k + tr^k)$ *and compute* $\Phi'_r(0), \Phi''_r(0)$*. Use the same scheme as in a) to get a steplength* σ_r *and set*

$$x_r^k \stackrel{\text{def}}{=} x_d^k + \sigma_r r^k. \tag{7}$$

c) Let $\Phi_z(t) \stackrel{\text{def}}{=} f(x_r^k + tz^k)$ *and compute* $\Phi'_z(0), \Phi''_z(0)$*. Use the same scheme as in a) to get a steplength* σ_z *and define the next iterate*

$$x^{k+1} \stackrel{\text{def}}{=} x_r^k + \sigma_z z^k. \tag{8}$$

Step 4 *If*

$$\|\nabla f(x^{k+1})\| > \epsilon$$

set $k = k+1$ *and continue with* **Step 1**

Step 5 *If*

$$\|\nabla f(x^{k+1})\| \leq \epsilon$$

compute another truncated Lanczos decomposition of the Hessian $\nabla^2 f(x^k)$*, using a maximum of* $\bar{\nu}$ *Lanczos steps and a start vector generated at random. Determine the index sets* $\mathcal{N}_- = \left\{i : \lambda_i^k < 0\right\}$ *and* $\mathcal{N}_{+0} = \left\{i : \lambda_i^k \geq 0\right\}$ *of negative and non-negative ritzvalues.*

If $|\mathcal{N}_-| = 0$ *stop.*

If $|\mathcal{N}_-| \neq 0$ *set* $k = k+1$ *and continue with* **Step 2**. ■

In the algorithm (UCA) we make use of two linesearch algorithms (LSA1) and (LSA2). The first one is the well-known Goldstein-Armijo algorithm (backtracking). The second one is only to be used if the second directional derivative of f with respect to the current direction of descent is negative. In (LSA2) the decrease of the objective will be compared with a concave function (see [Hei85]), instead of a linear one as in (LSA1).

Algorithm 2.2 (LSA1)
Given $x \in \mathcal{D}$, a direction of descent d ($\nabla^T f(x)d \leq 0$), an initial step σ_0 and parameters $0 < \delta_1 < 1$ and $0 < \beta_1 < 1$. Set $j = 0$ and proceed

Step 1 *If $\nabla^T f(x)d = 0$ set $\sigma = 0$ and stop.*

Step 2 *If*

$$f(x) - f(x + (\sigma_0\beta_1^j)d) \geq \delta_1(\sigma_0\beta_1^j)|\nabla^T f(x)d| \tag{9}$$

set $\sigma = \sigma_0\beta_1^j$ and stop

Step 3 *Set $j = j + 1$ and continue with* **Step 2**. ■

Algorithm 2.3 (LSA2)
Given $x \in \mathcal{D}$, a direction of descent $\tilde{v} \neq 0$ with $\nabla^T f(x)\tilde{v} \leq 0$ and $\tilde{v}^T\nabla^2 f(x)\tilde{v} < 0$, an initial step σ_0 and parameters $0 < \delta_2 < \delta_3 < 1$, $0 < \beta_2 < 1$ and $-1 \ll \zeta < 0$. Define

$$\Phi_v(t) \stackrel{\text{def}}{=} f(x + tv)$$

where $v = \frac{\tilde{v}}{\|\tilde{v}\|}$ and compute

$$\Phi'_v = \nabla^T f(x)v \leq 0 \quad \text{and} \quad \Phi''_v = v^T\nabla^2 f(x)v < 0. \tag{10}$$

Define the control function

$$P(\sigma) \stackrel{\text{def}}{=} \sigma\Phi'_v + \frac{1}{2}(\sigma)^2\Phi''_v + \zeta(\sigma)^3. \tag{11}$$

Set $j = 0$ and start the procedure at case A,B or C.

Case A *If*

$$f(x) - f(x + \sigma_0 v) > -\delta_3 P(\sigma_0)$$

continue with **Step A1**.

Step A1 *Set $\sigma_{j+1} = \frac{\sigma_j}{\beta_2}$ and $j = j + 1$.*

Step A2 *If*

$$f(x) - f(x + \sigma_j v) > -\delta_3 P(\sigma_j)$$

continue with **Step A1**.

Step A3 *If*

$$f(x) - f(x + \sigma_j v) > -\delta_2 P(\sigma_j)$$

set $\sigma = \sigma_j$ and stop.

Step A4 *Set* $\sigma_{j+1} = \frac{\sigma_j + \sigma_{j-1}}{2}$ *and* $j = j+1$. *Goto* **Step AB1**.

Case B *If*

$$f(x) - f(x + \sigma_0 v) < -\delta_2 P(\sigma_0)$$

or $x + \sigma_0 v \notin \mathcal{D}$ *continue with* **Step B1**.

Step B1 *Set* $\sigma_{j+1} = \sigma_j \beta_2$ *and* $j = j+1$.

Step B2 *If*

$$f(x) - f(x + \sigma_j v) < -\delta_2 P(\sigma_j)$$

continue with **Step B1**.

Step B3 *If*

$$f(x) - f(x + \sigma_j v) < -\delta_3 P(\sigma_j)$$

set $\sigma = \sigma_j$ *and stop.*

Step B4 *Set* $\sigma_{j+1} = \frac{\sigma_j + \sigma_{j-1}}{2}$ *and* $j = j+1$. *Goto* **Step AB1**.

Case C *If*

$$-\delta_2 P(\sigma_0) \leq f(x) - f(x + \sigma_0 v) \leq -\delta_3 P(\sigma_0)$$

set $\sigma = \sigma_0$ *and stop.*

Step AB1 *If*

$$-\delta_2 P(\sigma_j) \leq f(x) - f(x + \sigma_j v) \leq -\delta_3 P(\sigma_j) \tag{12}$$

set $\sigma = \sigma_j$ *and stop.*

Step AB2 *If*

$$f(x) - f(x + \sigma_j v) > -\delta_3 P(\sigma_j)$$

set

$$\sigma_{j+1} = \sigma_j + \frac{1}{2}|\sigma_{j-1} - \sigma_j|$$

else set

$$\sigma_{j+1} = \sigma_j - \frac{1}{2}|\sigma_{j-1} - \sigma_j|.$$

$j = j+1$ *and continue with* **Step AB1**. ■

Note that (LSA2) distinguishes the cases A,B and C. In case of A the initial steplength ist too short, in case of B it is too long. Only in case of C the initial steplength is taken. The algorithm modifies σ_j in such a way that (12) holds. Figure 1 illustrates the situation.

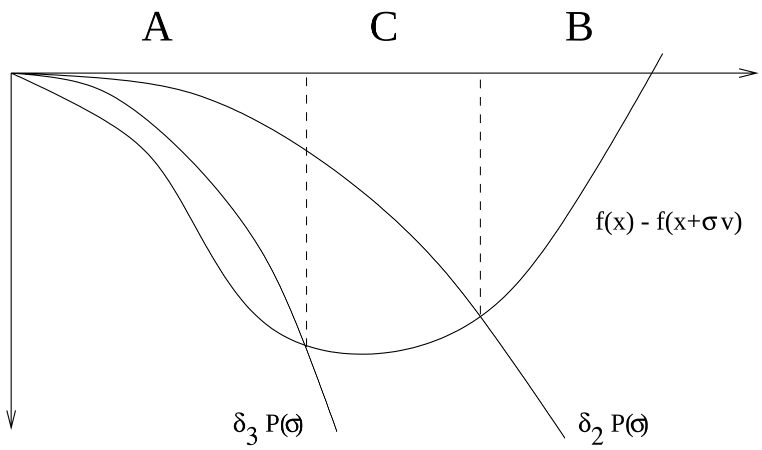

Figure 1 : The decrease in f and control functions $\delta_2 P$, $\delta_3 P$

From (LSA1) we obtain a steplength satifying the priciple of sufficient decrease (9) (see [OR70]) in a finite number of steps. (LSA2) is also a finite algorithm, if the level set $\mathcal{L}_f(f(x^0))$ is compact.

2.2 Convergence analysis

The theoretical results presented here are based on [Hei85] and [OR70]. The basis of the convergence theory are two results on the linesearch algorithms. Using these we prove some lemmas on the directions d, r, z and finally the convergence of algorithm (UCA).

Definition 2.1 *A function $\Phi : \mathbb{R}_{0+} \to \mathbb{R}_{0+}$ is said to be a F-function (forcing function), if for each sequence $(\alpha_i)_{i\to\infty}$ with $\alpha_i \geq 0$ for all $i = 1, 2, \ldots$ the conclusion*

$$\lim_{i\to\infty} \Phi(\alpha_i) = 0 \quad \Rightarrow \quad \lim_{i\to\infty} \alpha_i = 0. \tag{13}$$

holds. ∎

Lemma 2.1 *Let $f \in C^1(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$ and $\mathcal{L}_f(f(x^0))$ compact and Φ_1, Φ_2 are F-functions. If the sequences $(x^k)_{k\to\infty}$, $(d^k)_{k\to\infty}$ and $(\sigma_k)_{k\to\infty}$ satisfy*

$$\nabla^T f(x^k) d^k \leq 0, \tag{14}$$

$$\sigma_k \textit{ from } (\mathrm{LSA1}) \textit{ with the initial step } (\sigma_k)_0, \tag{15}$$

$$\|d^k\|(\sigma_k)_0 \geq \Phi_1\left(\frac{|\nabla^T f(x^k) d^k|}{\|d^k\|}\right), \tag{16}$$

and

$$f(x^{k+1}) \leq f(x^k + \sigma_k d^k) \tag{17}$$

then

$$\lim_{k\to\infty} \frac{\nabla^T f(x^k) d^k}{\|d^k\|} = 0. \tag{18}$$

If moreover

$$\frac{|\nabla^T f(x^k) d^k|}{\|d^k\|} \geq \Phi_2(\|d^k\|(\sigma_k)_0) \tag{19}$$

then

$$\lim_{k\to\infty} \|d^k\|\sigma_k = 0. \tag{20}$$

Proof: The proof of statement (18) is as in Theorem 14.2.25 from [OR70] and (20) follows from (18) by (13). ■

A related lemma can be proven for (LSA2).

Lemma 2.2 *Let $f \in C^2(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$ and $\mathcal{L}_f(f(x^0))$ compact. If the sequences $(x^k)_{k\to\infty}$, $(\tilde{v}^k)_{k\to\infty}$ and $(\sigma_k)_{k\to\infty}$ with*

$$\begin{array}{llll} v^k = \frac{\tilde{v}^k}{\|\tilde{v}^k\|}, & \sigma_k > 0 & \text{from (LSA2)} & \text{if } \|\tilde{v}^k\| > 0 \\ v^k = 0, & \sigma_k = 0 & & \text{if } \|\tilde{v}^k\| = 0 \end{array} \tag{21}$$

are generated as above, and

$$f(x^{k+1}) \leq f(x^k + \sigma_k v^k) \tag{22}$$

then

$$\lim_{k\to\infty} \sigma_k = 0, \tag{23}$$

$$\lim_{k\to\infty} \nabla^T f(x^k) v^k = 0 \tag{24}$$

and

$$\lim_{k\to\infty} (v^k)^T \nabla^2 f(x^k) v^k = 0. \tag{25}$$

Proof: Since $f \in C^2(\mathcal{D})$ and $\mathcal{L}_f(f(x^0))$ is compact, there exists $\mu > 0$ such that

$$f(x) - f(x + \sigma v) \geq -\sigma \nabla^T f(x) v - \frac{1}{2}\sigma^2 v^T \nabla^2 f(x) v - \mu\sigma^3$$

for all x, σ and v with x and $x + \sigma v$ in $\mathcal{L}_f(f(x^0))$. Note that $x^k + \sigma_k v^k \in \mathcal{L}_f(f(x^0))$ by construction and so with (10) we have

$$f(x^k) - f(x^k + \sigma_k v^k) \geq \sigma_k |\Phi'_{v^k}(0)| + \frac{1}{2}(\sigma_k)^2 |\Phi''_{v^k}(0)| - \mu(\sigma_k)^3.$$

From the second inequality of (12) we obtain

$$-\sigma_k(1 - \delta_3)|\Phi'_{v^k}(0)| - \frac{1}{2}(1 - \delta_3)(\sigma_k)^2 |\Phi''_{v^k}(0)| + (\mu - \delta_3\zeta)(\sigma_k)^3 \geq 0$$

and together with (21)

$$\sigma_k \geq \frac{\omega}{2}|\Phi''_{v^k}(0)| + \sqrt{\frac{\omega^2}{4}|\Phi''_{v^k}(0)| + 2\omega|\Phi'_{v^k}(0)|} > 0, \tag{26}$$

whenever $\|v^k\| > 0$, and $\sigma_k = 0$ otherwise. Here $\omega = \frac{2(1-\delta_3)}{\mu-\delta_3\zeta}$.

Because f is bounded from below on $\mathcal{L}_f(f(x^0))$ (12) implies

$$\lim_{k\to\infty} f(x^k) - f(x^k + \sigma_k v^k) = 0.$$

Together with (10) and (21) we have

$$\lim_{k\to\infty} \sigma_k = 0$$

and finally with (26) we obtain

$$\lim_{k\to\infty} \nabla^T f(x^k) v^k = 0 \quad \text{and} \quad \lim_{k\to\infty} (v^k)^T \nabla^2 f(x^k) v^k = 0.$$

■

To state some results concerning the directions of descent we need some information about the second directional derivative in the direction of d, r or z and the ritzvalues computed by the Lanczos algorithm.

Lemma 2.3 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Then the truncated Lanczos decomposition (ν steps) of A is given by*

$$AQ = QT + \beta q_{\nu+1} e_\nu^T \quad \text{and} \quad Q^T Q = I, \tag{27}$$

where $Q \in \mathbb{R}^{n\times\nu}$ and $T, I \in \mathbb{R}^{\nu\times\nu}$. Let

$$T = VDV^T \quad \text{and} \quad V^T V = I \tag{28}$$

be the spectral decomposition of the tridiagonal matrix T. Let $\mathcal{I} \subset \{1,\dots,\nu\}$, $S_{\mathcal{I}} \stackrel{\text{def}}{=} QV_{\mathcal{I}}$, and $d \stackrel{\text{def}}{=} S_{\mathcal{I}} \Delta_{\mathcal{I}}^{-1} S_{\mathcal{I}}^T c$ for an arbitrary $c \in \mathbb{R}^n$ and an arbitrary diagonal matrix $\Delta_{\mathcal{I}} \stackrel{\text{def}}{=} diag(\tilde{\delta}_i)_{i\in\mathcal{I}}$ with positive elements $\tilde{\delta}_i > 0$. Then

$$\begin{aligned} D_{\mathcal{I}} > 0 &\;\Rightarrow\; d^T A d > 0 \\ D_{\mathcal{I}} \ge 0 &\;\Rightarrow\; d^T A d \ge 0 \\ D_{\mathcal{I}} < 0 &\;\Rightarrow\; d^T A d < 0. \end{aligned} \tag{29}$$

Proof: From (28) follows

$$TV_{\mathcal{I}} = (TV)_{\mathcal{I}} = (VD)_{\mathcal{I}} = V_{\mathcal{I}} D_{\mathcal{I}}.$$

Hence with (27)

$$\begin{aligned} AS_{\mathcal{I}} = AQV_{\mathcal{I}} &= QTV_{\mathcal{I}} + \beta q_{\nu+1} e_\nu^T V_{\mathcal{I}} \\ &= QV_{\mathcal{I}} D_{\mathcal{I}} + \beta q_{\nu+1} e_\nu^T V_{\mathcal{I}} \\ &= S_{\mathcal{I}} D_{\mathcal{I}} + \beta q_{\nu+1} e_\nu^T V_{\mathcal{I}}. \end{aligned} \tag{30}$$

Using (30) d^TAd can be written

$$\begin{aligned}
d^TAd &= c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^TAS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \\
&= c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^T\left(S_{\mathcal{I}}D_{\mathcal{I}} + \beta q_{\nu+1}e_\nu^TV_{\mathcal{I}}\right)\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \\
&= c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}\underbrace{S_{\mathcal{I}}^TS_{\mathcal{I}}}_{=I}D_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \;+\; \beta c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tq_{\nu+1}e_\nu^TV_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \qquad (31)\\
&= c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}D_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \;+\; \beta c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}V_{\mathcal{I}}^T\underbrace{Q^Tq_{\nu+1}}_{=0}e_\nu^TV_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc \\
&= c^TS_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}D_{\mathcal{I}}\Delta_{\mathcal{I}}^{-1}S_{\mathcal{I}}^Tc.
\end{aligned}$$

Since $\tilde{\delta}_i > 0$ for all $i \in \mathcal{I}$ (31) completes the proof. ■

These lemmata are very useful in order to analyze the sequence (x^k, d^k, r^k, z^k). First we take a look at the directions d^k.

Lemma 2.4 *Let $f \in C^2(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$, $\mathcal{L}_f(f(x^0))$ compact, and the sequences $(x^k)_{k\to\infty}$, $(d^k)_{k\to\infty}$ and $((\sigma_d)_k)_{k\to\infty}$ be generated by the algorithm* (UCA). *Then*

$$\lim_{k\to\infty}\frac{\nabla^Tf(x^k)d^k}{\|d^k\|} = 0 \qquad (32)$$

and

$$\lim_{k\to\infty}\|d^k\|(\sigma_d)_k = 0. \qquad (33)$$

Proof: From lemma 2.3 we know that $(d^k)^T\nabla^2f(x^k)d^k \geq 0$. Hence $(\sigma_d)_k$ will always be computed by (LSA1). Therefore we verify (32) and (33) using lemma 2.1.

To make use of lemma 2.1 we have to show that the assumptions (14) to (17) and (19) hold. Since (14), (15) and (17) are satisfied by construction, only (16) and (19) have to be shown.

Because $\mathcal{L}_f(f(x^0))$ is compact and $x^k \in \mathcal{L}_f(f(x^0))$ for all k the maximum eigenvalue of the Hessian $\nabla^2f(x^k)$ is bounded from above by some $\bar{\lambda} > 0$. Moreover the minimum diagonal element of the matrix $\bar{D}_\nu^k$ is greater than or equal to $\rho > 0$. Using these two parameters we can estimate

$$\begin{aligned}
\tfrac{1}{\bar{\lambda}+\rho}(-\nabla^Tf(x^k)d^k) &= \sum_{i\in\mathcal{N}_{+0}}\tfrac{(\nabla^Tf(x^k)s_i^k)^2}{(\max(\lambda_i^k,\rho))(\bar{\lambda}+\rho)} \\
&\leq \underbrace{\sum_{i\in\mathcal{N}_{+0}}\frac{(\nabla^Tf(x^k)s_i^k)^2}{(\max(\lambda_i^k,\rho))^2)}}_{\|d^k\|^2} \leq \tfrac{1}{\rho}(-\nabla^Tf(x^k)d^k)
\end{aligned}$$

and

$$\rho \leq \frac{|\nabla^Tf(x^k)d^k|}{\|d^k\|^2} \leq \bar{\lambda} + \rho. \qquad (34)$$

There are two possible initial steps for (LSA1):

i) Let $(\sigma_d)_0 = 1$. Then

$$\frac{|\nabla^T f(x^k)d^k|}{\|d^k\|} = \frac{|\nabla^T f(x^k)d^k|}{\|d^k\|^2}\|d^k\|(\sigma_d)_0$$

and with (34)

$$\rho\|d^k\|(\sigma_d)_0 \le \frac{|\nabla^T f(x^k)d^k|}{\|d^k\|} \le (\bar{\lambda}+\rho)\|d^k\|(\sigma_d)_0. \tag{35}$$

ii) Since $(\sigma_d)_0 = -\frac{\Phi_d'(0)}{\Phi_d''(0)} < 1$

$$\frac{|\nabla^T f(x^k)d^k|}{\|d^k\|} \ge \frac{|\nabla^T f(x^k)d^k|}{\|d^k\|^2}\|d^k\|(\sigma_d)_0.$$

Otherwise we have

$$(\sigma_d)_0\|d^k\| = \frac{|\nabla^T f(x^k)d^k|}{(d^k)^T\nabla^2 f(x^k)d^k}\|d^k\| = \frac{|\nabla^T f(x^k)d^k|}{\|d^k\|}\frac{\|d^k\|^2}{(d^k)^T\nabla^2 f(x^k)d^k}.$$

Hence

$$\begin{aligned}\frac{|\nabla^T f(x^k)d^k|}{\|d^k\|} &= (\sigma_d)_0\|d^k\|\frac{(d^k)^T\nabla^2 f(x^k)d^k}{\|d^k\|^2} \\ &\le \bar{\lambda}(\sigma_d)_0\|d^k\| \;\le\; (\bar{\lambda}+\rho)(\sigma_d)_0\|d^k\|.\end{aligned} \tag{36}$$

Using (34) and (36) we obtain the same estimation as in case i)

$$\rho\|d^k\|(\sigma_d)_0 \le \frac{|\nabla^T f(x^k)d^k|}{\|d^k\|} \le (\bar{\lambda}+\rho)\|d^k\|(\sigma_d)_0. \tag{37}$$

Since $\Phi_1(\alpha) = \frac{1}{\bar{\lambda}+\rho}\alpha$ and $\Phi_2(\alpha) = \rho\alpha$ are F-functions (16) and (19) are satisfied. ∎

Lemma 2.5 *Let $f \in C^2(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$, $\mathcal{L}_f(f(x^0))$ compact, and the sequences $(x^k)_{k\to\infty}$, $(x_d^k)_{k\to\infty}$, $(r^k)_{k\to\infty}$ and $((\sigma_r)_k)_{k\to\infty}$ be generated by the algorithm* (UCA)*. If*

$$\lim_{k\to\infty}(x_d^k - x^k) = 0 \tag{38}$$

then

$$\lim_{k\to\infty}\frac{\nabla^T f(x_d^k)r^k}{\|r^k\|} = \lim_{k\to\infty}\frac{\nabla^T f(x^k)r^k}{\|r^k\|} = 0, \tag{39}$$

$$\lim_{k\to\infty}\|r^k\|(\sigma_r)_k = 0 \tag{40}$$

and

$$\lim_{k\to\infty}\frac{(r^k)^T\nabla^2 f(x_d^k)r^k}{\|r^k\|^2} = 0. \tag{41}$$

Proof: Since it is possible that $\Phi_r''(0) = r^T\nabla^2 f(x_d^k)r^k \geq 0$ we have to distinguish the subsequences $(x^k)_{k\in K_i}$, $(x_d^k)_{k\in K_i}$, $(r^k)_{k\in K_i}$, $((\sigma_r)_k)_{k\in K_i}$ with $i = 1,2$ and $K_1 \stackrel{\text{def}}{=} \{k : \Phi_r''(0) \geq 0\}$, $K_2 \stackrel{\text{def}}{=} \{0,1,2,\ldots\} \setminus K_1$. If $k \in K_1$ then $(\sigma_r)_k$ is computed by (LSA1) and by (LSA2) in case of $k \in K_2$.

Case 1 : (LSA1)

Like in the proof of lemma 2.4 only assumptions (16) and (19) have to be shown in order to make use of lemma 2.1. There are two possible initial steps:

i) Let $(\sigma_r)_0 = 1$. Then

$$\begin{aligned} \frac{|\nabla^T f(x^k) r^k|}{\|r^k\|} &= \frac{|\nabla^T f(x^k)(S_\nu^k)_{\mathcal{N}_-}(S_\nu^k)_{\mathcal{N}_-}^T \nabla f(x^k)|}{\|r^k\|} \\ &= \frac{\|(S_\nu^k)_{\mathcal{N}_-}^T \nabla f(x^k)\|^2}{\|r^k\|} = \frac{\|r^k\|^2}{\|r^k\|} = \|r^k\|, \end{aligned}$$

because the columns of S_ν^k are orthonormal. Hence we obtain

$$\frac{|\nabla^T f(x^k) r^k|}{\|r^k\|} = \|r^k\|(\sigma_r)_0. \tag{42}$$

ii) If $(\sigma_r)_0 = -\frac{\Phi_r'(0)}{\Phi_r''(0)} < 1$, then from (42) there follows

$$\frac{|\nabla^T f(x^k) r^k|}{\|r^k\|} = \|r^k\| > \|r^k\|(\sigma_r)_0. \tag{43}$$

On the other hand

$$\|r^k\|(\sigma_r)_0 = \frac{|\nabla^T f(x_d^k) r^k|}{(r^k)^T \nabla^2 f(x_d^k) r^k}\|r^k\| = \frac{|\nabla^T f(x_d^k) r^k|}{\|r^k\|}\frac{\|r^k\|^2}{(r^k)^T\nabla^2 f(x_d^k) r^k}$$

and with $\bar{\lambda} > 1$ as an upper bound on the eigenvalues of $\nabla^2 f(x)$ on the compact set $\mathcal{L}_f(f(x^0))$ we have

$$\bar{\lambda}\|r^k\|(\sigma_r)_0 \geq \frac{|\nabla^T f(x_d^k) r^k|}{\|r^k\|}. \tag{44}$$

Due to the continuity of the directional derivative and (38) there exists an index $k_1 \in K_1$ such that for all $k \geq k_1$ the estimates (42), (43) and (44) can be combined to

$$\|r^k\|(\sigma_r)_0 \leq \frac{|\nabla^T f(x_d^k) r^k|}{\|r^k\|} \leq \bar{\lambda}\|r^k\|(\sigma_r)_0. \tag{45}$$

Note that $\bar{\lambda}$ was choosen greater then 1. With (45) the assumptions (16) and (19) are satisfied and hence (39) and (40) hold.

To verify (41) we make use of lemma 2.3. With $\tilde{\delta}_i = 1$ for $i \in \mathcal{I}$ and $\mathcal{I} = \mathcal{N}_-$ we know that $(r^k)^T\nabla^2 f(x^k) r^k < 0$. Hence, for all $k \in K_1$

$$\frac{(r^k)^T \nabla^2 f(x^k) r^k}{\|r^k\|^2} \leq 0 \leq \frac{(r^k)^T\nabla^2 f(x_d^k) r^k}{\|r^k\|^2}$$

and from (38)

$$\lim_{k\in K_1} \frac{(r^k)^T \nabla^2 f(x_d^k) r^k}{\|r^k\|^2} = 0.$$

follows.

Case 2 : (LSA2)

This case is much easier to prove. For $k \in K_2$ the steplength $(\sigma_r)_0$ will be computed by (LSA2). The assumptions of lemma 2.2 are satisfied by construction and together with (38) and $f \in C^2(\mathcal{D})$ we obtain (39) to (41) for $k \in K_2$. ∎

Lemma 2.6 *Let $f \in C^2(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$, $\mathcal{L}_f(f(x^0))$ compact, and the sequences $(x^k)_{k\to\infty}$, $(x_r^k)_{k\to\infty}$, $(z^k)_{k\to\infty}$ and $((\sigma_z)_k)_{k\to\infty}$ be generated by the algorithm* (UCA). *If*

$$\lim_{k\to\infty} (x_r^k - x^k) = 0 \tag{46}$$

then

$$\lim_{k\to\infty} \frac{\nabla^T f(x_r^k) z^k}{\|z^k\|} = \lim_{k\to\infty} \frac{\nabla^T f(x^k) z^k}{\|z^k\|} = 0, \tag{47}$$

$$\lim_{k\to\infty} \|z^k\| (\sigma_z)_k = 0 \tag{48}$$

and

$$\lim_{k\to\infty} \frac{(z^k)^T \nabla^2 f(x_r^k) z^k}{\|z^k\|^2} = 0. \tag{49}$$

Proof: The proof of this lemma is analogous to the proof of lemma 2.5. One only has to substitute x_d, r, σ_r, Φ_r by x_r, z, σ_z, Φ_z. ∎

Finally we prove the convergence of the algorithm using the hitherto given lemmata.

Theorem 2.1 *Let $f \in C^3(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty. Let $x^0 \in \mathcal{D}$ and assume $\mathcal{L}_f(f(x^0))$ being compact. Let the sequence $(x^k)_{k\to\infty}$ be generated by the algorithm* (UCA). *If we assume that the minimum eigenvalue of the Hessian $\nabla^2 f(x^k)$ can be approximated sufficiently close by the Lanczos algorithm for all k, then $(x^k)_{k\to\infty}$ has at least one accumulation point and each accumulation point x^* satisfies*

$$\nabla f(x^*) = 0 \tag{50}$$

and

$$\nabla^2 f(x^*) \quad \textit{positive semidefinite.} \tag{51}$$

Proof: Since $\mathcal{L}_f(f(x^0))$ is compact and $(x^k)_{k\to\infty} \subset \mathcal{L}_f(f(x^0))$ an accumulation point x^* exists. Let $K \subset \{0, 1, 2, \ldots\}$ be the index set of the subsequence $(x^k)_{k\in K}$ converging to x^*. Then from lemma 2.4 - 2.6 we know

$$\lim_{k\in K} \nabla^T f(x^k) d^k = \lim_{k\in K} \nabla^T f(x^k) r^k = \lim_{k\in K} \nabla^T f(x^k) z^k = 0 \tag{52}$$

and

$$\lim_{k\in K}(x^k - x_d^k) = \lim_{k\in K}(x^k - x_r^k) = \lim_{k\in K}(x_d^k - x_r^k) = 0. \tag{53}$$

To estimate $\|\nabla f(x^*)\|$ from above we first have a look at

$$\|(S_\nu^k)^T\nabla f(x^k)\|^2 = \underbrace{\left((s_1^k)^T\nabla f(x^k)\right)^2}_{(1)} + \underbrace{\left\|(S_\nu^k)^T_{\mathcal{N}_-}\nabla f(x^k)\right\|^2}_{(2)} + \underbrace{\left\|(S_\nu^k)^T_{\mathcal{N}_{+0}}\nabla f(x^k)\right\|^2}_{(3)}. \tag{54}$$

If any of the index sets $\mathcal{N}_-$ or $\mathcal{N}_{+0}$ are empty the corresponding summand can be omited.

(1): $\left((s_1^k)^T\nabla f(x^k)\right)^2 = \left|(z^k)^T\nabla f(x^k)\right|^2$.

(2): $\left\|(S_\nu^k)^T_{\mathcal{N}_-}\nabla f(x^k)\right\|^2 = \left|\nabla^T f(x^k)(S_\nu^k)_{\mathcal{N}_-}(S_\nu^k)^T_{\mathcal{N}_-}\nabla f(x^k)\right|^2 = \left|(r^k)^T\nabla f(x^k)\right|^2$

(3): $\frac{1}{\bar\lambda+\rho}\left\|(S_\nu^k)^T_{\mathcal{N}_{+0}}\nabla f(x^k)\right\|^2 \le \left|\nabla^T f(x^k)(S_\nu^k)_{\mathcal{N}_{+0}}(S_\nu^k)^T_{\mathcal{N}_{+0}}\nabla f(x^k)\right|^2 = \left|(d^k)^T\nabla f(x^k)\right|^2$

Using this (54) becomes

$$\|(S_\nu^k)^T\nabla f(x^k)\|^2 \le (\bar\lambda+\rho)\left|(d^k)^T\nabla f(x^k)\right|^2 + \left|(r^k)^T\nabla f(x^k)\right|^2 + \left|(z^k)^T\nabla f(x^k)\right|^2.$$

Hence

$$\lim_{k\in K}\|(S_\nu^k)^T\nabla f(x^k)\| = 0. \tag{55}$$

By definition $S_\nu^k = Q_\nu^k V_\nu^k$. Since V_ν^k is a unitary matrix we obtain

$$\|(S_\nu^k)^T\nabla f(x^k)\| = \|(V_\nu^k)^T(Q_\nu^k)^T\nabla f(x^k)\| = \|(Q_\nu^k)^T\nabla f(x^k)\|. \tag{56}$$

The first column of the matrix Q_ν^k is the normalized initial vector of the Lanczos iteration $-\nabla f(x^k)$ (see **Step 1** of (UCA)). Therefore

$$\|(Q_\nu^k)^T\nabla f(x^k)\| = \sqrt{\left(\frac{\nabla^T f(x^k)\nabla f(x^k)}{\|\nabla f(x^k)\|}\right)^2 + \sum_{i=2}^{\nu}\left((q_i^k)^T\nabla f(x^k)\right)^2}$$

holds. Hence from (55) and (56) follows

$$\lim_{k\in K}\|\nabla f(x^k)\| = 0.$$

To verify (51) only the directions z^k are needed, as we have assumed that λ_1^k from the truncated Lanczos decomposition in **Step 1** or **Step 5** of (UCA) approximates the minimum eigenvalue of the Hessian accurately. From (49) and (46) we get (51) because

$$(z^k)^T\nabla^2 f(x^k)z^k = \min\{\lambda_{\min}(\nabla^2 f(x^k)), 0\}.$$

∎

Remark 2.1 *To proof the previous theorem we needed an assumption on the convergence of the minimum ritzvalue λ_1^k against the minimum eigenvalue of the Hessian $\nabla^2 f(x^k)$. A very interesting result is given in [GL89]. Theorem 9.2.2 implies that λ_1^k is a good approximation to the minimum eigenvalue whenever the first Lanczos vector q_1^k of the Lanczos algorithm and the normalized eigenvector z_1 to the minimum eigenvalue are not orthogonal. The approximation is the better, the nearer $(z_1)^T q_1^k$ is by* 1. *Therefore we use a random start vector in* **Step 5** *of* (UCA).

Nevertheless we cannot prevent a gab between the minimum eigenvalue of the Hessian of the objective function ($\lambda_{\min}(\nabla^2 f(x))$) and the minimum ritzvalue λ_1, because we truncate the computation of the Lanczos decomposition after at last ν steps. But usually λ_1 is a good approximation for $\lambda_{\min}(\nabla^2 f(x))$ and so an existing negative curvature direction can be identified in most cases. ■

3 Bound constrained problems

The generalization from the unconstrained case to the bound constrained one is not difficult. To specify the second order necessary condition we need the projection of the gradient on the box defined by Ω_{B}. We define it componentwise by

$$(P(\nabla f(x)))_i \stackrel{\text{def}}{=} \begin{cases} \min\{0, (\nabla f(x))_i\} & \text{if } x_i = l_i \\ (\nabla f(x))_i & \text{if } l_i < x_i < u_i \\ \max\{0, (\nabla f(x))_i\} & \text{if } x_i = u_i. \end{cases} \tag{57}$$

The first order necessary condition is fulfilled in a feasible point x^* if

$$P(\nabla f(x^*)) = 0. \tag{58}$$

Such a point is called a stationary point. Moreover, if the reduced Hessian

$$I_{\mathcal{F}(x^*)}^T \nabla^2 f(x^*) I_{\mathcal{F}(x^*)} \tag{59}$$

is positive semidefinite, then the second order necessary condition holds in x^*. The index set $\mathcal{F}$ is the set of the free variables

$$\mathcal{F}(x^*) \stackrel{\text{def}}{=} \{i \in \{1, \ldots, n\} \ : \ l_i < x_i < u_i\}.$$

The complementary set of $\mathcal{F}(x^*)$

$$\mathcal{A}(x^*) \stackrel{\text{def}}{=} \{i \in \{1, \ldots, n\} \ : \ x_i = l_i \text{ or } x_i = u_i\}.$$

is the so-called active set. Note, that if the strict complementarity condition

$$(\nabla f(x))_i < 0 \ \text{ if } \ x_i = l_i \quad \text{and} \quad (\nabla f(x))_i > 0 \ \text{ if } \ x_i = u_i \tag{60}$$

is fulfilled in a point x^* satisfying (58), then the positive definiteness of (59) is sufficient for x^* being a strict local minimizer.

The following algorithm based upon (UCA) can only find a point satisfying the second order necessary condition. To specify the algorithm (BCA) the projection on the feasible region

$$(P_\Omega(x))_i \stackrel{\text{def}}{=} \begin{cases} l_i & \text{if } x_i \le l_i \\ u_i & \text{if } x_i \ge u_i \\ x_i & \text{otherwise.} \end{cases} \tag{61}$$

is needed.

Instead of the projected gradient (57) we make use of the projection on the box (61) to define an equivalent condition to (58). It is easy to see that

$$x^* - P_\Omega(x^* - \nabla f(x^*)) = 0 \quad \Leftrightarrow \quad P(\nabla f(x^*)) = 0. \tag{62}$$

To verify the condition $\|x^* - P_\Omega(x^* - \nabla f(x^*))\| = 0$ has some numerical advantages, because $(x - P_\Omega(x - \nabla f(x)))_i$ is continuous in l_i and u_i.

To describe an algorithm for solving (UCP) we need the following index sets

Definition 3.1 *In a given point $x \in \Omega_B$*

$$\tilde{\mathcal{F}}(x) \stackrel{\text{def}}{=} \left\{ i \in \{1,\dots,n\} \;:\; \begin{array}{rl} l_i < x_i < u_i & \text{or} \\ x_i = l_i \text{ and } (\nabla f(x))_i < 0 & \text{or} \\ x_i = u_i \text{ and } (\nabla f(x))_i > 0 & \end{array} \right\} \tag{63}$$

is the augmented set of free variables and its complement

$$\mathcal{B}(x) \stackrel{\text{def}}{=} \left\{ i \in \{1,\dots,n\} \;:\; \begin{array}{rl} x_i = l_i \text{ and } (\nabla f(x))_i \ge 0 & \text{or} \\ x_i = u_i \text{ and } (\nabla f(x))_i \le 0 & \end{array} \right\} \tag{64}$$

is the so-called binding set. ∎

Moreover, to define a quasi unconstrained optimization problem on a face of the box we need a reduced function.

Definition 3.2 *Let $x \in \Omega_B$ be given and let $\mathcal{J}(x)$ be an index set associated with x, then we define the $\mathcal{J}(x)$-reduced function $f_{\mathcal{J}(x)}(y) : \mathbb{R}^{|\mathcal{J}(x)|} \to \mathbb{R}$ by*

$$f_{\mathcal{J}(x)}(y) \stackrel{\text{def}}{=} f(x_{\mathcal{J}(x)}(y)) \quad \text{with} \quad x_{\mathcal{J}(x)}(y) \stackrel{\text{def}}{=} \begin{cases} (I_{\mathcal{J}(x)}y)_i & \text{if } i \in \mathcal{J}(x) \\ x_i & \text{otherwise} \end{cases}$$

∎

Remark 3.1 *Note that the derivatives of the reduced function are given by*

$$\nabla f_{\mathcal{J}(x)}(y) = I_{\mathcal{J}(x)}^T \nabla f(x_{\mathcal{J}(x)}(y)) \quad \text{and} \quad \nabla^2 f_{\mathcal{J}(x)}(y) = I_{\mathcal{J}(x)}^T \nabla^2 f(x_{\mathcal{J}(x)}(y)) I_{\mathcal{J}(x)}.$$

∎

3.1 Outline of the algorithm

Algorithm 3.1 (BCA)
Given the starting point $x^0 \in \Omega_B$ and parameters $0 < \epsilon \ll 1$, $0 < \eta_1, \eta_2 < 1$ and $2 \le l \in \mathbb{N}$. Set $k = 0$ and proceed

Step 0 *If*

$$\|x^k - P_\Omega(x^k - \nabla f(x^k))\| \le \epsilon \tag{65}$$

continue with **Step 11**.

Step 1 *Determine the binding set $\mathcal{B}(x^k)$ and the active set $\mathcal{A}(x^k)$ in the current iterate x^k. If*

$$\mathcal{A}(x^k) = \mathcal{B}(x^k)$$

then goto **Step 5**. *Otherwise set $k_g = k$.*

Step 2 *Compute a steplength σ_g along the polygonal search path $P_\Omega(x^k - t\nabla f(x^k))$ using the linesearch algorithms* (LSA1) *and* (LSA2).

Let $\Phi(t, \nabla f(x^k)) \overset{\text{def}}{=} f(P_\Omega(x^k - t\nabla f(x^k)))$ and compute

$$\Phi'_{\nabla f(x^k)}(0) = -\nabla^T f(x^k) P(\nabla f(x^k)) = -\|P(\nabla f(x^k))\|^2 \tag{66}$$

and

$$\Phi''_{\nabla f(x^k)}(0) = P(\nabla f(x^k))^T \nabla^2 f(x^k) P(\nabla f(x^k)). \tag{67}$$

To get σ_g use the following scheme

(i) If $\Phi''_{\nabla f(x^k)}(0) > 0$ then use (LSA1) *with*

$$(\sigma_g)_0 = \min\left\{1, -\frac{\Phi'_{\nabla f(x^k)}(0)}{\Phi''_{\nabla f(x^k)}(0)}\right\}.$$

(ii) If $\Phi''_{\nabla f(x^k)}(0) = 0$ then use (LSA1) *with*

$$(\sigma_g)_0 = 1.$$

(iii) If $\Phi''_{\nabla f(x^k)}(0) < 0$ then use (LSA2) *with*

$$(\sigma_g)_0 = \begin{cases} \|x^k - x^{k-1}\| & \text{if } k > 0 \\ \|P(\nabla f(x^k))\| & \text{if } k = 0 \end{cases}$$

Set

$$x^{k+1} \overset{\text{def}}{=} P_\Omega(x^k - \sigma_g \nabla f(x^k)).$$

Step 3 *If*

$$\|x^{k+1} - P_\Omega(x^{k+1} - \nabla f(x^{k+1}))\| \le \epsilon \tag{68}$$

continue with **Step 11**.

Step 4 *If*

$$\mathcal{A}(x^{k+1}) = \mathcal{A}(x^k) = \cdots = \mathcal{A}(x^{k+1-l})$$

or

$$f(x^k) - f(x^{k+1}) \leq \eta_1 \max_{k_g \leq j < k} \{f(x^j) - f(x^{j+1})\}$$

then set $k = k+1$ *and continue with* **Step 5**. *Else set* $k = k+1$ *and goto* **Step 1**.

Step 5 *Set* $\mathcal{F} = \mathcal{F}(x^k)$ *and define the* $\mathcal{F}$*-reduced function* $f_{\mathcal{F}}(y)$. *Set* $i = 0$ *and* $y^0 = I_{\mathcal{F}}^T x^k$.

Step 6 *Use* **Step 1** *of* (UCA) *with* $f_{\mathcal{F}}$ *replacing* f.

Step 7 *Use* **Step 2** *of* (UCA) *with* $f_{\mathcal{F}}$ *replacing* f.

Step 8 *Use* **Step 3** *of* (UCA) *with* $f_{\mathcal{F}}$ *replacing* f *and obtain* y^{i+1}.

Step 9 *Set*

$$x^{k+1} = P_\Omega(x_{\mathcal{F}}(y^{i+1})).$$

If

$$\|x^{k+1} - P_\Omega(x^{k+1} - \nabla f(x^{k+1}))\| \leq \epsilon \tag{69}$$

continue with **Step 11**.

Step 10 *If*

$$\mathcal{A}(x^{k+1}) \neq \mathcal{A}(x^k)$$

or

$$f_{\mathcal{F}}(y^i) - f_{\mathcal{F}}(y^{i+1}) \leq \eta_2 \max_{0 \leq j < i} \{f_{\mathcal{F}}(y^j) - f_{\mathcal{F}}(y^{j+1})\}$$

or

$$\|\nabla f_{\mathcal{F}}(y^{i+1})\| \leq \epsilon$$

then set $k = k+1$ *and continue with* **Step 1**. *Otherwise set* $k = k+1$, $i = i+1$ *and proceed with* **Step 6**.

Step 11 *As in the unconstrained case compute the truncated Lanczos decomposition for the* $\tilde{\mathcal{F}}(x^k)$*-reduced function* $f_{\tilde{\mathcal{F}}(x^k)}$ *to determine the index sets* $\mathcal{N}_- = \left\{i : \lambda_i^k < 0\right\}$ *and* $\mathcal{N}_{+0} = \left\{i : \lambda_i^k \geq 0\right\}$ *of negative and non-negative ritzvalues of the reduced Hessian.*

If $|\mathcal{N}_-| = 0$ *stop.*

If $|\mathcal{N}_-| \neq 0$ *then continue with* **Step 7** *replacing* $f_{\mathcal{F}}$ *by* $f_{\tilde{\mathcal{F}}(x^k)}$. ■

Remark 3.2 *The algorithm* (BCA) *is a combination of the gradient projection method and the unconstrained minimizer* (UCA). *Thereby the gradient projection steps have the job to find the correct set of active variables by multiple activation/inactivation and* (UCA) *explores a face of the box defined by* Ω_B. ■

Remark 3.3 *Note that the algorithm* (BCA) *only stops in* **Step 11** *which only can be reached if a stationary point was identified by (65), (68) or (69). The replacement of $f_{\mathcal{F}}$ by $f_{\tilde{\mathcal{F}}(x^k)}$ is a special feature to determine directions of negative curvature in a stationary point not satisfying the strict complementarity condition (60). For the convergence result presented below it is irrelevant, since we assume (60) hold in x^*. This guarantees identification of the correct active set in a finite number of steps, and $\mathcal{F}(x^*) = \tilde{\mathcal{F}}(x^*)$.* ∎

Remark 3.4 *The first and second directional derivative of f in the direction of the projected gradient defined by (66) and (67) exists, because*

$$P_\Omega(x^k - t\nabla f(x^k)) = x^k - tP(\nabla f(x^k))$$

for $0 \leq t \leq \beta_1$, where β_1 is the first breakpoint of the piecewise linear path, if $P(\nabla f(x^k)) \neq 0$. ∎

3.2 Convergence analysis

The transfer of the convergence properties from theorem 2.1 to the bound constrained case follows from theorem 5.4. in [CM87]. Calamai and Moré refer to an algorithm combining the projected gradient steps with another method satisfying $x^{k+1} \in \Omega_B$, $f(x^{k+1}) \leq f(x^k)$ and $\mathcal{A}(x^{k+1}) \subset \mathcal{A}(x^k)$. Theorem 5.4. in [CM87] shows that a bounded sequence generated by this algorithm stays on the correct active set after a finite number of steps, whenever the strict complementary condition (60) holds in each stationary point.

Theorem 3.1 *Let $f \in C^3(\mathcal{D})$, $\mathcal{D}$ open, convex and nonempty, $x^0 \in \mathcal{D}$ and $\mathcal{L}_f(f(x^0)) \cap \Omega_B$ compact. Let the sequence $(x^k)_{k\to\infty}$ be generated by the algorithm* (BCA). *We assume that the minimum eigenvalue of the reduced Hessian can be approximated sufficiently good by the Lanczos algorithm in* **Step 11**.

If the strict complementarity condition (60) is satisfied in each stationary point of (BCP), then $(x^k)_{k\to\infty}$ has at least one accumulation point and each accumulation point x^* satisfies

$$P(\nabla f(x^*)) = 0 \tag{70}$$

and

$$I^T_{\tilde{\mathcal{F}}(x^*)} \nabla^2 f(x_{\tilde{\mathcal{F}}(x^*)}(y)) I_{\tilde{\mathcal{F}}(x^*)} \quad \textit{positive semidefinite.} \tag{71}$$

Proof: If we assume that (60) holds in each stationary point, then in **Step 11** $\tilde{\mathcal{F}}(x^k)$ can be replaced by $\mathcal{F}(x^k)$ and so the assumptions of theorem 5.4. from [CM87] are satisfied.

Hence, the bounded sequence $(x^k)_{k\to\infty}$ remains on one face of the box for all k sufficiently large. By construction this only happens if $\mathcal{A}(x^k) = \mathcal{B}(x^k)$ is satisfied for k sufficiently large. Hence we can use theorem 2.1 to complete the proof. ∎

4 Implementation

Now we have a look at the implementation of (BCA). In the algorithms we make use of the Lanczos decomposition of the Hessian or the reduced Hessian of the objective function. Here we present two implemented features which serve to increase stability and effectiveness.

4.1 Partial reorthogonalization

The Lanczos decomposition of a symmetric matrix A in exact arithmetic can be written concisely as

$$\begin{aligned} AQ_\nu - Q_\nu T_\nu &= \beta_{\nu+1} q_{\nu+1} e_\nu^T \\ Q_\nu^T Q_\nu &= I_\nu . \end{aligned} \tag{72}$$

In practical computing the columns of Q_ν, the so-called Lanczos vectors, lose their orthogonality. As a consequence of this $Q_\nu^T Q_\nu \neq I_\nu$ even for $\nu \ll n$. But without the orthogonality lemma 2.3 does not hold and the algorithm may fail.

This problem is well known and described in detail in [Par80a]. The only way to stabilize the computation of (72) is to orthogonalize the current computed Lanczos vector relatively to the previous ones. To use all the Lanczos vectors for reorthogonalization is very expensive and not necessary. Therefore Parlett developed in [Par80a], [Par80b] or [PS79] a strategy called "Selective Orthogonalization (SO)". The (SO) method determines exactly the ritzvectors relative to which the current Lanczos vector loses orthogonality and reorthogonalizes it.

Another way to prevent the breakdown was presented in [Sim84b] and [Sim84a]. Simons idea is to simulate the loss of orthogonality and to decide relative to which of the older Lanczos vectors a new one should be corrected. The foundation of the simulation is the following theorem proved in [Sim84a].

Lemma 4.1 *Let $\omega_{i,j} = q_i^T q_j$. Then the $\omega_{i,k}$ satisfy the recurrence*

$$\begin{aligned} \omega_{k,k} &= 1 \qquad \text{for } k = 1,\dots,j+1 , \\ \omega_{k,k-1} &= q_k^T q_{k-1} \qquad \text{for } k = 2,\dots,j+1 , \\ \beta_{j+1}\omega_{j+1,k} &= \beta_{k+1}\omega_{j,k+1} + (\alpha_k - \alpha_j)\,\omega_{j,k} + \beta_k \omega_{j,k-1} - \beta_j \omega_{j-1,k} \\ &\quad + q_j^T f_k - q_k^T f_j , \end{aligned} \tag{73}$$

for $1 \le k < j$ and $\omega_{j,k+1} = \omega_{k+1,j}$, where $\omega_{k,0} \equiv 0$ and the vectors f_i account for the rounding errors in the i-th step.

Proof: The recurrence (73) follows from the three-term-recurrence of the Lanczos algorithm by adding an error vector f. For further details see [Sim84a]. ∎

Introducing some reasonable estimates for the error-vectors it is possible to estimate the $\omega_{j,k}$'s by

$$\begin{aligned} \omega_{k,k} &= 1 \quad \text{for } k = 1,\dots,j+1 , \\ \omega_{k,k-1} &= \phi_k \quad \text{for } k = 2,\dots,j+1 , \\ \omega_{j+1,k} &= \tfrac{1}{\beta_{j+1}} \left(\beta_{k+1}\omega_{j,k+1} + (\alpha_k - \alpha_j)\,\omega_{j,k} + \beta_k\omega_{j,k-1} - \beta_j\omega_{j-1,k}\right) \\ &\quad + \theta_{j,k} , \end{aligned} \tag{74}$$

for $1 \leq k < j$ and $\omega_{j,k+1} = \omega_{k+1,j}$. For ϕ_k and $\theta_{j,k}$ Simon proposed

$$\phi_k = \epsilon_M n \frac{\beta_2}{\beta_k} \Phi ,$$

where $\Phi \in (-0.3, 0.3)$ and

$$\theta_{j,k} = \epsilon_M(\beta_{k+1} + \beta_{j+1})\Theta ,$$

where $\Theta \in (-0.6, 0.6)$. ϵ_M is the machine precision. An algorithm called (PRO) based on (74) is presented in [Sim84b] or [Fel96].

The values Φ and θ are generated at random and so the algorithm (PRO) may over- or underestimate the real loss of orthogonality. To prevent an underestimation we check the real values of $q_i^T q_j$ from time to time and, if necessary, we switch to the full reorthogonalization strategy. Numerical experiments showed, that this is more efficient then (SO) proposed by Parlett.

4.2 Krylov iteration

The reason why we make use of the Lanczos iteration is the good approximation of the outer eigenvalues by the ritzvalues. This property is very important, whenever directions of negative curvature exist. But if we suppose, that the reduced Hessian of the objective is positive definite, for example in a neighborhood of the solution, then the Lanczos algorithm can be interpreted as a Krylov subspace iteration to solve the Newton equation

$$\nabla^2 f_{\mathcal{F}}(x^k)d = -\nabla f_{\mathcal{F}}(x^k). \tag{75}$$

With the Lanczos method we obtain a good approximation of the Newton direction

$$d^k = -Q_\nu^k (T_\nu^k)^{-1} Q_\nu^k \nabla f_{\mathcal{F}}(x^k) \tag{76}$$

whenever $|\mathcal{N}_-| = 0$. To compute such a truncated Newton direction we do neither need the ritzvalues nor the ritzvectors.

If the ritzvalues are not of interest and only the solution of the symmetric linear system (75) is needed, it is possible to update an initial guess for the Newton direction without the computation of the ritzvectors or -values. Saad presents in [Saa96] a method founded on the Gaussian elimination without pivoting. Unfortunately this method may be sometimes problematic, because it prones to breakdowns.

We used the more sophisticated method called (SYMMLQ) proposed by Paige and Saunders in [PS75]. There the more complex LQ factorization of the tridiagonal matrix T_ν^k is adapted step by step with the truncated Newton direction d computed by forward- and backward substitution.

4.3 Control of the Lanczos steps

Whenever the reduced Hessian of the objective function is positive definite and the Lanczos decomposition is used as an iterative solver for the Newton equation (75), we are able to control the maximum number of computed Lanzcos vectors $\bar{\nu}$. Eisenstat and Walker showed in [EW96] that the inexact Newton method still converges, if the approximated Newton direction d satisfies

$$\|\nabla^2 f_{\mathcal{F}}(x^k)d + \nabla f_{\mathcal{F}}(x^k)\| \leq \mu_k \|\nabla f_{\mathcal{F}}(x^k)\|. \tag{77}$$

Here μ_k is a so-called forcing term and can be choosen

$$\mu_k \stackrel{\text{def}}{=} \gamma \min\left(1.0, \left(\frac{\|\nabla f_{\mathcal{F}}(x^k)\|}{\|\nabla f_{\mathcal{F}}(x^{k-1})\|}\right)^{\alpha}\right) \tag{78}$$

for some $0 \leq \gamma \leq 1$ and $1 < \alpha \leq 2$. We used $\gamma = 0.9$ and $\alpha = 2$.

To obtain a "better" approximation of the Newton direction, we increase the maximum number of Lanczos vectors $\bar{\nu}$ by 5 if (77) is violated. Furthermore we reduce it by 3, whenever

$$\|\nabla^2 f_{\mathcal{F}}(x^k)d + \nabla f_{\mathcal{F}}(x^k)\| \leq 0.1\mu_k \|\nabla f_{\mathcal{F}}(x^k)\|$$

holds.

A lower bound on $\bar{\nu}$ was given by

$$\min\left(|\mathcal{F}|, \min\left(30, \max(10, 10 + (\frac{\ln(|\mathcal{F}|)}{\ln(10)} - 2) \cdot 10)\right)\right)$$

and an upper bound by

$$\min\left(|\mathcal{F}|, \min\left(80, \max(30, 30 + (\frac{\ln(|\mathcal{F}|)}{\ln(10)} - 2) \cdot 25)\right)\right).$$

E.g. in case of $|\mathcal{F}| \geq 10000$ the lower bound is 30 and the upper bound 80.

4.4 Termination criteria

Computation terminates in the following cases

`es`	Description
-3	The number of iterations was greater then `maxiter`. The iteration counter increases when the algorithm (BCA) reaches **Step 1**.
-2	One of the linesearch algorithms was not able to find a steplength $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ in **Step 8** of (BCA).
-1	One of the linesearch algorithms was not able to find a steplength $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ in **Step 2** of (BCA).
1	$\|x^k - P_\Omega(x^k - \nabla f(x^k))\| \leq \epsilon$ and the approximation of the minimum eigenvalue $\lambda_{\min}$ was greater then 0 in **Step 11**.
2	$\|x^k - x^{k-1}\| \leq \epsilon_x(\|x^k\| + 1)$ for K_x consecutive iterates x^k.
3	$f(x^{k-1} - f(x^k) \leq \epsilon_f(\|f(x^k)\| + \sqrt{\epsilon_M})$ for the last K_f iterates x^k in follow.

4.5 Choice of the parameters

The parameters were choosen as follows

Name	Description	Value
ϵ_M	Machine precision	10^{-16}
ρ	Regularity threshold in (UCA)	10^{-10}
ϵ	Termination parameter in (UCA) and (BCA)	$10^{-6}\sqrt{n}$
η_1	Termination parameter for the projected gradient steps	0.1
η_2	Termination parameter for the second order steps	0.1
l	Maximum number of projected gradient steps without changing the active set	3
δ_1	"Armijo"-parameter used in (LSA1)	10^{-4}
β_1	Backtracking-parameter used in (LSA1)	0.5
δ_2	Parameter used in (LSA2)	0.1
δ_3	Parameter used in (LSA2)	0.9
β_2	Backtracking-parameter used in (LSA2)	0.5
ζ	Safeguard in (LSA2)	-10^{-6}
$\sigma_{\min}$	Minimum steplength	10^{-10}
$\sigma_{\max}$	Maximum steplength	10^{4}
ϵ_x	Termination parameter. See `es` = 2	$10^{3}\sqrt{\epsilon_M}$
K_x	Termination parameter. See `es` = 2	10
ϵ_f	Termination parameter. See `es` = 3	$\sqrt{\epsilon_M}$
K_f	Termination parameter. See `es` = 3	10
`maxiter`	Maximum number of iterations. See `es` = -3	500

5 Numerical results

To test the algorithm (BCA) we use the CONSTRAINED AND UNCONSTRAINED TESTING ENVIRONMENT CUTE. CUTE containes a lot of unconstrained and bound constrained programming problems, coded in a special format called SIF and the necessary tools for decoding the SIF Files and evaluating the objective and the constraints. For more details see [BCGT97].

The computations were performed under LINUX on an Pentium II with 300/66 MHz and 256 MB main memory. The FORTRAN code (double precision) was compiled with g77 -O6 -static. The given data are

Name	Description
`name`	Name of the problem.
`c`	Classification of the problem. The problem is unconstrained, if a `U` is given and bounded if a `B` is given.
`e`	Termination status, as described in subsection 4.4.
`it`	Number of main iterations. The iteration counter increases when the algorithm (BCA) reaches **Step 1**.
`n`	Number of variables. Dimension of x.
`nb`	Number of active variables in the final point x^e of the iteration.
`nf`	Number of free variables in the final point x^e of the iteration.
`#f`	Number of objective function evaluations.
`#gf`	Number of gradient evaluations.
`#hf`	Number of Hessian evaluations.
`#hfv`	Number of reduced Hessian vector products.
`%den`	Number of nonzeros in the Hessian in percent of n^2.
`norm pgf`	Norm of the projected gradient of the objective $\|x^e - P_\Omega(x^e - \nabla f(x^e))\|$ in the final point of the interation.
`lam min`	Final minimum eigenvalue approximation of the reduced Hessian computed in **Step 11** if `es` $= 1$ or **Step 6** else.
`s`	Strict complementary condition indicator.
`obj.fun`	Objective function value $f(x^e)$ in the final point.
`time`	CPU-time in seconds.

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
3PK	U	1	5	30	0	30	21	17	13	263	47.8	0.1246E-06	0.3657E-04	T	0.1720E+01	.62E-01
AKIVA	U	1	3	2	0	2	9	8	7	25	100.0	0.2579E-10	0.1775E+01	T	0.6166E+01	.23E-01
ALLINIT	B	1	3	4	1	3	15	11	8	33	100.0	0.2502E-11	0.1350E+02	T	0.1671E+02	.16E-01
ALLINITU	U	1	3	4	0	4	17	11	9	42	100.0	0.4783E-13	0.6210E+01	T	0.5744E+01	.16E-01
ARGLINA	U	1	1	100	0	100	2	2	2	4	100.0	0.1887E-12	0.2000E+01	T	0.1000E+03	.32E+01
ARGLINB	U	-2	1	10	0	10	57	4	3	27	100.0	0.3183E-04	0.0000E+00	T	0.4634E+01	.23E-01
ARGLINC	U	-2	2	10	0	10	35	3	3	11	64.0	0.4501E-09	0.4941E-11	T	0.6135E+01	.00E+00
ARWHEAD	U	1	3	100	0	100	7	7	7	29	3.0	0.1173E-12	0.1200E+02	T	0.0000E+00	.16E-01
BARD	U	1	4	3	0	3	20	13	11	51	100.0	0.6800E-10	0.7411E-02	T	0.8215E-02	.16E-01
BDEXP	B	1	4	100	0	100	18	16	16	207	4.9	0.7863E-06	0.6188E-09	T	0.6118E-06	.70E-01
BDQRTIC	U	1	4	100	0	100	11	11	11	178	8.8	0.2583E-06	0.2242E+01	T	0.3788E+03	.86E-01
BEALE	U	1	4	2	0	2	11	10	9	33	100.0	0.1316E-09	0.3015E+00	T	0.3931E-20	.78E-02
BIGGS6	U	1	10	6	0	6	126	68	53	383	100.0	0.1033E-08	0.9351E-05	T	0.3763E-16	.78E-01
BIGGSB1	B	1	27	1000	2	998	1878	1217	556	30385	0.3	0.3153E-05	0.7269E-03	T	0.1500E-01	.66E+02
BOX3	U	1	3	3	0	3	11	10	9	42	100.0	0.1318E-09	0.9116E-03	T	0.5887E-18	.16E-01
BQP1VAR	B	1	1	1	1	0	2	2	1	3	100.0	0.0000E+00	no r.Hes.	T	0.0000E+00	.78E-02
BQPGABIM	B	1	3	50	14	36	6	5	5	75	11.8	0.1749E-06	0.5170E+02	T	-.3790E-04	.31E-01
BQPGASIM	B	1	4	50	10	40	6	5	5	68	11.8	0.4186E-06	0.4964E+02	T	-.5520E-04	.16E-01
BQPGAUSS	B	3	49	2003	94	1909	468	305	180	11217	0.4	0.3299E-04	0.2504E+00	T	-.3626E+00	.13E+03
BRKMCC	U	1	2	2	0	2	4	4	4	14	100.0	0.5880E-12	0.1944E+01	T	0.1690E+00	.78E-02
BROWNAL	U	1	3	10	0	10	8	8	8	33	100.0	0.2614E-07	0.1346E-01	T	0.1496E-15	.23E-01
BROWNBS	U	1	3	2	0	2	19	8	7	24	100.0	0.0000E+00	0.2000E+01	T	0.0000E+00	.78E-02
BROWNDEN	U	1	4	4	0	4	9	9	9	52	100.0	0.3018E-09	0.1231E+04	T	0.8582E+05	.23E-01
BROYDN7D	U	1	4	10	0	10	26	17	15	186	54.0	0.2823E-11	0.2196E+01	T	0.1261E+01	.31E-01
BRYBND	U	1	3	10	0	10	14	12	10	116	88.0	0.1753E-08	0.9650E+00	T	0.1771E-19	.23E-01
CAMEL6	B	1	4	2	0	2	13	10	10	38	100.0	0.7668E-08	0.7040E+01	T	-.1032E+01	.78E-02
CHAINWOO	U	1	47	1000	0	1000	1079	593	523	14565	0.3	0.1415E-05	0.1450E+01	T	0.4648E+02	.44E+02
CHEBYQAD	B	1	3	10	0	10	48	20	10	123	100.0	0.1865E-07	0.3066E+00	T	0.4773E-02	.55E-01
CHENHARK	B	3	31	1000	300	700	2692	2597	2506	132563	0.5	0.1790E-04	0.2499E-02	T	-.2000E+01	.24E+03
CHNROSNB	U	1	7	10	0	10	26	23	23	284	28.0	0.1838E-10	0.4965E+00	T	0.2722E-21	.39E-01
CLIFF	U	1	8	2	0	2	28	28	28	110	100.0	0.1611E-09	0.1000E-03	T	0.1998E+00	.31E-01
COSINE	U	3	38	1000	0	1000	737	375	341	15648	0.3	0.1226E-02	0.2904E-07	T	-.9806E+03	.50E+02
CRAGGLVY	U	1	6	4	0	4	24	21	21	123	62.5	0.1862E-06	0.1805E-04	T	0.3801E-09	.16E-01
CUBE	U	1	7	2	0	2	48	34	33	129	100.0	0.8199E-10	0.1998E+00	T	0.1514E-19	.31E-01
CVXBQP1	B	1	2	100	100	0	3	3	2	15	6.7	0.0000E+00	no r.Hes.	T	0.2272E+03	.78E-02
DECONVB	B	1	7	61	32	29	136	58	19	339	58.1	0.5386E-08	0.4538E-11	F	0.7690E-02	.34E+00
DECONVU	U	1	52	61	0	61	1111	484	250	11182	58.1	0.3016E-11	0.1300E-13	T	0.6737E-19	.11E+02
DENSCHNA	U	1	3	2	0	2	7	7	7	26	100.0	0.6641E-11	0.7639E+00	T	0.1103E-22	.78E-02
DENSCHNB	U	1	2	2	0	2	13	7	7	25	100.0	0.1089E-10	0.2000E+01	T	0.2170E-22	.78E-02
DENSCHNC	U	1	4	2	0	2	11	11	11	42	100.0	0.7899E-09	0.1875E+01	T	0.2178E-19	.16E-01
DENSCHND	U	1	10	3	0	3	47	37	32	153	100.0	0.1208E-06	0.7154E-06	T	0.2455E-10	.31E-01
DENSCHNE	U	1	3	3	0	3	26	12	10	40	33.3	0.1807E-09	0.2000E+01	T	0.8167E-20	.16E-01
DENSCHNF	U	1	3	2	0	2	7	7	7	26	100.0	0.6289E-09	0.1473E+03	T	0.6514E-21	.78E-02
DIXMAANA	U	1	2	15	0	15	11	9	7	31	32.4	0.2648E-21	0.1875E+01	T	0.1000E+01	.16E-01

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
DIXMAANB	U	1	2	3000	0	3000	16	12	8	143	0.2	0.1511E-16	0.1938E+01	T	0.1000E+01	.16E+01
DIXMAANC	U	1	3	15	0	15	14	11	8	84	32.4	0.7027E-08	0.1875E+01	T	0.1000E+01	.16E-01
DIXMAAND	U	1	2	300	0	300	30	18	10	142	1.7	0.2206E-08	0.1740E+01	T	0.1000E+01	.14E+00
DIXMAANE	U	1	2	15	0	15	23	14	9	108	32.4	0.3855E-07	0.1333E+00	T	0.1000E+01	.16E-01
DIXMAANF	U	1	3	15	0	15	17	13	10	120	32.4	0.5056E-09	0.1333E+00	T	0.1000E+01	.16E-01
DIXMAANG	U	1	3	15	0	15	23	16	12	143	32.4	0.7641E-08	0.1333E+00	T	0.1000E+01	.31E-01
DIXMAANH	U	1	4	15	0	15	26	18	13	164	32.4	0.1706E-08	0.1331E+00	T	0.1000E+01	.23E-01
DIXMAANI	U	1	4	15	0	15	33	18	11	136	32.4	0.7634E-09	0.8889E-02	T	0.1000E+01	.31E-01
DIXMAANJ	U	1	5	15	0	15	35	21	15	201	32.4	0.1400E-10	0.8889E-02	T	0.1000E+01	.31E-01
DIXMAANK	U	1	6	15	0	15	39	19	13	169	32.4	0.2206E-12	0.8889E-02	T	0.1000E+01	.31E-01
DIXMAANL	U	1	4	15	0	15	43	26	17	215	32.4	0.2833E-11	0.8888E-02	T	0.1000E+01	.39E-01
DIXON3DQ	U	1	1	10	0	10	2	2	2	22	26.0	0.8055E-14	0.5455E-01	T	0.6163E-29	.78E-02
DJTL	U	-2	18	2	0	2	2595	2492	2492	9967	100.0	0.2345E-02	0.5435E+06	T	-.8952E+04	.18E+01
DQDRTIC	U	1	1	10	0	10	2	2	2	12	10.0	0.3463E-12	0.2000E+01	T	0.1515E-26	.00E+00
DQRTIC	U	1	10	500	0	500	30	30	30	554	0.2	0.2002E-05	0.5693E-11	T	0.2306E-07	.49E+00
EDENSCH	U	1	5	36	0	36	13	13	13	164	8.2	0.4614E-08	0.2636E+01	T	0.2193E+03	.31E-01
EIGENALS	U	1	6	110	0	110	48	30	21	294	100.0	0.9039E-07	0.8204E-01	T	0.4684E-13	.14E+01
EIGENBLS	U	1	17	110	0	110	237	180	175	5055	100.0	0.7605E-06	0.1444E-01	T	0.4440E-11	.15E+02
EIGENCLS	U	1	22	462	0	462	388	244	174	6706	100.0	0.1898E-05	0.3938E-01	T	0.5397E-11	.62E+03
ENGVAL1	U	1	3	2	0	2	8	8	8	30	100.0	0.1674E-08	0.4000E+01	T	0.0000E+00	.78E-02
ENGVAL2	U	1	5	3	0	3	20	17	15	72	100.0	0.1542E-08	0.3724E+00	T	0.5051E-21	.23E-01
ERRINROS	U	1	12	50	0	50	108	68	63	1098	5.9	0.5662E-07	0.4705E-03	T	0.3990E+02	.26E+00
EXPFIT	U	1	3	2	0	2	14	10	8	28	100.0	0.1121E-07	0.8469E+01	T	0.2405E+00	.16E-01
EXPLIN	B	1	6	120	114	6	45	19	10	91	0.2	0.2724E-06	0.1000E+02	T	-.7235E+06	.31E-01
EXPLIN2	B	-2	6	120	117	3	109	13	7	51	0.2	0.1441E-04	0.3426E+02	T	-.7245E+06	.31E-01
EXPQUAD	B	-2	7	120	7	113	76	24	9	103	2.5	0.2821E-04	0.7748E+01	T	-.3626E+07	.62E-01
EXTROSNB	U	1	11	5	0	5	75	57	57	397	52.0	0.4605E-09	0.5846E-02	T	0.2688E-19	.55E-01
FLETCBV2	U	1	1	10	0	10	3	3	3	34	28.0	0.2400E-08	0.8747E-01	T	-.6154E+00	.78E-02
FLETCBV3	U	1	134	10	0	10	4501	2092	724	7896	28.0	0.2680E-06	0.5663E-08	T	-.2581E-01	.14E+01
FLETCHBV	U	1	60	10	0	10	1612	774	290	3210	28.0	0.1377E-10	0.2717E+01	T	-.2168E+07	.55E+00
FLETCHCR	U	1	5	10	0	10	36	28	28	336	28.0	0.1703E-12	0.4988E+00	T	0.5944E-27	.39E-01
FMINSRF2	U	1	6	64	0	64	68	22	22	610	11.8	0.2405E-07	0.3537E-15	T	0.1000E+01	.20E+00
FMINSURF	U	-2	4	16	0	16	75	12	13	178	100.0	0.7274E-09	-.2755E-15	T	0.1000E+01	.39E-01
FREUROTH	U	-2	5	2	0	2	75	8	9	36	100.0	0.3766E-06	0.8207E+00	T	0.4898E+02	.16E-01
GENROSE	U	1	22	500	0	500	818	417	261	5082	0.6	0.3895E-07	0.2000E+01	T	0.1000E+01	.73E+01
GROWTHLS	U	1	8	3	0	3	183	117	101	490	100.0	0.6175E-08	0.4389E+00	T	0.1004E+01	.11E+00
GULF	U	1	7	3	0	3	48	31	27	129	100.0	0.5557E-08	0.1360E-04	T	0.9293E-17	.12E+00
HADAMALS	B	1	33	1024	824	200	1700	847	39	742	100.0	0.5048E-07	0.1095E+01	T	0.7460E+03	.54E+03
HAIRY	U	1	7	2	0	2	95	52	30	94	100.0	0.4431E-14	0.7607E+03	T	0.2000E+02	.39E-01
HARKERP2	B	1	9	100	99	1	11	10	10	80	100.0	0.1148E-08	0.1000E+01	T	-.5000E+00	.35E+01
HATFLDA	B	1	5	4	0	4	23	21	19	111	62.5	0.2621E-07	0.1049E-01	T	0.6211E-15	.31E-01
HATFLDB	B	1	5	4	1	3	20	18	16	88	62.5	0.1565E-11	0.2545E+00	T	0.5573E-02	.16E-01
HATFLDC	B	1	2	25	0	25	6	6	6	85	11.4	0.8111E-08	0.1333E+01	T	0.7113E-17	.78E-02
HATFLDD	U	1	4	3	0	3	15	12	9	40	100.0	0.1472E-06	0.1046E-02	T	0.6615E-07	.16E-01

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
HATFLDE	U	1	5	3	0	3	18	15	13	61	100.0	0.4857E-07	0.5875E-02	T	0.5120E-06	.16E-01
HEART6LS	U	-2	112	6	0	6	2149	1231	616	4326	100.0	0.3826E-05	0.3902E-03	T	0.2972E-18	.82E+00
HEART8LS	U	1	26	8	0	8	587	330	162	1472	100.0	0.7532E-09	0.4597E+00	T	0.2805E-21	.26E+00
HELIX	U	1	6	3	0	3	22	18	16	76	100.0	0.4380E-09	0.1433E+01	T	0.3482E-20	.23E-01
HILBERTA	U	1	1	2	0	2	2	2	2	6	100.0	0.1480E-15	0.6574E-01	T	0.1315E-30	.00E+00
HILBERTB	U	1	1	5	0	5	2	2	2	12	100.0	0.6883E-14	0.1000E+02	T	0.2156E-29	.16E-01
HIMMELBB	U	1	4	2	0	2	20	14	8	24	100.0	0.6211E-07	0.3147E-09	T	0.1653E-18	.23E-01
HIMMELBF	U	-3	500	4	0	4	28404	18761	9381	46909	100.0	0.5046E-03	-.2276E-03	T	0.3186E+03	.11E+02
HIMMELBG	U	1	2	2	0	2	10	7	6	21	100.0	0.9297E-11	0.4000E+01	T	0.8694E-23	.78E-02
HIMMELBH	U	1	1	2	0	2	9	3	3	8	50.0	0.7994E-14	0.2000E+01	T	-.1000E+01	.78E-02
HIMMELP1	B	1	3	2	1	1	30	13	8	19	100.0	0.3047E-11	0.1035E+00	T	-.2390E+02	.78E-02
HS1	B	1	8	2	0	2	43	31	31	122	100.0	0.2424E-09	0.3994E+00	T	0.1123E-21	.16E-01
HS110	B	1	1	50	50	0	8	2	1	6	100.0	0.0000E+00	no r.Hes.	T	-.9990E+10	.78E-02
HS2	B	1	2	2	1	1	17	9	5	15	100.0	0.3443E-08	0.1191E+04	T	0.4941E+01	.78E-02
HS25	B	1	8	3	0	3	104	49	31	134	100.0	0.2873E-08	0.1360E-04	T	0.1819E-15	.16E+00
HS3	B	1	2	2	1	1	7	3	3	7	100.0	0.1421E-18	0.2000E-04	T	0.5049E-33	.00E+00
HS38	B	1	7	4	0	4	110	73	54	303	62.5	0.6974E-09	0.7196E+00	T	0.3209E-21	.47E-01
HS3MOD	B	1	3	2	1	1	19	6	4	10	100.0	0.0000E+00	0.2000E+01	T	0.0000E+00	.16E-01
HS4	B	1	2	2	2	0	6	3	2	6	25.0	0.0000E+00	no r.Hes.	T	0.2667E+01	.78E-02
HS45	B	1	2	5	5	0	8	3	2	2	100.0	0.0000E+00	no r.Hes.	T	0.1000E+01	.00E+00
HS5	B	1	3	2	0	2	18	10	5	15	100.0	0.9546E-13	0.1732E+01	T	-.1913E+01	.78E-02
HYDC20LS	U	-3	500	99	0	99	96770	96753	96747	3080619	21.9	0.5640E-01	0.1096E+03	T	0.1430E+00	.20E+04
INDEF	U	3	190	1000	0	1000	25486	2014	857	17155	0.5	0.3408E+02	-.2065E+02	T	-.8566E+10	.14E+03
JENSMP	U	-2	6	2	0	2	77	11	12	48	100.0	0.4802E-05	0.4484E+04	T	0.1244E+03	.78E-02
JIMACK	U	3	9	3549	0	3549	692	692	691	48103	1.9	0.2410E-04	0.2515E-01	T	0.8668E+00	.74E+04
JNLBRNG1	B	1	7	5625	2031	3594	172	92	12	397	0.1	0.1433E-05	0.1441E-01	T	-.1805E+00	.14E+02
JNLBRNG2	B	1	8	5625	2429	3196	70	44	18	782	0.1	0.1557E-05	0.1987E-01	T	-.4147E+01	.13E+02
JNLBRNGA	B	1	9	15625	5657	9968	120	70	20	1005	< 0.1	0.2628E-05	0.5894E-02	T	-.2685E+00	.56E+02
KOWOSB	U	1	3	4	0	4	19	13	11	66	100.0	0.1104E-10	0.2892E-02	T	0.3078E-03	.16E-01
LIARWHD	U	1	4	36	0	36	11	11	11	43	8.2	0.8480E-08	0.2802E+01	T	0.6772E-18	.16E-01
LINVERSE	B	1	8	19	7	12	70	34	9	112	38.0	0.1071E-06	0.8362E-13	F	0.6000E+01	.47E-01
LOGHAIRY	U	1	489	2	0	2	9202	4692	2681	8569	100.0	0.2344E-08	0.6339E+01	T	0.1823E+00	.25E+01
LOGROS	B	1	5	2	0	2	100	63	61	242	100.0	0.1059E-06	0.4000E+00	T	0.0000E+00	.55E-01
MANCINO	U	-2	4	100	0	100	11	8	7	83	100.0	0.3889E-05	0.3595E+07	T	0.1878E-17	.13E+02
MARATOSB	U	-2	31	2	0	2	1413	902	902	3607	100.0	0.2911E-05	0.1000E+01	T	-.1000E+01	.64E+00
MAXLIKA	B	1	12	8	1	7	147	80	37	285	100.0	0.8687E-12	0.1622E-01	T	0.1136E+04	.10E+01
MCCORMCK	B	1	2	10	1	9	11	6	4	42	28.0	0.9601E-07	0.2565E+01	T	-.9598E+01	.78E-02
MDHOLE	B	1	3	2	1	1	93	55	48	185	100.0	0.1314E-09	0.2000E+03	T	0.4316E-22	.23E-01
MEXHAT	U	1	7	2	0	2	33	29	29	114	100.0	0.1107E-07	0.3763E+04	T	-.4001E-01	.31E-01
MEYER3	U	-2	15	3	0	3	368	188	183	913	100.0	0.4736E-01	0.3329E-01	T	0.8795E+02	.20E+00
MOREBV	U	1	1	10	0	10	3	3	3	34	44.0	0.2041E-06	0.3821E-01	T	0.6582E-13	.78E-02
MSQRTALS	U	1	3	4	0	4	22	13	9	49	100.0	0.1020E-06	0.4972E-02	T	0.6564E-14	.78E-02
MSQRTBLS	U	1	3	9	0	9	18	12	8	84	100.0	0.1576E-07	0.4677E+00	T	0.9477E-16	.23E-01
NCB20	U	1	10	1010	0	1010	322	99	43	1140	3.8	0.2765E-05	0.1908E-13	T	0.9060E+03	.28E+02

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
NCB20B	U	-2	2	21	0	21	52	3	4	19	99.5	0.9908E-07	0.2887E-01	T	0.4200E+02	.78E-02
NCVXBQP1	B	1	2	100	100	0	21	8	1	11	6.7	0.0000E+00	no r.Hes.	T	-.1996E+07	.23E-01
NCVXBQP2	B	1	2	100	100	0	16	6	1	11	6.7	0.0000E+00	no r.Hes.	T	-.1333E+07	.78E-02
NCVXBQP3	B	1	3	100	99	1	35	16	4	26	6.7	0.9095E-12	0.6000E+02	T	-.6633E+06	.31E-01
NOBNDTOR	B	1	1	100	52	48	11	7	3	54	3.8	0.5651E-08	0.3059E+00	T	-.5521E+00	.31E-01
NONCVXU2	U	3	46	1000	0	1000	2066	982	645	29709	0.7	0.4149E-03	0.7180E-02	T	0.2318E+04	.13E+03
NONCVXUN	U	-3	500	1000	0	1000	96165	95311	95061	5318575	0.7	0.1165E-02	0.7502E-02	T	0.2325E+04	.16E+05
NONDIA	U	-2	4	1000	0	1000	9	7	8	42	0.3	0.4213E-10	-.1820E-12	T	0.1787E-25	.22E+00
NONDQUAR	U	1	8	100	0	100	139	83	83	1972	4.9	0.6497E-06	0.3304E-06	T	0.8848E-07	.72E+00
NONMSQRT	U	2	25	9	0	9	326	64	69	733	33.3	0.2573E-04	0.3211E-07	T	0.7520E+00	.10E+00
NONSCOMP	B	1	3	25	0	25	9	9	9	138	11.7	0.2333E-06	0.4347E-10	T	0.3868E-13	.16E-01
OBSTCLAE	B	1	7	5625	2723	2902	76	45	14	545	0.1	0.2738E-06	0.2729E-01	T	0.1863E+01	.10E+02
OBSTCLBL	B	1	1	100	84	16	8	5	2	10	3.8	0.6245E-16	0.1719E+01	T	0.2875E+01	.78E-02
OBSTCLBM	B	1	6	15625	4308	11317	77	42	8	294	< 0.1	0.3828E-05	0.1102E-01	T	0.7296E+01	.27E+02
OBSTCLBU	B	1	1	100	84	16	6	4	2	10	3.8	0.8925E-16	0.1719E+01	T	0.2875E+01	.78E-02
OSBORNEA	U	1	7	5	0	5	35	27	24	168	100.0	0.1467E-08	0.3887E-04	T	0.5465E-04	.39E-01
OSBORNEB	U	1	6	11	0	11	33	19	15	187	100.0	0.6161E-09	0.6488E-02	T	0.4014E-01	.94E-01
OSLBQP	B	1	1	8	7	1	2	2	2	4	12.5	0.1241E-15	0.1000E+01	F	0.6250E+01	.78E-02
PALMER1	B	-2	4	4	0	4	59	12	11	64	100.0	0.2963E-09	-.8907E-10	T	0.1175E+05	.23E-01
PALMER1A	B	1	6	6	0	6	65	43	40	321	100.0	0.3608E-10	0.1506E-02	T	0.8988E-01	.55E-01
PALMER1B	B	1	4	4	0	4	33	19	17	102	100.0	0.2902E-07	0.2604E+00	T	0.3447E+01	.23E-01
PALMER1C	U	1	2	8	0	8	5	5	5	56	100.0	0.5132E-07	0.3043E-03	T	0.9760E-01	.78E-02
PALMER1D	U	1	2	7	0	7	4	4	4	41	100.0	0.1632E-07	0.2202E-02	T	0.6527E+00	.16E-01
PALMER1E	B	-2	4	8	1	7	73	8	9	78	100.0	0.1222E-08	-.4648E-11	F	0.3110E+01	.23E-01
PALMER2	B	1	3	4	0	4	24	12	9	53	100.0	0.1326E-11	0.4638E-10	T	0.3651E+04	.16E-01
PALMER2A	B	1	14	6	0	6	151	100	91	723	100.0	0.5287E-08	0.6756E-04	T	0.1711E-01	.12E+00
PALMER2B	B	-2	5	4	0	4	65	18	17	105	100.0	0.5331E-06	0.7391E+00	T	0.6233E+00	.23E-01
PALMER2C	U	1	2	8	0	8	4	4	4	46	100.0	0.3030E-07	0.3192E-04	T	0.1437E-01	.78E-02
PALMER2E	B	3	33	8	0	8	773	375	187	1691	100.0	0.9191E-04	-.6555E-06	T	0.1163E+00	.40E+00
PALMER3	B	-1	2	4	2	2	26	4	2	14	100.0	0.1787E+07	-.1735E+02	F	0.3004E+04	.78E-02
PALMER3A	B	1	16	6	0	6	163	100	94	750	100.0	0.1983E-10	0.6632E-04	T	0.2043E-01	.12E+00
PALMER3B	B	1	4	4	0	4	22	13	11	66	100.0	0.1029E-10	0.2563E+01	T	0.4228E+01	.23E-01
PALMER3C	U	1	2	8	0	8	4	4	4	46	100.0	0.5285E-08	0.2989E-04	T	0.1954E-01	.78E-02
PALMER3E	B	1	4	8	0	8	52	32	18	173	100.0	0.4163E-10	0.1723E-03	T	0.5074E-04	.39E-01
PALMER4	B	1	3	4	0	4	49	15	8	41	100.0	0.2868E-07	0.9167E-09	T	0.2285E+04	.23E-01
PALMER4A	B	1	7	6	0	6	85	56	53	425	100.0	0.1752E-06	0.3501E-03	T	0.4061E-01	.70E-01
PALMER4B	B	1	3	4	0	4	20	13	11	66	100.0	0.1745E-09	0.3161E+01	T	0.6835E+01	.23E-01
PALMER4C	U	1	1	8	0	8	6	5	4	45	100.0	0.1397E-07	0.3064E-04	T	0.5031E-01	.78E-02
PALMER4E	B	1	36	8	0	8	962	458	235	2136	100.0	0.1343E-08	0.2813E-03	T	0.1480E-03	.48E+00
PALMER5A	B	3	52	8	0	8	1253	543	479	4694	100.0	0.1259E-03	0.9177E-07	T	0.9503E-01	.65E+00
PALMER5B	B	1	10	9	0	9	102	59	48	524	100.0	0.1767E-09	0.3366E-05	T	0.9752E-02	.78E-01
PALMER5C	U	1	1	6	0	6	2	2	2	14	100.0	0.4988E-12	0.3191E+01	T	0.2128E+01	.00E+00
PALMER5D	B	-2	2	8	0	8	6	3	4	34	25.0	0.1008E-11	-.1274E-13	T	0.8734E+02	.78E-02
PALMER5E	B	3	48	8	0	8	1359	717	726	7245	100.0	0.7603E-04	0.2633E-08	T	0.2451E-01	.98E+00

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
PALMER6A	B	-1	2	6	2	4	30	5	2	20	100.0	0.6111E+05	-.8110E+00	F	0.3143E+01	.78E-02
PALMER6C	U	-2	1	8	0	8	4	3	2	26	100.0	0.9329E+05	0.2536E-05	T	0.6783E+05	.16E-01
PALMER6E	B	1	5	8	0	8	59	36	29	290	100.0	0.9790E-10	0.6348E-04	T	0.2240E-03	.55E-01
PALMER7A	B	3	41	6	0	6	1670	917	923	7365	100.0	0.9550E-03	0.2186E-08	T	0.1039E+02	.97E+00
PALMER7C	U	1	2	8	0	8	4	4	4	46	100.0	0.1468E-07	0.6696E-05	T	0.6020E+00	.78E-02
PALMER7E	B	3	43	8	0	8	728	364	181	1637	100.0	0.6510E-03	-.1423E-05	T	0.1015E+02	.32E+00
PALMER8A	B	1	6	6	0	6	62	36	30	238	100.0	0.1568E-06	0.1477E-02	T	0.7401E-01	.39E-01
PALMER8C	U	1	2	8	0	8	4	4	4	46	100.0	0.1172E-08	0.6071E-05	T	0.1598E+00	.78E-02
PALMER8E	B	1	7	8	0	8	224	110	66	622	100.0	0.6759E-10	0.1755E-03	T	0.6339E-02	.11E+00
PENALTY1	U	1	13	1000	0	1000	44	42	42	210	100.0	0.2138E-06	0.1265E-02	T	0.9686E-02	.34E+03
PENALTY2	U	1	17	4	0	4	223	145	145	872	100.0	0.4393E-08	0.3073E-05	T	0.9376E-05	.13E+00
PENALTY3	U	1	11	50	0	50	343	211	206	2493	100.0	0.1012E-06	0.4015E-01	T	0.2418E+04	.12E+02
PENTDI	B	1	1	1000	998	2	3	2	1	1	0.5	0.9741E-09	0.1575E+02	F	-.7500E+00	.39E-01
PFIT1LS	U	-2	13	3	0	3	257	157	151	753	100.0	0.2491E-06	0.8217E-04	T	0.5401E-17	.12E+00
PFIT2LS	U	1	14	3	0	3	326	198	193	962	100.0	0.2128E-10	0.1288E-02	T	0.1093E-23	.16E+00
PFIT3LS	U	1	18	3	0	3	535	339	334	1667	100.0	0.9115E-08	0.3225E-02	T	0.1678E-18	.27E+00
PFIT4LS	U	-2	22	3	0	3	690	438	433	2164	100.0	0.4218E-06	0.4090E-02	T	0.5873E-17	.35E+00
POWELLSG	U	1	6	4	0	4	19	19	19	112	75.0	0.1402E-06	0.7975E-05	T	0.3375E-10	.23E-01
POWER	U	1	7	10	0	10	21	21	21	260	100.0	0.1174E-06	0.2011E-04	T	0.2474E-10	.23E-01
PROBPENL	B	3	23	500	4	496	1951	98	60	1036	100.0	0.9341E-03	-.9394E-06	T	-.5298E+04	.98E+02
PSPDOC	B	1	1	4	1	3	7	4	4	18	87.5	0.5031E-07	0.1726E+00	T	0.2414E+01	.00E+00
QR3DLS	B	1	12	155	0	155	522	493	484	17105	75.3	0.1217E-05	0.1898E-02	T	0.9080E-10	.44E+02
QRTQUAD	B	-2	7	120	5	115	143	61	24	268	2.5	0.1760E-05	0.7745E+01	T	-.3625E+07	.13E+00
QUARTC	U	1	9	25	0	25	26	25	25	332	4.0	0.1994E-07	0.9429E-22	T	0.1817E-10	.47E-01
QUDLIN	B	1	1	12	12	0	3	2	0	0	0.7	0.0000E+00	no r.Hes.	F	-.7200E+04	.00E+00
ROSENBR	U	-2	7	2	0	2	33	25	26	104	100.0	0.1454E-06	0.3994E+00	T	0.1127E-16	.23E-01
S308	U	1	4	2	0	2	11	10	10	38	100.0	0.7261E-11	0.2274E+01	T	0.7732E+00	.78E-02
S368	B	1	4	8	3	5	10	8	6	45	100.0	0.6272E-09	0.2375E+01	T	-.9375E+00	.16E-01
SCHMVETT	U	1	2	3	0	3	4	4	4	18	100.0	0.1470E-12	0.7761E+00	T	-.3000E+01	.78E-02
SENSORS	U	-2	4	100	0	100	121	31	14	167	100.0	0.2657E-05	0.9900E+02	T	-.2106E+04	.95E+01
SIM2BQP	B	1	1	2	2	0	3	2	0	0	25.0	0.0000E+00	no r.Hes.	T	0.0000E+00	.00E+00
SIMBQP	B	1	1	2	1	1	6	4	2	4	100.0	0.0000E+00	0.1000E+02	T	0.0000E+00	.78E-02
SINEALI	B	1	20	10	0	10	726	466	458	5489	28.0	0.9715E-07	0.1104E-01	T	-.9010E+03	.58E+00
SINEVAL	U	1	4	2	0	2	88	57	56	221	100.0	0.2533E-07	0.2500E+00	T	0.1301E-18	.39E-01
SINQUAD	U	1	5	5	0	5	15	15	15	74	76.0	0.1172E-06	0.2281E-04	T	0.9034E-10	.23E-01
SISSER	U	1	6	2	0	2	17	17	17	66	100.0	0.4251E-07	0.1006E-04	T	0.1623E-10	.23E-01
SNAIL	U	1	5	2	0	2	135	86	84	331	100.0	0.7069E-10	0.2000E+01	T	0.1249E-20	.62E-01
SPARSINE	U	1	15	1000	0	1000	1446	1395	1380	76830	3.0	0.3155E-05	0.1190E+01	T	0.2947E-11	.55E+03
SPARSQUR	U	1	7	1000	0	1000	21	21	21	484	3.0	0.1069E-05	0.2995E-04	T	0.1151E-08	.33E+01
SPECAN	B	1	4	9	0	9	21	12	8	94	33.3	0.1899E-10	0.1949E+03	T	0.1646E-12	.28E+01
SPMSRTLS	U	1	4	28	0	28	26	16	10	146	25.5	0.2929E-06	0.1875E+00	T	0.4110E-13	.39E-01
SROSENBR	U	1	4	10	0	10	10	9	9	37	20.0	0.2085E-13	0.3994E+00	T	0.1087E-27	.23E-01
TESTQUAD	U	1	18	1000	0	1000	250	250	250	13799	0.1	0.2751E-05	0.3029E+01	T	0.1034E-14	.27E+02
TOINTGOR	U	1	5	50	0	50	10	10	10	206	11.2	0.3108E-06	0.2640E+00	T	0.1374E+04	.62E-01

name	c	e	it	n	nb	nf	#f	#gf	#hf	#hfv	%den	norm pgf	lam min	s	obj.fun	time
TOINTGSS	U	1	1	10	0	10	2	2	2	22	44.0	0.5304E-13	0.1401E+01	T	0.1000E+02	.78E-02
TOINTPSP	U	1	3	50	0	50	20	14	14	197	11.2	0.1643E-07	0.1006E+01	T	0.2256E+03	.47E-01
TOINTQOR	U	1	2	50	0	50	5	5	5	89	11.2	0.1052E-06	0.1351E+01	T	0.1175E+04	.31E-01
TORSION1	B	1	1	100	68	32	8	5	2	25	3.8	0.3643E-15	0.6921E+00	T	-.4923E+00	.16E-01
TORSION2	B	1	2	100	68	32	9	6	3	37	3.8	0.3643E-15	0.6921E+00	T	-.4923E+00	.16E-01
TORSION3	B	1	1	100	88	12	8	5	2	7	3.8	0.9205E-16	0.2000E+01	T	-.1271E+01	.78E-02
TORSION4	B	1	2	100	88	12	9	6	3	19	3.8	0.9205E-16	0.2000E+01	T	-.1271E+01	.16E-01
TORSION5	B	1	1	100	100	0	1	1	0	0	< 0.1	0.0000E+00	no r.Hes.	T	-.2897E+01	.00E+00
TORSION6	B	1	6	14884	12316	2568	50	29	8	275	< 0.1	0.1275E-06	0.4680E-01	T	-.2859E+01	.15E+02
TORSIONA	B	1	1	16	12	4	5	3	1	1	25.0	0.3468E-09	0.5910E+01	T	-.3086E+00	.16E-01
TORSIONB	B	1	1	16	12	4	2	2	2	6	25.0	0.0000E+00	0.2000E+01	T	-.3086E+00	.78E-02
TORSIONC	B	1	1	16	16	0	1	1	0	0	0.4	0.0000E+00	no r.Hes.	T	-.1037E+01	.00E+00
TORSIOND	B	1	1	16	16	0	2	2	1	3	25.0	0.0000E+00	no r.Hes.	T	-.1037E+01	.00E+00
TORSIONE	B	1	1	16	16	0	1	1	0	0	0.4	0.0000E+00	no r.Hes.	T	-.2519E+01	.00E+00
TORSIONF	B	1	1	16	16	0	2	2	1	3	25.0	0.0000E+00	no r.Hes.	T	-.2519E+01	.78E-02
TQUARTIC	U	1	1	5	0	5	2	2	2	7	52.0	0.4441E-15	0.3845E+00	T	0.4930E-31	.00E+00
TRIDIA	U	1	4	30	0	30	9	9	9	156	9.8	0.2531E-06	0.1438E+01	T	0.4778E-14	.31E-01
VARDIM	U	1	5	10	0	10	15	15	15	62	100.0	0.5194E-11	0.2000E+01	T	0.1747E-25	.23E-01
VAREIGVL	U	1	8	50	0	50	39	25	23	632	100.0	0.1866E-10	0.1084E-14	T	0.7310E-20	.39E+00
VIBRBEAM	U	-2	5	8	0	8	53	24	13	131	100.0	0.2189E-02	0.1839E+00	T	0.3315E+00	.70E-01
WATSON	U	3	5	12	0	12	71	33	24	274	100.0	0.2919E-04	0.7505E-08	T	0.1993E-05	.11E+00
WEEDS	B	1	7	3	0	3	40	24	18	87	100.0	0.2097E-10	0.4414E-02	T	0.2587E+01	.16E-01
WOODS	U	1	11	1000	0	1000	437	292	163	855	0.2	0.3068E-09	0.7196E+00	T	0.3240E-21	.35E+01
YFIT	B	1	8	3	0	3	58	43	42	207	100.0	0.1391E-06	0.9472E-04	T	0.6670E-12	.47E-01
YFITU	U	1	8	3	0	3	58	43	42	207	100.0	0.1391E-06	0.9472E-04	T	0.6670E-12	.55E-01
ZANGWIL2	U	1	1	2	0	2	2	2	2	5	100.0	0.0000E+00	0.1600E+01	T	-.1820E+02	.16E-01

Finally we give a comparison of the results obtained for the problems with dimension greater then or equal to 1000 by our algorithm and the limited memory quasi-Newton method LBFGS-B 2.1. of Nocedal et.al. [BLNZ95]. We used the same termination criteria as given in subsection 4.4 and the maximum subspace dimension in LBFGS-B 2.1. was set to $m = 17$. The given data are

Name	Description
`name`	Name of the problem.
`c`	Classification of the problem.
`n`	Number of variables. Dimension of x.
`n.pgf.pl2`	Norm of the projected gradient of the objective $\|x^e - P_\Omega(x^e - \nabla f(x^e))\|$ in the final point of our method.
`o.fun.pl2`	Objective function value $f(x^e)$ in the final point of our method.
`t.pl2`	CPU-time in seconds needed by our method.
`n.pgf.lmqp`	Norm of the projected gradient of the objective $\|x^e - P_\Omega(x^e - \nabla f(x^e))\|$ in the final point of LBFGS-B.
`o.fun.lmqn`	Objective function value $f(x^e)$ in the final point of LBFGS-B.
`t.lmqn`	CPU-time in seconds needed by LBFGS-B.

name	c	n	n.pgf.pl2	o.fun.pl2	t.pl2	n.pgf.lmqn	o.fun.lmqn	t.lmqn
BIGGSB1	B	1000	0.315305E-05	0.150000E-01	.66E+02	0.194222E+00	0.198497E+00	.51E-01
BQPGAUSS	B	2003	0.329910E-04	-.362578E+00	.13E+03	0.253196E+01	0.499919E+02	.51E-01
CHAINWOO	U	1000	0.141524E-05	0.464850E+02	.44E+02	0.237257E-02	0.330127E+03	.11E+03
CHENHARK	B	1000	0.178992E-04	-.200000E+01	.24E+03	0.492647E-04	-.200000E+01	.63E+02
COSINE	U	1000	0.122560E-02	-.980615E+03	.50E+02	0.853830E-07	-.999000E+03	.37E+00
DIXMAANB	U	3000	0.151113E-16	0.100000E+01	.16E+01	0.153401E-05	0.100000E+01	.21E+00
HADAMALS	B	1024	0.504818E-07	0.746002E+03	.54E+03	0.423994E+02	0.758905E+04	.19E+02
INDEF	U	1000	0.340776E+02	-.856584E+10	.14E+03	0.389085E+02	-.594184E+23	.23E+00
JIMACK	U	3549	0.240988E-04	0.866793E+00	.74E+04	0.108445E-03	0.866793E+00	.84E+04
JNLBRNG1	B	5625	0.143321E-05	-.180548E+00	.14E+02	0.203277E+01	0.882780E+01	.59E+00
JNLBRNG2	B	5625	0.155722E-05	-.414656E+01	.13E+02	0.296635E+01	0.741889E+01	.26E+00
JNLBRNGA	B	15625	0.262766E-05	-.268510E+00	.56E+02	0.921316E-01	-.249031E+00	.12E+02
NCB20	U	1010	0.276494E-05	0.905997E+03	.28E+02	0.113636E-02	0.950647E+03	.22E+02
NONCVXU2	U	1000	0.414898E-03	0.231788E+04	.13E+03	0.251954E-02	0.231869E+04	.32E+02
NONCVXUN	U	1000	0.116454E-02	0.232465E+04	.16E+05	0.212479E-02	0.231933E+04	.30E+03
NONDIA	U	1000	0.421261E-10	0.178727E-25	.22E+00	0.345933E-06	0.156117E-17	.31E+00
OBSTCLAE	B	5625	0.273845E-06	0.186300E+01	.10E+02	0.140328E+00	0.219372E+01	.12E+02
OBSTCLBM	B	15625	0.382794E-05	0.729576E+01	.27E+02	0.395521E-01	0.730669E+01	.45E+02
PENALTY1	U	1000	0.213778E-06	0.968618E-02	.34E+03	0.115926E-05	0.968618E-02	.17E+01
PENTDI	B	1000	0.974128E-09	-.750000E+00	.39E-01	0.000000E+00	-.750000E+00	.12E-01
SPARSINE	U	1000	0.315522E-05	0.294653E-11	.55E+03	0.105433E-02	0.107623E-06	.10E+03
SPARSQUR	U	1000	0.106900E-05	0.115132E-08	.33E+01	0.222204E-05	0.289818E-08	.73E+00
TESTQUAD	U	1000	0.275117E-05	0.103380E-14	.27E+02	0.108011E-02	0.122498E-08	.59E+02
TORSION6	B	14884	0.127499E-06	-.285880E+01	.15E+02	0.100470E-01	-.285799E+01	.16E+03
WOODS	U	1000	0.306765E-09	0.323957E-21	.35E+01	0.313370E-05	0.528041E-14	.82E+00

6 Conclusion

On balance the result of the numerical experiments is positive. The combination of projected gradient steps and a more sophisticated unconstrained solver, as proposed by Moré and Toraldo, also works effective if second order directions are used in the unconstrained phase. The difficulties in solving for example `HYDC20LS` or `PALMERxy` are either scaling problems or bounds to close to singularities. Most of these programming problems can be solved using a scaling of the variables or more conformed bounds. Note that the algorithm in the implemented version does not make use of any scaling.

An advantage of the implementation is the usage of (SYMMLQ) if the reduced Hessian is positive definite. In the case of a strictly convex programming problem the algorithm does not need the computation of any ritzvectors. That is wy we achieve the good results in solving `OBSTCLxy` and `TORSIONx`.

Of a special interest is the comparison between our algorithm and the limited memory quasi-Newton method LBFGS-B 2.1. of Nocedal et.al. This algorithm (see [BLNZ95]) makes use of limited memory BFGS matrices as presented in [BNS94] and was especially designed for large scale problems. Moreover a gradient projection method, similar to our approach, is implemented to handle the bound constraints. In case of small problems the limited memory quasi-Newton method was faster. But the given results show that the use of second order information makes our approach much more reliable in case of large scale problems. Even by solving problems with a convex objective function, as `JNLBRNGx`, `OBSTCLxy`, and `TORSION6` LBFGS-B 2.1. failed.

The performance of (BCA) depends strongly on the approximation of the eigenvalues and the truncated Newton direction (76). Therefore the choice of the initial vector for the Lanczos iteration and the control of the number of Lanczos steps is very important. Concerning the convergence analysis the correctness of the eigenvalue approximation can only be assumed. Nevertheless, the numerical experiments show, that the implementation overcomes these difficulties and is very efficient for large scale problems.

References

[BCGT97] I. Bongartz, A.R. Conn, N. Gould, and Ph.L. Toint. *CUTE: Constrained and unconstrained testing environment.* http://www.dci.clrc.-ac.uk/Activity.asp?CUTE, 1997.

[BLNZ95] R.H. Byrd, P. Lu, J. Nocedale, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Computation*, 16(5):1190–1208, 1995.

[BNS94] R.H. Byrd, J. Nocedal, and R.B. Schnabel. Representation of quasinewton matrices and their use in limited memory methods. *Math. Programming*, 63(4):129–156, 1994.

[CM87] P.H. Calamai and J.J. Moré. Projected gradient methods for linearly constrained problems. *Math. Programming*, 39:93–116, 1987.

[EW96] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact newton method. *SIAM J. Comput.*, 17(1):16–32, 1996.

[Fel96] R. Felkel. An interior-point method for large-scale qp problems. Technical Report 1850, TH Darmstadt, 1996.

[GL89] G.H. Golub and C.F. van Loan. *Matrix Computations.* John Hopkins University Press, 2 edition, 1989.

[Hei85] Norbert Heinrich. *Eine neue Modifikation des Newtonverfahrens für nichtrestringierte und linear restringierte Optimierungsprobleme mit Mehrfachinaktivierung im linear restringierten Fall.* PhD thesis, TH Darmstadt, 1985.

[MT91] J.J. Moré and G. Toraldo. On the resolution of large quadratic programming problems with bound constraints. *SIAM J. Optimization*, 1:93–113, 1991.

[OR70] J.M. Orthega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Academic press, 1970.

[Par80a] B.N. Parlett. A new look at the lanczos algorithm for solving symmetric systems of linear equations. *Linear algebra and its applications*, 29:323–346, 1980.

[Par80b] B.N. Parlett. *The symmetric eigenvalue problem.* Prectice-Hall Int., 1980.

[PS75] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Analysis*, 12(4):617–629, 1975.

[PS79] B.N. Parlett and D.S. Scott. The lanczos algorithm with selective orthogonalization. *Math. of. Computation*, 33(145):217–238, 1979.

[Saa96] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, 1996.

[Sim84a] H.D. Simon. Analysis of the symmetric lanczos algorithm with reorthogonalization methods. *Linear Algebra and its Applications*, 61:101–131, 1984.

[Sim84b] H.D. Simon. The lanczos algorithm with partial reorthogonalization. *Math. of Computation*, 42:115–142, 1984.

[Spe93] P. Spellucci. *Numerische Verfahren der nichtlinearen Optimierung.* Birkhäuser, 1993.

[ST85] P. Spellucci and W. Törnig. *Eigenwertberechnungen in den Ingenieurwissenschaften.* Teubner, 1985.