EXPÉDITEUR

M. I. PEREZ
Département de MATHÉMATIQUES - Lab. d'Analyse Numérique
Université de DIJON
214, rue de Mirande

21000 DIJON
FRANCE

CARTE POSTALE

Comp Sci

DESTINATAIRE

Doctor
A.R. Conn
~~Dept. of Combinatorics and~~ Opt.
University of Waterloo
Waterloo, Ontario
CANADA N2L 3G1

Dijon, le 3 sept. 85

Monsieur,
Dear Sir,

Je vous serais reconnaissant de bien vouloir m'envoyer un tiré à part de votre article intitulé :
I would greatly appreciate receiving a reprint of your paper entitled:

An approach to non-linear l_1 data fitting

81-17

paru dans :
which was published in :

Computer Sc. Dpt. CS-81-17 May (81)

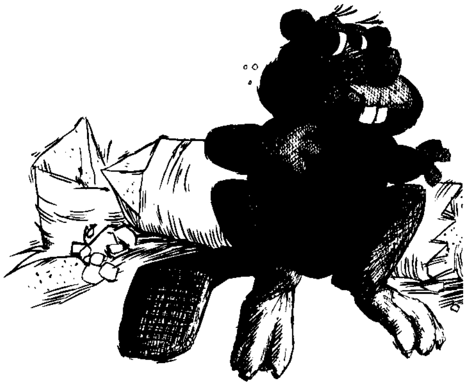
Oct 7/85
Report mailed
this date
Don

Avec mes remerciements
With many thanks

Ignacio D. V.

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



An Approach to Nonlinear l_1 Data Fitting

Richard H. Bartels
Andrew R. Conn

CS-81-17

May, 1981

An Approach to Nonlinear l_1 Data Fitting

Richard H. Bartels

The University of Waterloo
Computer Science Department

Andrew R. Conn

The University of Waterloo
Computer Science Department

ABSTRACT

The traditional method of data fitting is by the least squares (l_2) technique. When the data is good -- reasonably accurate with normally distributed errors -- this method is ideal. When the data is bad -- contaminated by occasional wild values -- then the l_1 technique (minimizing sums of absolute values of residuals) has much to recommend it. This paper surveys the strategy of a globally and superlinearly convergent algorithm to minimize sums of absolute values of C^2 functions. The approach to be presented is closely related to the use of a certain, piecewise differentiable penalty function to solve nonlinear programming problems.

1. Introduction

In this paper we outline the strategy of a method for solving the problem

$$\underset{x}{\text{minimize}}\ F(x) = \sum_{i=1}^{m} |f_i(x)| \tag{1.1}$$

where $x \in \mathbf{R}^n$ and $f_i \in C^2$ for all $i \in \{1, \ldots, m\}$. Such problems arise, for example, in data fitting contexts, where the functions f_i are defined by

$$f_i(x) = h(t_i;x) - y_i \ .$$

Here the y_i are considered to be observations of the functional $h(t,x)$ at t_i. If h is linear in x; that is, $h(t_i;x) = h_i^T x$, then (1.1) is equivalent to a specialized linear programming problem

$$\underset{u,v,x}{\text{minimize}}\ \sum_{i=1}^{m} (u_i + v_i) \tag{1.2}$$

$$\text{subject to } h_i^T x + u_i - v_i = y_i\ ;\ i = 1, \ldots, m$$

$$\text{and} \qquad u_i \geqslant 0,\ v_i \geqslant 0,\ ; \qquad i = 1, \ldots, m$$

but with x unrestricted.

The conventional method of fitting data y to a functional form h would replace the absolute value in (1.1) by the square, giving an l_2 estimation of x. Such estimates are very sensitive to the presence of outliers -- occasional observations y_i which are wildly out of line with the rest. The l_1 estimate of x has a certain capacity to ignore outliers. A simple linear example will suffice to illustrate.

Let

$$h(t;x) = x_1 + tx_2 \ ,$$

and consider the data

t	y
1	0.75
2	2.00
3	3.00
4	4.25
5	4.75
6	6.50
7	7.25
8	0.00

which was obtained, very roughly, from the form h with $x_1 = 0$, $x_2 = 1$. Note that the y value corresponding to $t = 8$ is wild. (Perhaps an 8 in the first position was misread as a zero when the data was collected or transcribed.) The l_1 estimation of x for this data is $\bar{x}_1 = -0.1875$, $\bar{x}_2 = +1.0625$, which is not unreasonable, considering the accuracy of the good portion of the data. The l_2 estimation of x for this data is $\hat{x}_1 = +1.848$, $\hat{x}_2 = +0.381$. A plot of the data and the two lines

$$\bar{y} = \bar{x}_1 + t\bar{x}_2 \qquad (l_1\text{line})$$
$$\hat{y} = \hat{x}_1 + t\hat{x}_2 \qquad (l_2\text{line})$$

reveals that the l_1 line accurately reproduces the sense of the first seven data values, ignoring the eighth completely. The l_2 line is as strongly influenced by the eighth data value as it is by all of the first seven put together. Consequently, the l_2 line shows no reasonable agreement with any portion of the data.

It is interesting to note that the l_1 line for the above example interpolates two of the data points, the third and the seventh. That is

$$f_3(\bar{x}) = h(t_3;\bar{x}) - y_3 = 0$$
$$f_7(\bar{x}) = h(t_7;\bar{x}) - y_7 = 0 \ .$$

This results from the fact that $\bar{x}$ is determined from the linear program (1.2), in which the components of x are unrestricted, implying that these two components must be basic at an optimum. And the structure of (1.2) in turn implies that u, v pairs associated with two equality constraints must be nonbasic at an optimum; i.e. the corresponding two deviations must be zero. More generally, if $x \in \mathbf{R}^n$, one can expect to have at least n of the data points interpolated.

The algorithm surveyed in this paper is a natural extension of the algorithm in [1] to nonlinear problems, using the material of [3,4] as a theoretical foundation. As in [1], we exploit the interpolation feature.

For the sake of simplicity, no constraints appear in (1.1). The discussion in [1] however, where linear constraints for the linear version of (1.1) were incorporated by adding penalty terms to F which could be handled by the algorithm in a natural way, generalizes directly to nonlinear problems with linear and nonlinear constraints. Our current software, in fact, solves the problem

$$\underset{x}{\text{minimize}} \ \sum_{i=1}^{m} |f_i(x)|$$

$$\text{subject to } \ f_j(x) \geqslant 0\,, \quad \text{for } j = m+1, \ldots, m+l$$

$$\text{and} \qquad f_k(x) = 0\,, \quad \text{for } k = m+l+1, \ldots, m+l+s$$

The exposition is further simplified by assuming the availability of exact first and second derivatives rather than considering gradient differencing and/or quasi-Newton methods, though both can be employed in obvious ways.

2. Notation, Definitions, Assumptions

All norms ($\|\cdot\|$) used below are Euclidian norms.

We will organize our thoughts around the set (possibly vacuous) of those f_i in (1.1) which will be zero at a local minimum of F. For a given x and $\epsilon \geqslant 0$ let

$$\mathbf{A}(x,\epsilon) = \{j : |f_j(x)| \leqslant \epsilon \textbf{ and } 1 \leqslant j \leqslant m\}$$

denote the set of ϵ*−active indices* at x. Let

$$\mathbf{I}(x,\epsilon) = \{1, \ldots, m\} - \mathbf{A}(x,\epsilon)$$

represent the ϵ*−inactive indices* and

$$\sigma_i(x) = \operatorname{sgn}[f_i(x)]; \quad i \in \mathbf{I}(x,\epsilon) \ .$$

Let

$$\phi(x,\epsilon) = \sum_{i \in \mathbf{I}(x,\epsilon)} \sigma_i(x) f_i(x) \ ,$$

and note that

$$\phi(x,0) = \sum_{i \in \mathbf{I}(x,0)} \sigma_i(x) f_i(x) = \sum_{i \in \mathbf{I}(x,0)} |f_i(x)| = \sum_{i=1}^{m} |f_i(x)| = F(x)$$

for any x. It is easily seen, moreover, that for any designated point $\hat{x}$ there is an $\hat{\epsilon} > 0$ such that for all $0 \leqslant \epsilon \leqslant \hat{\epsilon}$

$$\mathbf{A}(\hat{x},\epsilon) = \mathbf{A}(\hat{x},0)$$

and

$$\phi(\hat{x},\epsilon) = F(\hat{x}) \ .$$

With ϵ fixed, there is a neighborhood $\hat{N}$ of $\hat{x}$ depending upon ϵ, for which $x \in \hat{N}$ implies

$$\mathbf{A}(x,\epsilon) = \mathbf{A}(\hat{x},0) \ ;$$

and also given any $\delta > 0$ there is a neighborhood $\hat{N}_\delta \subseteq \hat{N}$ of $\hat{x}$ within which

$$|\phi(x,\epsilon) - F(x)| < \delta \ .$$

Thus, we approach the task of solving (1.1) from the point of view of selecting a trial active index set $\mathbf{A}$ and its complimentary index set $\mathbf{I}$ suggested locally by x and some activity tolerance ϵ. (Usually $\mathbf{A} = \mathbf{A}(x,\epsilon)$, but we have sometimes found it useful to consider $\mathbf{A} \subseteq \mathbf{A}(x,\epsilon)$, and so we will distinguish the $\mathbf{A}$ of our choice from $\mathbf{A}(x,\epsilon)$ in what follows.) Then, with $\mathbf{A}$ chosen, we begin solving

$$\underset{x}{\text{minimize}}\ \phi(x) = \sum_{i \in \mathbf{I}} \sigma_i(x) f_i(x) \tag{2.1}$$
$$\text{subject to } f_j(x) = 0; \ j \in \mathbf{A}$$

until we are sufficiently close to a minimum of F to satisfy a convergence criterion, or until it is evident that we are dealing with the wrong selection of $\mathbf{A}$ in which case (2.1) is redefined (by redefining $\mathbf{A}$) for further steps of the algorithm. The approach to solving problems of the form (2.1) will be the one taken in [3,4], for which it can be shown, under reasonable assumptions, that the sequence of function values F which are generated decrease monotonically to a locally minimal value.

For ease of discussion we will begin with the following

Assumption

1. The vectors $\nabla f_j(x)$, $j \in \mathbf{A}$ are linearly independent for each $x, \mathbf{A}$ we choose to consider. (See section 5 for further comments on this.)

We summarize the applicable results and defer their full development to a later paper.

Definition: A point $\bar{x}$ is a *stationary point* of F, if there exist scalars u_j, $j \in \mathbf{A}(\bar{x}, 0)$, such that

$$\sum_{i \in \mathbf{I}(\bar{x},0)} \sigma_i(\bar{x}) \nabla f_i(\bar{x}) = \sum_{j \in \mathbf{A}(\bar{x},0)} u_j \nabla f_j(\bar{x}) \ .$$

Definition: A stationary point $\bar{x}$ of F is a *first-order point* of F, if

$$-1 \leqslant u_j \leqslant +1 \, ; \quad j \in \mathbf{A}(\bar{x}, 0) \ . \tag{2.3}$$

Definition: A first-order point $\bar{x}$ of F is a *strict second-order point* of F, if

$$-1 < u_j < +1 \, , \quad j \in \mathbf{A}(\bar{x}, 0) \ ,$$

and for each nonzero vector $p \in \mathbf{R}^n$ satisfying

$$p^T \nabla f_j(\bar{x}) = 0 \, , \quad j \in \mathbf{A}(\bar{x}, 0) \ , \tag{2.4}$$

it follows that

$$p^T B(\bar{x}) p > 0 \tag{2.5}$$

where

$$B(\bar{x}) = \sum_{i \in \mathbf{I}(\bar{x},0)} \sigma_i(\bar{x}) \nabla^2 f_i(\bar{x}) - \sum_{j \in \mathbf{A}(\bar{x},0)} u_j \nabla^2 f_j(\bar{x}) \ . \tag{2.6}$$

The following are corollaries of results from [2].

Theorem 1: A necessary condition for the point $\bar{x}$ to be a local minimizer for F is that $\bar{x}$ be a first-order point for F.

Theorem 2: A sufficient condition for $\bar{x}$ to be a strong local minimizer for F is that $\bar{x}$ be a strict second-order point for F.

If $\mathbf{A}$ is chosen as $\mathbf{A}(\bar{x}, 0)$ in (2.1), then the following is a standard result.

Theorem 3: A necessary condition for $\bar{x}$ to be an optimal point for (2.1) is that $\bar{x}$ be a stationary point for F.

The indication which we may take that our choice of (2.1) -- that is, our choice of $\mathbf{A}$ -- is an incorrect representation of (1.1) is that in proceeding toward an optimum of (2.1) we appear to be nearing a stationary point of F which is not a first-order point of F. On such occasions we change (2.1) by *dropping* an index j_0 from $\mathbf{A}$ as suggested by the following sequence of definitions and results.

Definition: The values $u_j = u_j(x)$, $j \in \mathbf{A}$ defined from (2.1) for a given $x \in \mathbf{R}^n$ by the solution to the least squares problem

$$\underset{u}{\text{minimize}} \ \| \ \nabla f_{\mathbf{A}}(x) u - \nabla \phi \ \| \ \ , \tag{2.7}$$

where

$$\nabla \phi(x) = \sum_{i \in \mathbf{I}} \sigma_i \nabla f_i(x) \tag{2.8}$$

and $\nabla f_{\mathbf{A}}(x)$ is a matrix with columns $\nabla f_j(x)$, $j \in \mathbf{A}$, are the *first-order multiplier estimates* at x.

Definition: Any vector $p \in \mathbf{R}^n$ defines a *descent direction* for F at a point x, if $F(x+\alpha p) < F(x)$ for all $\alpha > 0$ in some neighborhood of 0.

It follows easily from the material in [1] that, if $\bar{x}$ is a stationary point of (2.1) which is not a first-order point for F and $u_j = u_j(x)$, $j \in \mathbf{A}$ are the associated first-order multiplier estimates, then the vector d given by the solution to the linear system

$$\begin{aligned} \nabla f_{j_0}(\bar{x})^T d &= -\mathrm{sgn}(u_{j_0}) \\ \nabla f_j(\bar{x})^T d &= 0, \quad j \in \mathbf{A} - \{j_0\} , \end{aligned} \tag{2.9}$$

where $j_0 \in \mathbf{A}$ is some index for which $|u_{j_0}| > 1$, will define a descent direction for F at $\bar{x}$. This result can be extended for the nonlinear problem under discussion to the following.

Theorem 4: If $\bar{x}$ is a stationary point for (2.1) which is not a first-order point for F, if x is close enough to $\bar{x}$ in the sense that $\mathbf{A}(x, \epsilon) = \mathbf{A}(\bar{x}, 0)$, and if $\mathbf{A}$ in (2.1) coincides with $\mathbf{A}(\bar{x}, 0)$, if $\sigma_i(x) = \mathrm{sgn}[f_i(x)] = \mathrm{sgn}[f_i(\bar{x})]$ for all $i \in \mathbf{A}$ and if u_j, $j \in \mathbf{A}$ are the first-order multiplier estimates at x, then the vector $d \in \mathbf{R}^n$ determined as in (2.9) with $\bar{x}$ replaced by x will define a descent direction for F.

Definition: The step direction d defined above will be called the *dropping direction*

The algorithm for (1.1) to be described uses the dropping direction whenever the selected version of (2.1) (i.e. the choice of **A**) is not an appropriate model for (1.1). Otherwise one or both of a pair of directions based directly upon (2.1) is used.

Definition: A *horizontal direction* at x for (2.1) is a vector $h \in \mathbf{R}^n$ which solves the following equality-constrained quadratic programming problem

$$\begin{aligned} &\underset{h}{\text{minimize}} \; \tfrac{1}{2} h^T Q h + h^T \nabla \phi(x) \\ &\text{subject to } \nabla f_{\mathbf{A}}(x)^T j = 0 \end{aligned} \tag{2.10}$$

for a positive definite matrix Q. During the course of the algorithm two choices of Q are considered:

$$Q = \nabla^2 \phi(x) + D \tag{2.11}$$

for a diagonal matrix D with nonnegative diagonal entries (possibly zero) as needed to ensure positive definiteness, and (as suggested by (2.6))

$$Q = B(x) = \nabla^2 \phi(x) - \sum_{j \in \mathbf{A}} u_j \nabla^2 f_j(x) , \tag{2.12}$$

where u_j, $j \in \mathbf{A}$ are the first-order multiplier estimates at x. If it is necessary to distinguish which choice of Q is used to produce h, we will use h° to indicate that (2.11) was used and $h^{\circ\circ}$ to indicate that (2.12) was used.

Definition: Let $\tilde{x}$ be a designated "reference point". A *vertical direction* at x referenced to $\tilde{x}$ is a vector $v \in \mathbf{R}^n$ which solves the following least squares problem

$$\underset{v}{\text{minimize}} \; \| \nabla f_{\mathbf{A}}(\tilde{x})^T v + f_{\mathbf{A}}(x) \| , \tag{2.13}$$

where $f_{\mathbf{A}}(x)$, $j \in \mathbf{A}$ (NB: x **not** $\tilde{x}$) denotes the vector of function values $f_j(x)$, arranged consistently with the columns of $\nabla f_{\mathbf{A}}(\tilde{x})$.

During the course of our algorithm a line search will be used for dropping directions and for some horizontal directions. Whenever $x + \alpha^* d$ or $x + \alpha^* h$ is written for any point x, it will be assumed that $\alpha^* > 0$ has been chosen to provide *sufficient decrease* in the sense that

$$F(x + \alpha^* d) < F(x) - \eta [d^T \nabla \phi(x)]^2 \tag{2.14}$$

for some chosen tolerance $\eta > 0$, and similarly for h.

Assumptions (continued)

2. $f_i \in C^2$ for all i.
3. Any points x to be considered are confined to a compact set **S**.
4. (2.14) holds with respect to each α^* chosen.
5. There are only finitely many stationary points $\bar{x}$ in **S**.
6. All first-order points of F in **S** are strict second-order points.
7. There exist numbers $U \geqslant L > 0$, $\pi > 0$ such that

$$L \| y \|^2 \leqslant y^T B(x_k) y \leqslant U \| y \|^2$$

for any y satisfying $y^T \nabla f_j(x_k) = 0$, all $j \in \mathbf{A}$, whenever $\| x_k - \bar{x} \| < \pi$ for any first-order point $\bar{x}$ in **S**.

Under the above we have the following results more or less directly from [3,4].

Theorem 5: Let $\mathbf{A} = \mathbf{A}(\bar{x}, 0)$ define the problem (2.1), and assume that $\bar{x}$ is a first-order point for F. Then there is a neighborhood of $\bar{x}$ such that, for all $\tilde{x}$, x in that neighborhood,

$$F(x + v) < F(x) \ , \tag{2.15}$$

where v is the vertical direction at x referenced to $\tilde{x}$.

Theorem 6: Let $\bar{x}$ be a stationary point for (2.1), with $\mathbf{A} = \mathbf{A}(\bar{x}, 0)$. Assume that $\bar{x}$ is also a strict second-order point for F. Then there is a neighborhood of $\bar{x}$ such that, for any x in that neighborhood with

$$F(x + h^{\circ\circ} + v) < F(x) \tag{2.16}$$

where v is the vertical direction at $x + h^{\circ\circ}$ referenced to x.

Theorem 7: Given any instance of (2.1) and any x, the horizontal directions h°, $h^{\circ\circ}$ as given in (2.11), (2.12) respectively are descent directions for F.

Definition: For any designated point x, and any chosen d, $h \in \{ h^{\circ}, h^{\circ\circ} \}$, v consistent with the above, the transitions

$$x \rightarrow x + \alpha^* d \tag{2.17}$$

$$x \rightarrow x + \alpha^* h$$

$$x \rightarrow x + v$$

$$x \rightarrow x + h^{\circ\circ} + v$$

will be called respectively a *dropping step*, a *horizontal step*, a *vertical step* and a *Newton step* taken from the point x.

We are now prepared to outline our algorithm and its convergence properties.

3. Algorithm: Strategy and Convergence

Assume that **A** has been chosen and we are considering the resulting problem (2.1) at some point $\hat{x}$. We can be either (a) very close to a stationary point of (2.1) which, by virtue of having correctly identified **A**, is a first-order point of F, (b) nowhere near a first-order point of F and trying to move closer to a stationary point of (2.1) as a means of decreasing the value of F, (c) somewhat near a stationary point of (2.1) and interested in testing whether **A** is corectly chosen. Thus, we associate three regions with each stationary point $\bar{x}$ of (2.1). R_1 consists of those points x which, loosely stated, are so distant from $\bar{x}$ that (2.2) is far from satisfied. In such circumstances the multiplier estimates u_j will not be required, which is fortunate since they can be expected to be quite unreliable. R_2 consists of those points x which are closer to $\bar{x}$ in the sense that (2.2) is rather well satisfied, and the associated first-order multiplier estimates clearly indicate the character of the true multipliers. R_3 consists of those points which are within the intersection of the neighborhoods mentioned by theorems 5 and 6 and so close to $\bar{x}$ that (2.2) is very well satisfied.

To be more specific, let $\tau_{12} > \tau_{23} > 0$ be given. Then

$$R_1 = \left\{ x : \| \nabla f_{\mathbf{A}}(x) u(x) - \nabla \phi(x) \| \geqslant \tau_{12} \right\}$$

$$R_2 = \left\{ x : x \notin R_1 \bigcup R_3 \ \textbf{and} \ \| \nabla f_{\mathbf{A}}(x) u(x) - \nabla \phi(x) \| < \tau_{12} \right\}$$

$$R_3 = \left\{ x : x \in \{\text{nbd. of Thm. 5}\} \bigcap \{\text{nbd. of Thm. 6}\} \ \textbf{and} \ \| \nabla f_{\mathbf{A}}(x) u(x) - \nabla \phi(x) \| < \tau_{23} \right\}$$

In practice we can never determine exactly into which region our given point $\hat{x}$ falls, but this is inessential to the success of the algorithm. The global decrease of F to a locally minimal value is guaranteed under reasonable assumptions, and these regions are used only to govern our strategy of approach to the corresponding minimizer $\bar{x}$. Failure to determine the regions optimally, at worst, defers the time after which superlinearity of the convergence rate sets in. Our desire is to avoid estimating multipliers and confine ourselves to horizontal steps (viz. (2.17)) in R_1, to estimate multipliers and determine whether a dropping step is necessary in R_2, to use vertical steps as well as horizontal steps in R_2 if dropping is not called for, and finally to switch to Newton steps for fast convergence to $\bar{x}$ in R_3.

We base our assumption about which region contains a given point $\hat{x}$ largely upon the magnitude of $\| \nabla f_{\mathbf{A}}(\hat{x}) u - \nabla \phi(\hat{x}) \|$, and we back up this assumption by verification tests which we apply to the dropping direction, to the vertical step and to the Newton step as appropriate.

With this general introduction, the algorithm is most easily described in terms of a table. The tolerance parameters $\tau_{12} > \tau_{23} > 0$ have been introduced above. Three more positive parameters δ_1, δ_2 and δ_3 are needed for determining whether to reject respectively a dropping step direction, a vertical step or a Newton step on the grounds that they will not yield acceptable decrease in F. Each iteration of the algorithm starts with a current point $\hat{x}$, chooses $\mathbf{A}$ anew as $\mathbf{A}(\hat{x}, \epsilon)$, and uses $\hat{x}$, $\mathbf{A}$ to test a condition (column 1 in the table below) which serves to pick out a table row. Each row is associated with an assumption about the R-region in which $\hat{x}$ is located (column 2). A verification of this assumption may be carried out by performing a follow-up test (column 3). Finally, an adjustment to $\hat{x}$ is made in one fashion if the test result is positive and in another fashion if the result is negative (columns 4 and 5 respectively).

Prevailing Condition	Corresponding Assumption	Follow-up Test	Positive Result	Negative Result
$\|\nabla f_{\mathbf{A}} u - \nabla \phi\| \geqslant \tau_{12}$	$\hat{x} \in R_1$	none	$\hat{x} \leftarrow \hat{x} + \alpha^* h^{\circ}$	not applicable
$\tau_{23} \leqslant \|\nabla f_{\mathbf{A}} u - \nabla \phi\| < \tau_{12}$	$\hat{x} \in R_2$ and no drop	$\|u_j\| \leqslant 1$, all $j \in \mathbf{A}$ or $\|u_{j_0}\| > 1$, some $j_0 \in \mathbf{A}$ but $d^T \nabla \phi \geqslant -\delta_1$	$\hat{x} \leftarrow \hat{x} + \alpha^* h^{\circ}$ followed by $\hat{x} \leftarrow \hat{x} + v$ if $F(\hat{x} + v) < F(\hat{x}) - \delta_2 \|f_{\mathbf{A}}\|$	$\hat{x} \leftarrow \hat{x} + \alpha^* d$
$\|\nabla f_{\mathbf{A}} u - \nabla \phi\| < \tau_{23}$ and Positive Result of table lines 2 or 3 ocurred last step and $\hat{x} \leftarrow \hat{x} + v$ was successful in the case of line 2	$\hat{x} \in R_3$	$F(\hat{x} + h^{\circ\circ} + v)$ $< F(\hat{x}) - \delta_3$	$\hat{x} \leftarrow \hat{x} + h^{\circ\circ} + v$	$\hat{x} \leftarrow \hat{x} + \alpha^* h^{\circ\circ}$ followed by $\hat{x} \leftarrow \hat{x} + v$ if $F(\hat{x} + v) <$ $F(\hat{x}) - \delta_2 \|f_{\mathbf{A}}\|$

In addition to the above, ϵ is reduced whenever $F(\hat{x}+v) < F(\hat{x}) - \delta_2\|f_{\mathbf{A}}\|$ fails to hold, whenever linear dependencies in $\nabla f_{\mathbf{A}}$ are detected while computing d, or whenever the condition $|f_{j_0}(\hat{x}+\alpha^* d)| < \epsilon$ is detected following the "Negative Result" outcome in line 2. The reduction of ϵ is heuristic in the first case (division by two), in the second case the reduction is just enough to discard indices j from $\mathbf{A}$ associated with vectors ∇f_j which are contributing dependencies and in the third case the reduction is just enough to ensure that j_0 will not reappear in $\mathbf{A}$ at the next pass through the table, i.e. just enough to ensure that f_{j_0} is properly "dropped".

Let $k \in \{1, 2, 3, \dots\}$ index the sequence of iterations which are performed, and let $\hat{x}_k$, ϵ_k and $\mathbf{A}_k$ denote the $\hat{x}$, ϵ and $\mathbf{A}$ encountered at the k-th iteration step. Under the 7 assumptions stated on an earlier page, with the set $\mathbf{S}$ mentioned therein, and for all $\eta > 0$ in (2.14) sufficiently small, the discussion in [3,4] provides us with the following

Results

a. The sequence of values ϵ_k is bounded away from zero, and the elements of the sequence of index sets $\mathbf{A}_k$ are identical for all k sufficiently large.

b. There exists a first-order point $\bar{x} \in \mathbf{S}$ such that $\lim_{k \to \infty} \hat{x}_k = \bar{x}$.

c. For all k sufficiently large, only the positive outcome of line 3 in the table occurs (i.e only Newton steps are taken).

d. $\lim_{k \to \infty} \dfrac{\| \hat{x}_{k+1} - \bar{x} \|}{\| \hat{x}_{k-1} - \bar{x} \|} = 0$, implying superlinear convergence.

4. Test Problems

Some representative problems from [5,6] have been tested. They are referred to below by the labels they were given in these references. In each case we report the number of evaluations of the function F and the number of iteration steps of the algorithm ($x \to x + \Delta x$) which were required to go from the stated initial point to a point having 7-place agreement with the solution (effectively single-precision accuracy on our computer, the Honeywell 6600). It should be noted that our method, exclusive of what might be required by the line search being used, needs one evauation of all f_i, ∇f_i (that is, one evaluation of F and $\nabla \phi$) together with one updated version of the matrix Q. (approximating $\nabla^2 \phi$ or B according to the nearness of x to a stationary point) for each iteration. In addition, if the iteration step involves a vertical segment v, a further evaluation of f_j, $j \in \mathbf{A}$, is required. Any disparity shown below between the number of function evaluations and the number of iteration steps is the responsibility of the particular line search being employed -- an exceedingly crude and simplistic one in our current code.

Where possible we have tried to give a comparative figure for function evaluations and iteration steps as they are indicated in [5,6]. It will be seen from this that our method appears to be comparable to the one described in [6] and significantly better than the one described in [5]. Further computational testing and refinement are in progress.

Problem 1 ([5] Example 5.1):

$f_1(x) = x_1^2 + x_2 - 10$

$f_2(x) = x_1 + x_2^2 - 7$

$f_3(x) = x_1^2 - x_2^3 - 1$

Initial Point: $x_1 = 1, x_2 = 2$
Minimal Point:
$x_1 = 2.842503,$
$x_2 = 1.920175$
Minimum F Value: 4.704243E-01
Function Evaluations: 14

Iteration Steps: 6
(The algorithm of [5] had not attained comparable accuracy after 12 iterations and 88 function evaluations.)

Problem 2 ([5] Example 5.2, [6] Problem 4):

$$f_1(x) = x_1^2 + x_2^2 = x_3^2 - 1$$

$$f_2(x) = x_1^2 + x_2^2 + (x_3-2)^2$$

$$f_3(x) = x_1 + x_2 + x_3 - 1$$

$$f_4(x) = x_1 + x_2 - x_3 + 1$$

$$f_5(x) = 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2$$

$$f_6(x) = x_1^2 - 9x_3$$

Initial Point: $x_1 = 1$, $x_2 = 1$, $x_3 = 1$
Minimal Point:
$x_1 = 0.5360725$
$x_2 =$ -3.2E-10 ≈ 0
$x_3 = 0.03193041$
Minimum F Value: 7.894227
Function Evaluations: 20
Iteration Steps: 10
(The algorithm of [5] attained a comparable value of F only after 11 iteration steps and 109 function evaluations. The results in [6] are recorded only to 5 figures. Agreement in the minimal F value to that number of figures was reported attained after 10 iterations and 11 function evaluations.)

Problem 3 ([5] Example 6.1):

$$y = y(t) = \tfrac{1}{2}e^{-t} - e^{-2t} + \tfrac{1}{2}e^{-3t} + \tfrac{3}{2}e^{-3t/2}\sin(7t) + e^{-5t/2}\sin(5t)$$

$$h(t;x) = x_1 e^{-x_2 t}\cos(x_3 t + x_4) + x_5 e^{-x_6 t}$$

$$f_i(x) = h(t_i;x) - y(t_i) = h(t_i;x) - y_i$$

where $t_i = 0.0 + (i-1)/10.0$, $i = 1, \ldots, 51$.
Initial point:
$x_1 = 2$, $x_2 = 2$, $x_3 = 7$
$x_4 = 0$, $x_5 = -2$, $x_6 = 1$
Minimal Point:
$x_1 = -2.240744$
$x_2 = 1.857688$
$x_3 = 6.770049$
$x_4 = 1.496694$
$x_5 = 0.1658920$
$x_6 = 0.7422845$
Minimum F Value: 0.5598131
Function Evaluations: 78
Iteration Steps: 68
(The Algorithm of [5] converges to a different minimal point having the same minimum F value. After 11 iterations and 116 function evaluations it had attained the F value 0.559818.)

Problem 4 ([6] Problem 3):

$f_1(x) = x_1^2 + x_2^2 + x_1 x_2$

$f_2(x) = \sin(x_1)$

$f_3(x) = \cos(x_2)$

Initial Point: $x_1 = 3, x_2 = 1$
Minimal Point:
$x_1 = 0.000000$
$x_2 =$ -8.3E-04 ≈ 0.0
Minimum F Value: 1.000000
Function Evaluations: 20
Iteration Steps: 16
(The Algorithm of [6] reports termination at the point $x_1 = 0$, $x_2 =$ 2.0E-04 after 15 iterations and 15 function evaluations.)

5. Of Things Not Mentioned

One would expect l_1 minimization to be used most frequently in the context of data fitting; e.g. the simple linear model problem in the first section or test problem 3 above. Moreover, l_1 minimization will be chosen in these contexts over l_2 minimization often because there is a suspicion about the validity of a small portion of the data; e.g. the simple linear model problem. The vast majority of the data may be excellent, and this will mean that the vast majority of the functions $f_i(x) = h(t_i;x) - y_i$ may be nearly zero at some points x which the algorithm must consider. The gradients ∇f_i are likely to be linearly dependent at such points; i.e. the algorithm will have to contend with degeneracy. We believe that horizontal, vertical and Newton steps remain reasonable and retain their theoretic properties under some weaker constraint qualification than that of linear independence (Assumption 1) if the choice of $\mathbf{A}$ is taken to be a subset of $\mathbf{A}(x,\epsilon)$ associated with a spanning collection of the vectors $\nabla f_j(x)$, $j \in \mathbf{A}(x,\epsilon)$. A similar situation is not evident for dropping steps. Thus, the algorithm as described could be expected to have dificulty in R_2 regions about degenerate stationary points. This is an area of further research.

Bibliography

[1] Bartels, R.H. and Conn, A.R. Linearly constrained discrete l_1 problems. ACM Trans. on Math. Software 6 (1980) 594-608

[2] Coleman, T.F. and Conn, A.R. Second-order conditions for an exact penalty function. Math. Prog. 19 (1980) 178-185.

[3] Coleman, T.F. and Conn, A.R. Nonlinear programming via an exact penalty function: global analysis. Math. Prog. *to appear.*

[4] Colman, T.F. and Conn, A.R. Nonlinear programming via an exact penalty function: asymptotic analysis. Math. Prog. *to appear.*

[5] El-Attar, R.A.; Vidyasagar, M. and Dutta, S.R.K. An Algorithm for l_1-norm minimization with application to nonlinear l_1-approximation. SIAM J. Num. Anal. 16 (1979) 70-86.

[6] Overton, M.L. and Murray, W. A Projected Lagrangian algorithm for nonlinear l_1 optimization. SIAM J. Sci. and Stat. Comp. *to appear.*